# LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý thầy cô ở Khoa Công nghệ thông tin – Trường Đại Học Công Nghệ Thông Tin & Truyền Thông Thái Nguyên đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Và hơn hết để đồ án tốt nghiệp này đạt được kết quả tốt đẹp em xin chân thành gửi tới lời cảm ơn thầy ThS. **Nguyễn Hồng Tân** đã tận tâm hướng dẫn trực tiếp và gián tiếp giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài quan trọng nhất đối với một sinh viên. Một lần nữa, em xin chân thành cảm ơn thầy.

Không thể không nhắc tới sự giúp đỡ nhiệt tình của các anh chị HR và team Smart Switch tầng 7 Application 2 của Trung tâm nghiên cứu và phát triển điện thoại di động Samsung Viêt Nam - SVMC, đã tạo điều kiện thuận lợi nhất cho em trong suốt thời gian ôn luyện thuật toán APS2 – Level Advand và thực tập làm đề tài đồ án:

**“Xây dựng ứng dụng chuyển dữ liệu trên smartphone”**

Báo cáo đồ án được thực hiện trong khoảng thời gian gần 15 tuần. Bước đầu đi vào thực tế, tìm hiểu về lĩnh vực sáng tạo trong nghiên cứu khoa học, kiến thức của em còn hạn chế và còn nhiều bỡ ngỡ. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Sau cùng, em xin kính chúc tất cả các quý thầy cô trong khoa Công nghệ thông tin và trường thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Em xin chân thành cảm ơn!

# LỜI CAM ĐOAN

Em xin cam đoan rằng số liệu và kết quả nghiên cứu trong đồ án này là trung thực không trùng lặp với các đề tài khác. Em cũng xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện đồ án này đã được cảm ơn và các thông tin trích dẫn trong đồ án đã chỉ rõ nguồn gốc.

**Sinh viên**

**Nguyễn Văn Lâm**

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc41464739)

[LỜI CAM ĐOAN 3](#_Toc41464740)

[MỤC LỤC 4](#_Toc41464741)

[DANH MỤC HÌNH ẢNH 7](#_Toc41464742)

[MỞ ĐẦU 8](#_Toc41464743)

[i. Lý do chọn đề tài 8](#_Toc41464744)

[ii. Mục đích của đề tài 8](#_Toc41464745)

[iii. Ý nghĩa của đề tài 8](#_Toc41464746)

[iv. Tại sao đề tài này cần thiết 9](#_Toc41464747)

[v. Mục tiêu của đề tài 9](#_Toc41464748)

[vi. Phương hướng của đề tài 10](#_Toc41464749)

[vii. Kết quả đạt được 10](#_Toc41464750)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 11](#_Toc41464751)

[1.1. Các thành phần cơ bản trong android 11](#_Toc41464752)

[1.1.1 Cấu trúc của ứng dụng android 11](#_Toc41464753)

[1.1.2 Lớp Linux Kernel 11](#_Toc41464754)

[1.1.3 Vòng đời của Activity 16](#_Toc41464755)

[1.1.4 Service 16](#_Toc41464756)

[1.1.5 Broadcast Receiver 19](#_Toc41464757)

[1.2 Content providers 22](#_Toc41464758)

[1.2.1 Khái niệm 22](#_Toc41464759)

[1.2.2 Cách sử dụng Content Provider 23](#_Toc41464760)

[1.2.3 Content URI 24](#_Toc41464761)

[1.3 Peer-To-Peer Và Client-Server 24](#_Toc41464762)

[1.3.1 Client – Server là gì? 24](#_Toc41464763)

[1.3.2 Peer to peer (P2P) là gì? 26](#_Toc41464764)

[1.3.3 Sự khác nhau 26](#_Toc41464765)

[1.3.4 Ưu điểm 27](#_Toc41464766)

[1.4 Wifi-Direct 28](#_Toc41464767)

[1.4.1 Khái niệm 28](#_Toc41464768)

[1.4.2 So sánh Wifi-Direct, Bluetooth, NFC 28](#_Toc41464769)

[1.4.3 Cách hoạt động 30](#_Toc41464770)

[CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ ỨNG DỤNG 32](#_Toc41464771)

[2.1 Khảo sát hệ thống 32](#_Toc41464772)

[2.2 Phân tích hệ thống 32](#_Toc41464773)

[2.2.1 Mô tả bài toán 32](#_Toc41464774)

[2.2.2 Ý nghĩa của bài toán 33](#_Toc41464775)

[2.2.3 Tại sao đề tài này cần thiết 34](#_Toc41464776)

[2.2.4 Xác định yêu cầu của hệ thống 34](#_Toc41464777)

[2.2.5 Xác định tác nhân của hệ thống 34](#_Toc41464778)

[2.2.6 Chức năng chính của ứng dụng 34](#_Toc41464779)

[2.3 Thiết kế hệ thống 36](#_Toc41464780)

[2.3.1 Xây dựng biểu đồ use case 36](#_Toc41464781)

[2.3.2 Xây dựng biểu đồ trạng thái 40](#_Toc41464782)

[2.3.3 Xây dựng biểu đồ trình tự 42](#_Toc41464783)

[2.3.4 Xây dựng biểu đồ hoạt động 44](#_Toc41464784)

[CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG 46](#_Toc41464785)

[3.1 Phương pháp xây dựng ứng dụng 46](#_Toc41464786)

[3.2 Xây dựng ứng dụng Transfer Data 47](#_Toc41464787)

[3.2.1 Màn hình chính 47](#_Toc41464788)

[3.2.2 Màn hình tìm các thiết bị xung quanh 48](#_Toc41464789)

[3.2.3 Màn hình chờ kết nối 49](#_Toc41464790)

[3.2.4 Popup chờ xác nhận kết nối 50](#_Toc41464791)

[3.2.5 Popup xác nhận kết nối 51](#_Toc41464792)

[3.2.6 Màn hình chờ gửi dữ liệu 52](#_Toc41464793)

[3.2.7 Màn hình hiển thị danh sách các dữ liệu gửi 53](#_Toc41464794)

[3.2.8 Màn hình lọc danh bạ 54](#_Toc41464795)

[3.2.9 Màn hình lọc tin nhắn 55](#_Toc41464796)

[3.2.10 Màn hình chọn hình ảnh 56](#_Toc41464797)

[3.2.11 Màn hình chọn âm thanh và tệp tin 57](#_Toc41464798)

[3.2.12 Màn hình hiển thị quá trình chuyển dữ liệu trên máy gửi và nhận, restore dữ liệu trên máy nhận 58](#_Toc41464799)

[3.2.13 Mã hoá và giải mã giữ liệu 59](#_Toc41464800)

[KẾT LUẬN 60](#_Toc41464801)

[TÀI LIỆU THAM KHẢO 61](#_Toc41464802)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 62](#_Toc41464803)

# DANH MỤC HÌNH ẢNH

[Hình 1.1.1 Cấu trúc của ứng dụng Android 11](#_Toc41380739)

[Hình 1.1.3 Vòng đời của Activity 16](#_Toc41380740)

[Hình 1.1.4 Service 17](#_Toc41380741)

[Hình 1.1.5-1 Braodcast Receiver 19](#_Toc41380742)

[Hình 1.1.5‑2 Đăng kí Broadcast Receiver trong android 20](#_Toc41380743)

[Hình 1.2.1‑1 Content Provider 22](#_Toc41380744)

[Hình 1.2.1‑2 Tải dữ liệu lên UI sử dụng CursorLoader 23](#_Toc41380745)

[Hình 2.3.1‑1 Biểu đồ Use Case 36](#_Toc41380746)

[Hình 2.3.1‑2 Biểu đồ phân rã Use Case 36](#_Toc41380747)

[Hình 2.3.2‑1 Biểu đồ trạng thái UC kết nối 40](#_Toc41380748)

[Hình 2.3.2‑2 Biểu đồ trạng thái UC chuyển dữ liệu 40](#_Toc41380749)

[Hình 2.3.2‑3 Biểu đồ trạng thái UC chọn loại dữ liệu 41](#_Toc41380750)

[Hình 2.3.2‑4 Biểu đồ trạng thái UC lọc dữ liệu 41](#_Toc41380751)

[Hình 2.3.3‑1 Biểu đồ trình tự UC kết nối 42](#_Toc41380752)

[Hình 2.3.3‑2 Biểu đồ trình tự UC truyền dữ liệu 42](#_Toc41380753)

[Hình 2.3.3‑3 Biểu đồ trình tự UC chọn loại dữ liệu 43](#_Toc41380754)

[Hình 2.3.3‑4 Biểu đồ trình tự UC lọc dữ liệu 43](#_Toc41380755)

[Hình 2.3.4‑1 Biểu đồ hoạt động UC kết nối 44](#_Toc41380756)

[Hình 2.3.4‑2 Biểu đồ hoạt động UC chuyển dữ liệu 44](#_Toc41380757)

[Hình 2.3.4‑3 Biểu đồ hoạt động UC chọn loại dữ liệu 45](#_Toc41380758)

[Hình 2.3.4‑4 Biểu đồ hoạt động UC lọc dữ liệu 45](#_Toc41380759)

# MỞ ĐẦU

## Lý do chọn đề tài

Hiện nay điện thoại di động là một thứ không thể thiếu với mỗi chúng ta. Điện thoại ngày càng phát triển và thông minh hơn, các hãng điện thoại liên tục nghiên cứu và cho ra những mẫu sản phẩm mới với những tính năng và thiết kế mới. Vì lý do đó thời gian người dùng nâng cấp lên điện thoại mới ngày càng được rút ngắn. Trung bình người dùng sẽ đổi điện thoại sau khỏng 1 đến 1,5 năm.

Mỗi khi người dùng nâng cấp sang một chiếc điện thoại mới việc chuyển dữ liệu từ máy điện thoại cũ sang điện thoại mới là rất cần thiết. Vì vậy em đã quyết định chọn đề tài “*Xây dựng ứng dụng chuyển dữ liệu từ smartphone cũ sang smartphone mới*” ứng dụng sẽ giúp người dùng chuyển dữ liệu từ điện thoại cũ sang điện thoại mới 1 cách dễ dàng hơn.

## Mục đích của đề tài

Ứng dụng truyển dữ liệu từ smartphone cũa sang smartphone mới dựa trên hai mục đích chính như sau:

Thứ nhất là nghiên cứu các công nghệ lập trình: lập trình ứng dụng với android

Thứ hai là phát triển ứng dụng có tính thực tiễn cao, có khả năng triển khai ứng dụng vào thực tế, giúp những người dùng khi họ nâng cấp điện thoại có thể chuyển dữ liệu cá nhân của mình sang smartphone mới một cách nhanh chóng và tiện lợi.

Em hy vọng dựa trên nền tảng lý thuyết đã được thầy cô truyền thụ lại, kết hợp với việc tìm hiểu công nghệ của cá nhân, luận văn tốt nghiệp của em sẽ đạt được mục đích mong đợi

## Ý nghĩa của đề tài

Trước hết đề tài “*Xây dựng ứng dụng chuyển dữ liệu từ smartphone cũ sang smartphone mới*” là một ứng dụng như một đề tài thực hiện việc áp dụng những kiến thức quý báu đã được các thầy cô của trường Đại học Công Nghệ Thông Tin & Truyền Thông Thái Nguyên nhiệt tình truyền thụ lại cho em. Và đặc biệt là sự theo dõi và quan tâm giúp đỡ của thầy ThS.Nguyễn Hồng Tân trong suốt quá trình em thực hiện đề tài này.

Thứ hai là những kinh nghiệm quý báu mà em có được trong quá trình thực hiện đề tài sẽ là hành trang tuyệt vời giúp ích rất nhiều cho công việc sau này của mình.

Thứ ba trong thời đại công nghệ thông tin phát triển như vũ bão ngày nay, công nghệ không ngừng cải tiến, những công nghệ mới ra đời đòi hỏi người lập trình viên phải cố gắng nghiên cứu và tìm hiểu để áp dụng vào thực tiễn. Điều này cũng được thể hiện qua sự cố gắng của em khi quyết định thực hiện đề tài này với các công nghệ được dự báo sẽ phát triển mạnh mẽ trong tương lai.

Do vậy, em rất hi vọng sẽ tạo nên một sản phẩm hoàn chỉnh nhất để có thể đáp lại sự nhiệt tình giúp đỡ của các thầy cô.

## Tại sao đề tài này cần thiết

Vì các phương pháp truyền dữ liệu từ smartphone cũ sang smartphone mới hiện nay rất tốn thời gian và công sức của người dùng. Vì vậy em đã được hiện đề tài này nhắm tạo ra một phần mềm giúp cho việc chuyển dữ liệu tiện lợi và nhanh chóng hơn. Các tác vụ diễn ra hoàn toàn tự động sẽ giảm công sức và thời gian của người sử dụng

## Mục tiêu của đề tài

* Nâng cao kiến thức về lập trình Android.
* Hiểu biết thêm về các công nghệ kết nối áp dụng vào việc chuyển dữ liệu.
* Tìm hiểu được về các cơ chế bảo mật dữ liệu
* Hoàn thành được app android cho phép chuyển dữ liệu từ điện thoại cũ sang điện thoại mới

## Phương hướng của đề tài

* Nghiên cứu về wifi direct kết nối 2 thiết bị ngang hàng
* Tìm hiểu cách truyền file bằng socket
* Tìm hiểu cách quản lý và lấy data của hệ điều hành android
* Tìm hiểu mã hoá và giải mã.

## Kết quả đạt được

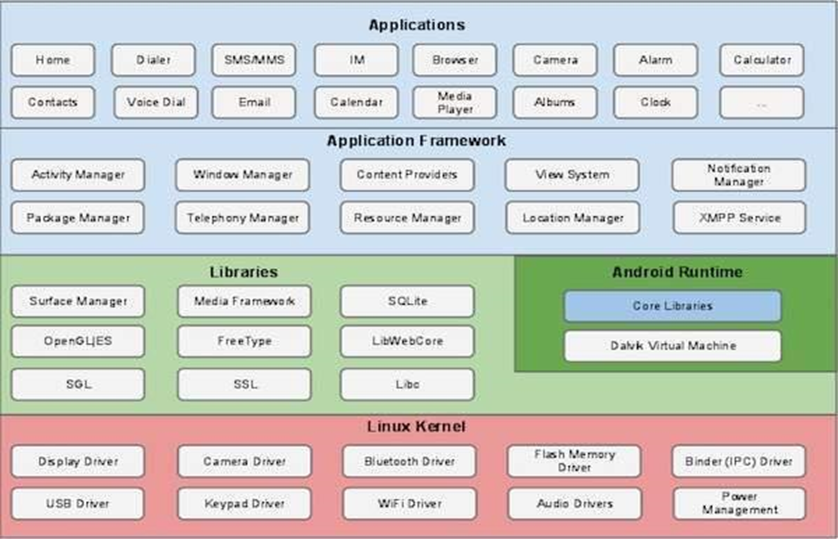
* Hoàn thành xong ứng dụng TransferData chương trình thực hiện được các yêu cầu đề ra của đề tài tốt nghiệp
* Ứng dựng TransferData đã phần nào xây dựng và đáp ứng một số chức năng chính:
* Tìm và kết nối 2 thiết bị
* Backup dữ liệu (Danh bạ, tin nhắn, ảnh…)
* Truyền dữ liệu giữa thiết bị
* Mã hoá và giải mã dữ liệu
* Khôi phục dữ liệu

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## Các thành phần cơ bản trong android

### Cấu trúc của ứng dụng android

Hệ điều hành Android là một ngăn xếp của các thành phần phần mềm mà có thể đại khái phân chia thành 5 khu vực và 4 lớp chính. Hình dưới đây minh họa sơ đồ cấu trúc Android.



Hình 1.1.1‑1 Cấu trúc của ứng dụng Android

### Lớp Linux Kernel

Linux Kernel là lớp thấp nhất. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình, … Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.

* Libraries

Phía trên Linux Kernel là tập hợp các bộ thư viện mã nguồn mở WebKit, bộ thư viện nổi tiếng libc, cơ sở dữ liệu SQLite hữu ích cho việc lưu trữ và chia sẻ dữ liệu, bộ thư viện thể phát, ghi âm về âm thanh, hoặc video. Thư viện SSL chịu trách nhiệm cho bảo mật Inter Linux Kernel là lớp thấp nhất. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình, … Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.

* Libraries

Phía trên Linux Kernel là tập hợp các bộ thư viện mã nguồn mở WebKit, bộ thư viện nổi tiếng libc, cơ sở dữ liệu SQLite hữu ích cho việc lưu trữ và chia sẻ dữ liệu, bộ thư viện thể phát, ghi âm về âm thanh, hoặc video. Thư viện SSL chịu trách nhiệm cho bảo mật Internet.

* Android Libraries

Phần này gồm các thư viện dựa trên Java. Nó bao gồm các Framework Library giúp xây dựng, vẽ đồ họa và truy cập cơ sở dữ liệu trở nên dễ dàng hơn. Dưới đây là một số Android Library cốt lõi có sẵn cho lập trình viên Android:

* android.app − Cung cấp truy cập tới mô hình ứng dụng và nó là nền móng cho tất cả ứng dụng Android.
* android.content − Việc truy cập nội dung, các thông điệp giữa các ứng dụng và các thành phần ứng dụng trở nên dễ dàng hơn.
* android.database − Được sử dụng để truy cập dữ liệu được công bố bởi Provider và bao gồm các lớp quản lý cơ sở dữ liệu SQLite.
* android.opengl − Một Java Interface cho OpenGL ES 3D thông qua API.
* android.os − Cung cấp cho các ứng dụng sự truy cập tới các dịch vụ chuẩn của hệ điều hành như thông báo, dịch vụ hệ thống và giao tiếp nội tiến trình.
* android.text − Được sử dụng để phục hồi và thao tác text trên một thiết bị hiển thị.
* android.view − Các khối kiến trúc nền tảng của ứng dụng UI.
* android.widget − Một tập hợp các UI được xây dựng trước như button, label, list view, layout manager, radio button, …
* android.webkit − Một tập hợp các lớp cho phép khả năng để trình duyệt trên web được xây dựng bên trong các ứng dụng.
* Android Runtime

Đây là thành phần thứ 3 trong cấu trúc, thuộc về lớp 2 tính từ dưới lên. Phần này cung cấp một thành phần quan trọng gọi là Dalvik Virtual Machine là một máy ảo Java đặc biệt, được thiết kế tối ưu cho Android.

Máy ảo Dalvik VM sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ, đa luồng, mà thực chất là bên trong ngôn ngữ Java. Máy ảo Dalvik cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó.

Android Runtime cũng cung cấp bộ thư viện cốt lõi, cho phép các lập trình viên Android sử dụng để viết các ứng dụng Android.

* Application Framework

Lớp Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong các lớp Java. Các lập trình viên cũng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.

Application Framework bao gồm các dịch vụ chính sau:

* Activity Manager − Điều khiển các khía cạnh của vòng đời ứng dụng và Activity Stack.
* Content Providers − Cho phép các ứng dụng công bố và chia sẻ dữ liệu với các ứng dụng khác.
* Resource Manager − Cung cấp sự truy cập tới các resource được nhúng (không phải code) như chuỗi, thiết lập màu, UI layout.
* Notifications Manager − Cho phép các ứng dụng hiển thị thông báo tới người dùng.
* View System − Một tập hợp các view được sử dụng để tạo UI cho ứng dụng.
* Aplication

Bạn sẽ thấy tất cả các ứng dụng Android ở lớp trên cùng. Ứng dụng bạn viết sẽ được cài đặt vào lớp này.

Ví dụ của những ứng dụng này là Contacts, Books, Browser, Games, …net.

* Android Libraries

Phần này gồm các thư viện dựa trên Java. Nó bao gồm các Framework Library giúp xây dựng, vẽ đồ họa và truy cập cơ sở dữ liệu trở nên dễ dàng hơn. Dưới đây là một số Android Library cốt lõi có sẵn cho lập trình viên Android:

* android.app − Cung cấp truy cập tới mô hình ứng dụng và nó là nền móng cho tất cả ứng dụng Android.
* android.content − Việc truy cập nội dung, các thông điệp giữa các ứng dụng và các thành phần ứng dụng trở nên dễ dàng hơn.
* android.database − Được sử dụng để truy cập dữ liệu được công bố bởi Provider và bao gồm các lớp quản lý cơ sở dữ liệu SQLite.
* android.opengl − Một Java Interface cho OpenGL ES 3D thông qua API.
* android.os − Cung cấp cho các ứng dụng sự truy cập tới các dịch vụ chuẩn của hệ điều hành như thông báo, dịch vụ hệ thống và giao tiếp nội tiến trình.
* android.text − Được sử dụng để phục hồi và thao tác text trên một thiết bị hiển thị.
* android.view − Các khối kiến trúc nền tảng của ứng dụng UI.
* android.widget − Một tập hợp các UI được xây dựng trước như button, label, list view, layout manager, radio button, …
* android.webkit − Một tập hợp các lớp cho phép khả năng để trình duyệt trên web được xây dựng bên trong các ứng dụng.
* Android Runtime

Đây là thành phần thứ 3 trong cấu trúc, thuộc về lớp 2 tính từ dưới lên. Phần này cung cấp một thành phần quan trọng gọi là Dalvik Virtual Machine là một máy ảo Java đặc biệt, được thiết kế tối ưu cho Android.

Máy ảo Dalvik VM sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ, đa luồng, mà thực chất là bên trong ngôn ngữ Java. Máy ảo Dalvik cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó

Android Runtime cũng cung cấp bộ thư viện cốt lõi, cho phép các lập trình viên Android sử dụng để viết các ứng dụng Android.

* Application Framework

Lớp Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong các lớp Java. Các lập trình viên cũng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.

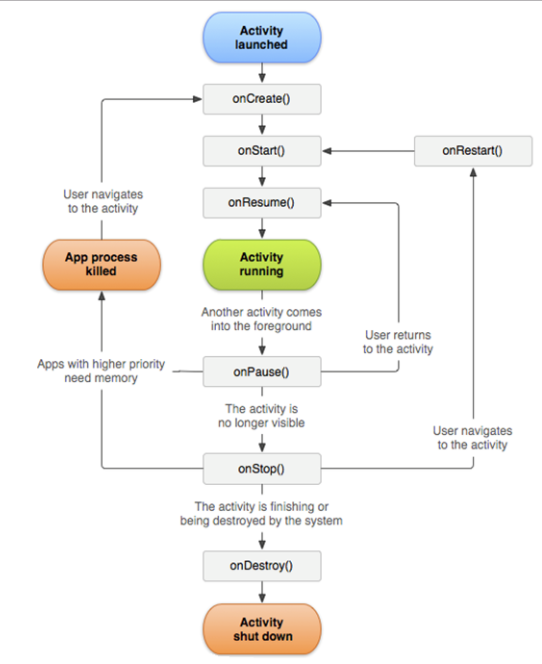
Application Framework bao gồm các dịch vụ chính sau:

* Activity Manager − Điều khiển các khía cạnh của vòng đời ứng dụng và Activity Stack.
* Content Providers − Cho phép các ứng dụng công bố và chia sẻ dữ liệu với các ứng dụng khác.
* Resource Manager − Cung cấp sự truy cập tới các resource được nhúng (không phải code) như chuỗi, thiết lập màu, UI layout.
* Notifications Manager − Cho phép các ứng dụng hiển thị thông báo tới người dùng.
* View System − Một tập hợp các view được sử dụng để tạo UI cho ứng dụng.
* Aplication

Bạn sẽ thấy tất cả các ứng dụng Android ở lớp trên cùng. Ứng dụng bạn viết sẽ được cài đặt vào lớp này.

Ví dụ của những ứng dụng này là Contacts, Books, Browser, Games, …

### Vòng đời của Activity



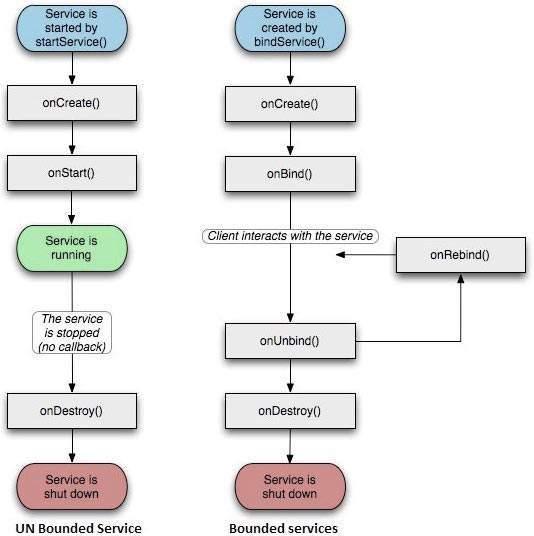
Hình 3‑1 Vòng đời của Activity

### Service

Một Service là một thành phần mà chạy trong Background để thực hiện các hoạt động mà không cần tương tác với người dùng và nó làm việc ngay cả khi nếu ứng dụng bị hủy. Về cơ bản, một Service nhận hai trạng thái:

|  |  |
| --- | --- |
| Trạng thái | Mô tả |
| Started | Một Service được bắt đầu (started) khi một Component, chẳng hạn như activity, bắt đầu bằng việc gọi ***startService***(). Khi đã được bắt đầu, một Service có thể chạy ngầm định trong Background, ngay cả khi thành phần mà bắt đầu nó đã bị hủy |
| Bound | Một Service được gắn kết (bound) khi một Component kết nối tới nó bởi việc gọi ***bindService****()*. Một bound service cung cấp một giao diện Client-Server cho phép các thành phần để tương tác với Service đó, gửi yêu cầu, nhận kết quả, và có thể thực hiện một số tiên trình khác với Interprocess Communication (IPC) |

Một Service có các phương thức callback mà bạn có thể triển khai để giám sát các thay đổi trong trạng thái của Service và bạn có thể thực hiện công việc trên các giai đoạn cụ thể. Sơ đồ bên trái minh họa vòng đời của Service khi Service được tạo với phương thức startService() và sơ đồ bên phải minh họa vòng đời Service khi Service được tạo với bindService():



Hình 4‑1 Service

Để tạo một Service, bạn tạo một lớp Java mà kế thừa lớp cơ sở Service hoặc một trong các lớp con đang tồn tại của nó. Lớp cơ sở Service định nghĩa các phương thức callback đa dạng và các phương thức quan trọng nhất được liệt kê dưới đây. Bạn không cần triển khai toàn bộ chúng. Điều quan trọng là bạn hiểu từng phương thức và cách triển khai chúng để đảm bảo ứng dụng của bạn vận hành như người dùng mong đợi

|  |  |
| --- | --- |
| **Callback** | **Mô tả** |
| onStartCommand() | Hệ thống gọi phương thức này khi thành phần khác, chẳng hạn một activity, yêu cầu rằng Service bắt đầu, bằng việc gọi ***startService****()*. Nếu bạn triển khai phương thức này, thì bạn cũng cần dừng nó khi công việc đã được thực hiện, bằng cách gọi phương thức ***stopSelf****()* hoặc ***stopService****()* |
| onBind() | Hệ thống gọi phương thức này khi thành phần khác muốn liên kết với Service bởi gọi ***bindService****()*. Nếu bạn triển khai phương thức này, bạn phải cung cấp một giao diện mà client sử dụng để giao tiếp với Service đó, bằng việc trả về một đối tượng ***IBinder***. Bạn phải luôn luôn triển khai phương thức này, nếu bạn không muốn cho phép liên kết, thì bạn nên trả về *null*. |
| onUnbind() | Hệ thống gọi phương thức này khi tất cả client đã bị ngắt kết nối với một giao diện nào đó đã được công bố bởi Service này |
| onRebind() | Hệ thống gọi phương thức này khi client mới đã kết nối tới Service này, sau khi nó đã được thông báo trước đó rằng tất cả đã bị ngắt kết nối trong phương thức ***onUnbind(Intent)*** của nó |
| onCreate() | Hệ thống gọi phương thức này khi Service là được tạo đầu tiên bởi sử dụng ***onStartCommand****()* hoặc ***onBind****()*. Lời gọi này là cần thiết để thực hiện cài đặt one-time |
| onDestroy() | Hệ thống gọi phương thức này khi Service không còn được sử dụng nữa và đang bị hủy. Service của bạn nên triển khai phương thức này để xóa bất cứ Resource nào như thread, registered listener, receiver, ... |

### Broadcast Receiver

**Broadcast Receiver** phản hồi các thông báo phát ra từ các ứng dụng khác hoặc từ chính hệ thống. Những thông báo này đôi khi được gọi là các event hoặc intent. Ví dụ, các ứng dụng cũng có thể khởi tạo các tín hiệu broadcast để thông báo cho ứng dụng khác biết rằng một số dữ liệu đã được về tới thiết vị và là có sẵn cho chúng để sử dụng, vì thế Broadcast Receiver thông dịch thông tin đó và khởi tạo hành động thích hợp.

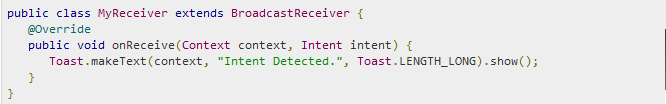
Sau đây là hai bước quan trọng để làm Broadcast Receiver làm việc cho các Intent:

* Tạo Broadcast Receiver.
* Đăng ký Broadcast Receiver.

Có thể có một bước bổ sung nếu bạn đang triển khai các Custom Intent của bạn, thì khi đó bạn sẽ phải tạo và phát các Intent đó

* **Tạo Broadcast Receiver trong Android**

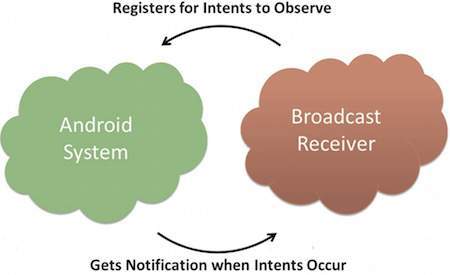
Một Broadcast Receiver được triển khai như là một lớp con của lớp **BroadcastReceiver** và ghi đè phương thức onReceive(), nơi mà mỗi thông báo được nhận như là một tham số của đối tượng Intent.



Hình 5‑1 Broadcast Receiver

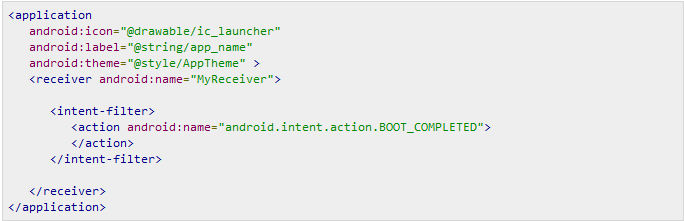
* **Đăng ký Broadcast Receiver trong Android**

Một ứng dụng nghe các Intent được phát ra cụ thể bằng cách đăng ký một Broadcast Receiver trong AndroidManifest.xml file. Giả sử chúng ta đang đăng ký MyReceiver cho system event (sự kiện được tạo từ hệ thống) đã được tạo là ACTION\_BOOT\_COMPLETED, mà được kích hoạt bởi hệ thống một khi hệ điều hành Android đã hoàn thành tiến trình boot.



Hình 5‑2 Đăng kí Broadcast Receiver trong android

* **Broadcast-Receiver**



Bây giờ, bất cứ khi nào thiết bị Android của bạn được boot, thì thông báo sẽ được nhận bởi *MyReceiver* và trình triển khai logic bên trong phương thức **onReceive()** sẽ được thực thi.

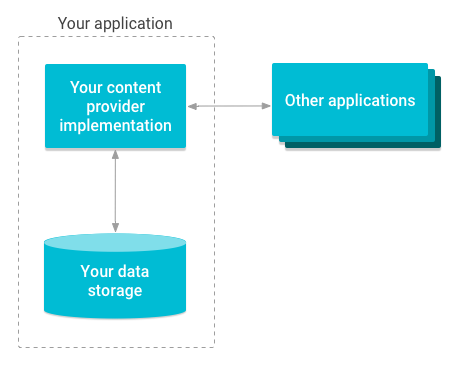
Có một số system event được định nghĩa là là các trường final static trong lớp **Intent**. Bảng dưới liệt kê một số system event quan trọng:

|  |  |
| --- | --- |
| **Event** | **Miêu tả** |
| android.intent.action.BATTERY\_CHANGED | Thông báo này chứa trạng thái nạp, mức độ, và thông tin khác về pin |
| android.intent.action.BATTERY\_LOW | Chỉ trạng thái low battery trên thiết bị |
| android.intent.action.BATTERY\_OKAY | Chỉ rằng pin bây giờ là tốt sau khi low battery |
| android.intent.action.BOOT\_COMPLETED | Đây là tín hiệu broadcast thông báo sau khi hệ thống đã kết thúc boot |
| android.intent.action.BUG\_REPORT | Chỉ activity để báo cáo một bug |
| android.intent.action.CALL | Thông báo một lời gọi tới ai đó được xác định bởi dữ liệu |
| android.intent.action.CALL\_BUTTON | Người dùng nhấn nút call để tới Dialer (trình gọi điện) hoặc giao diện UI thích hợp khác để tạo một cuộc gọi |
| android.intent.action.DATE\_CHANGED | Date đã được thay đổi |
| android.intent.action.REBOOT | Reboot thiết bị |

## Content providers

### Khái niệm

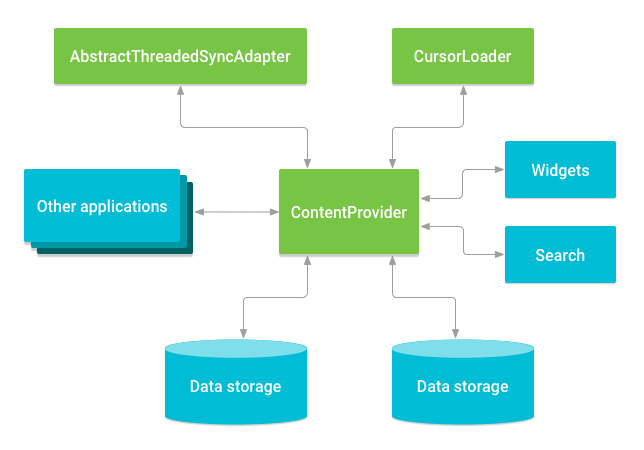
Content provider là một thành phần để quản lý truy cập dữ liệu, nó cung cấp các phương thức khác nhau để các ứng dụng có thể truy cập dữ liệu từ một ứng dụng khác bằng cách sử dụng ContentResolver. Content Provider có thể giúp cho một ứng dụng quản lý quyền truy cập đến dữ liệu được lưu bởi ứng dụng đó, hoặc các ứng dụng khác, và đó là một cách để ta có thể chia sẻ dữ liệu cho các ứng dụng khác nhau. Hình dưới đây biểu diễn cho việc cách content providers quản lý việc truy cập tới bộ nhớ.



Hình 1‑1 Content Provider

Content Provider điều phối việc truy cập tới bỗ lưu trữ dữ liệu thông qua các API và các component như hình dưới, nó bao gồm:

* Chia sẻ dữ liệu từ ứng dụng của bán tới các ứng dụng khác
* Gửi dữ liệu sang widget
* Trả về một kết quả gợi ý khi search cho ứng dụng của bạn thông qua Seach Framework sử dụng SearchRecentSuggestionsProvider
* Đồng bộ dữ liệu của ứng dụng với server bằng cách sử dụng AbstractThreadedSyncAdapter
* Tải dữ liệu lên UI sử dụng CursorLoader



Hình 1‑2 Tải dữ liệu lên UI sử dụng CursorLoader

Content Provider hoạt động rất giống với một cơ sở dữ liệu, bạn có thể truy vấn, chỉnh sửa nội dung, cũng như là thêm xóa các nội dung sử dụng các phương thức: insert(), update(), delete(), query().

### Cách sử dụng Content Provider

Để sử dụng Content Provider làm theo các bước sau:

* Xác định kiểu dữ liệu
* Xác định Uniform Resource Indentifier (URI)
* Khai báo Content Provider trong manifest
* Implement lớp ContentProvider và các phương thức được yêu cầu

Các phương thức cần được Override trong lớp Content Provider:

* onCreate(): Phương thức này được gọi khi Provider được bắt đầu, nếu quá trình khởi tạo thành công trả về true, ngược lại là false
* query(): Phương thức nhận yêu cầu từ Client. Kết quả được trả về như một đối tượng Cursor.
* insert(): Phương thức chèn một dòng dữ liệu mới vào Content Provider.
* delete(): Phương thức xóa một dòng dữ liệu đã tồn tại.
* update(): Phương thức cập nhật một dòng dữ liệu nào đó đã tồn tại.
* getType(): Phương thức trả về kiểu MIME của dữ liệu tại các URI.

### Content URI

Content URI là một URI định danh dữ liệu trong một provider. Content URI bao gồm kí hiệu tên của toàn bộ provider và một tên chỉ tới một bảng. Khi bạn gọi một phương thức truy cập tới bảng trong provider thì Content URI của bảng đó là sẽ là một tham số. Để truy vấn data qua provider, ta sẽ sử dụng URI có định dạng như sau:

**content://authority/path/id**

* content: luôn là content://
* authority: một xâu để xác định tên của Content Provider, ví dụ như contact, browser,...
* path: có thể không có hoặc được chia làm nhiều phần, và được phân cách bằng gạch chéo "/", dùng để định ra các thành phần con của dữ liệu. Ví dụ như để lấy danh sách các liên hệ trong danh bạ thì URI sẽ là content://contacts/people.
* id: chỉ định rõ một bản ghi trong tập hợp dữ liệu, mỗi bản ghi sẽ được đánh dấu id là một số duy nhất

## 1.3 Peer-To-Peer Và Client-Server

### Client – Server là gì?

**Client – Server** được biết đến với tên gọi **máy khách và máy chủ**. Server là một máy chủ web, đây là nơi lưu trữ nội dung của một trang web, và máy tính của bạn được gọi là Client, nó sẽ lấy thông tin từ Server để hiển thị lại trên máy tính của bạn. Đó là cách mà Internet truyền thống sử dụng.

Mỗi Server độc lập có thể lưu trữ hàng trăm nghìn dữ liệu khác nhau và chịu tải hàng trăm Client cùng lúc, nhưng đó cũng chính là vấn đề của mô hình Client – Server, bởi chúng phụ thuộc rất nhiều vào phần cứng của Server.

Thứ nhất, Server phụ thuộc rất nhiều vào phần cứng. Sẽ không có vấn đề gì nếu Server của bạn chỉ lưu trữ mỗi một nội dung nhất định, nhưng thực tế thì mỗi Server phải xử lí hàng trăm yêu cầu khác nhau từ mỗi Client, chẳng hạn như Toonima – bạn muốn đọc bài A thì Server phải xử lí và cho ra nội dung bài A, một bạn khác muốn đọc bài B thì Server phải xử lí và cho ra nội dung bài B – mỗi lần như thế sẽ khiến Server sử dụng một phần CPU và RAM để xử lí, Server sẽ cần nâng cấp nếu khả năng xử lí không đủ đáp ứng nhu cầu của Client. Đó chính là giới hạn về phần cứng của Server.

Thứ hai, mỗi Client sử dụng một phần nhỏ đường truyền mạng của Server, ví dụ: Server của bạn có đường truyền mạng khoảng 100MB/s, tức nếu bạn có 100 Client cùng sử dụng thì mỗi người sẽ có 1 MB/s để dùng, nhưng nếu có khoảng 1000 Client thì mỗi người chỉ còn 100KB/s mà thôi. Đấy chính là giới hạn về đường truyền của Server.

Và cuối cùng, đó chính là **giá thành của dung lượng**, nếu có 1 file nặng khoảng 1MB và được 1000 người tải xuống thì tổng dung lượng của dữ liệu được tải xuống sẽ là 1GB. Bạn sẽ không phải trả tiền cho 1MB dữ liệu này, nhưng nhà quản trị web phải trả số tiền đó, thử nghĩ 1 bộ phim nặng khoảng 1GB và có khoảng 10.000 người tải xuống thì số tiền họ phải trả là bao nhiêu? Giá thành quá đắt chính là giới hạn của mô hình Server – Client.

Vì 3 giới hạn trên của mô hình Client và Server mà người ta nghĩ ra một mô hình khác, tiết kiệm hơn và gọi nó là Peer to Peer, hay còn gọi là mạng ngang hàng, mạng đồng đẳng.

### Peer to peer (P2P) là gì?

Peer to Peer là mô hình phát triển dựa trên mô hình Client-Server, mô hình mới này sẽ giúp mọi Client trở thành Server. Và người ta chỉ cần một Server điều khiển hoạt động cho từng Client khác nhau. Với Peer to Peer, Client không chỉ lấy dữ liệu mà còn có thể chia sẻ lại dữ liệu này với Client khác và việc chia sẻ lại dữ liệu này người ta gọi nó là seeding. Việc seeding rất quan trọng trong sự phát triển của Peer to Peer, nếu không có seeding thì Peer to Peer chỉ giống như mô hình client – server mà thôi.

Không giống như Client – Server – hiệu suất của mọi thứ sẽ giảm đi khi có quá đông Client tham gia vào mạng. Hiệu suất của Peer to Peer sẽ tăng lên nếu số Client trong mạng càng đông. Tức càng có nhiều người seeding một dữ liệu thì người chưa có dữ liệu đó sẽ càng có cơ hội nhận được dữ liệu muốn download.

Trong Peer to Peer, tốc độ download tệp tin của bạn sẽ càng nhanh nếu có nhiều người seeding tệp tin đó cho bạn, hãy tưởng tượng tệp tin bạn cần download nặng 1GB, bạn có 1 người seeding với tốc độ 1MB/s thì sẽ chỉ mất 1000 giây để tải xong tệp tin, nhưng nếu có 10 người như thế thì bạn sẽ chỉ mất 100 giây cho việc download tệp tin. Nhưng tính năng chia nhỏ tệp tin để download như thế này không phải phần mềm Peer to Peer nào cũng có, BitTorrent chính là phần mềm đầu tiên sử dụng tính năng này.

Trong Peer to Peer, Server sẽ dùng để điều khiển hoạt động của các Client như: Server sẽ có thông tin về những máy đang kết nối trong hệ thống, các file đang tồn tại trong máy của Client…, từ đó phân chia công việc cho hợp lí giữa các Client. Việc chia sẻ file bây giờ sẽ dựa vào các client với nhau.

### Sự khác nhau

Client Server là mô hình máy khách - máy chủ, có thể truy nhập mạng hay tài nguyên mạng từ các máy chủ. Đây là giải pháp khắc phục mạng quá tải và phá vỡ sự ngăn cách trong cấu trúc vật lý cũng như hệ điều hành khác nhau trên mạng.

So sánh giữa Peer to Peer và Client Server:

|  |  |  |
| --- | --- | --- |
| Mô hình mạng | Client-Server | Peer-to-peer |
| Chỉ tiêu đánh giá |
| Độ an toàn và tính bảo mật thông tin | Có độ an toàn và bảo mật thông tin cao nhất. Quản trị mạng có thể điều chỉnh quyền truy nhập thông tin | Độ an toàn và bảo mật kém, phụ thuộc vào mức truy nhập được chia sẻ |
| Khả năng cài đặt | Khó cài đặt | Dễ cài đặt |
| Đòi hỏi phần cứng và phần mềm | Đòi hỏi có máy chủ, hệ điều hành mạng và các phần cứng bổ sung | Không cần máy chủ, hệ điều hành mạng, phần cứng bổ sung rất ít |
| Quản trị mạng | Phải có quản trị mạng | Không cần quản trị mạng |
| Xử lý và lưu trữ tập trung | Có | Không |
| Chi phí cài đặt | Cao | Thấp |

### Ưu điểm

* Tất cả các máy khi tham gia mạng lưới đều có thể đóng góp thông tin bao gồm băng thông, dữ liệu và cả khả năng tính toán. Càng nhiều máy tham gia thì lượng thông tin cung cấp cho người dùng càng nhiều.
* Nhờ tính chất phân tán giúp các mạng khác vẫn hoạt động tốt khi một máy trong mạng lưới gặp sự cố.
* Máy tính trong hệ thống đóng vai trò như máy chủ và máy khách
* Phần mềm sử dụng dễ dàng và được tích hợp tốt
* Phần mềm có nhiều công cụ hỗ trợ người dùng và các chức năng khác nhau
* Phần mềm hỗ trợ các giao thức qua mạng như SOAP hoặc XML-RPC

## 1.4 Wifi-Direct

### Khái niệm

**Wi-Fi Direct là công nghệ** không dây cho phép các thiết bị Wi-Fi Direct thiết lập kết nối Wi-Fi trực tiếp mà cả hai có thể gửi và nhận tệp **và 2 thiết bị có thể dễ kết nối (hàng ngang) với nhau mà không cần trải qua nhiều bước cài đặt, không cần tới bộ định tuyến độc lập (Router)**. Điều này có nghĩa là người dùng khi di chuyển có thể in tài liệu, chia sẻ dữ liệu, đồng bộ hóa tệp và hiển thị thông tin từ máy tính xách tay và netbook, cũng như từ các thiết bị như điện thoại thông minh và máy tính bảng. Các sản phẩm được chứng nhận Wi-Fi Direct có thể kết nối với các thiết bị không dây khác mà không cần tham gia mạng không dây truyền thống hoặc điểm truy cập Wi-Fi.

Công dụng của **Wi-Fi Direct** là giúp 2 thiết bị (điện thoại, laptop, máy tính bảng,...) kết nối với nhau và truyền tải dữ liệu với tốc độ cao, dung lượng lớn (phim, nhạc, hình ảnh, file, các tập tin khác) mà không cần dây kết nối. Bạn có thể kết nối điện thoại với TV để gửi ảnh gia đình từ kỳ nghỉ gần nhất hoặc gửi hợp đồng trực tiếp tới máy tính bảng để lấy chữ ký – tương tự như kết nối Bluetooth và cho phép truyền tệp mà không cần Wi-Fi kết nối mạng hoặc Bluetooth. Một số thiết bị sẽ sử dụng Bluetooth ban đầu và chuyển sang Wi-Fi trực tiếp khi kích thước tệp đủ lớn và cả hai thiết bị đều cho phép kết nối.

Hiện nay công nghệ Wi-Fi Direct giữa các hãng sản xuất chưa đồng bộ với nhau, nên chỉ dùng trực tiếp cho những máy cùng hãng sản xuất. Nếu muốn sử dụng các máy [điện thoại](http://www.thegioididong.com/dtdd" \t "_blank" \o "điện thoại di động) khác hãng thì bạn phải cài đặt thêm phần mềm nhưng hỗ trợ chưa thực sự tốt lắm.

### So sánh Wifi-Direct, Bluetooth, NFC

Đây là các chuẩn giao tiếp pear-to-pear thường thấy trên điện thoại (kể cả other phone vì nó dùng bluetooth) trên mặt lý thuyết.

* Wifi-Direct
* Ưu điểm:

Giao tiếp nhanh như wifi chuẩn n bình thường (có thể phát video hd) và tốc độ có thể lên 250 mb/s. Tầm giao tiếp cao (tối đa tới 200 mét theo wifi alliance). Nó có ưu điểm như wifi thường như vậy vì nó được phát triển bởi Liên minh Wifi và đã nhận được Wifi Certified. Có thể thay thế cho bluetooth và nfc vì tốc độ cao và tầm phủ sóng rộng. Bảo mật cao vì trang bị chuẩn WPA2 với khả năng mã hoá AES 256 bit. Mức tiêu thụ điện Wi-Fi Direct thấp vì chương trình WMM Power Save cho phép cải thiện thời lượng pin của thiết bị từ 15-40% so với hiện nay

* Nhược điểm:

Ít thiết bị hỗ trợ. Giá thành đầu cuối khá đắt. Thời gian thiết lập kết nối khá lâu vì thủ tục rườm rà.

* Bluetooth
* Ưu điểm:

Nhiều điện thoại hỗ trợ. Giá thành rẻ. Tầm giao tiếp rộng. Tiêu thụ ít điện năng. Bảo mật thấp hơn Wifi direct một chút nhưng vẫn ngon cho người dùng thông thường vì Bluetooth 4.0 với khả năng hỗ trợ AES 128bit nhưng có thời gian kết nối nhanh hơn cũng rất lý tưởng cho những phiên kết nối ngắn, điều thường thấy trên các thiết bị di động.

* Nhược điểm:

Phạm vi kết nối ngắn. Dễ bị nhiễu bởi các thiết bị điện tử khác như lò vi ba, ti vi, tủ lạnh... Bắt sóng kém khi có vật cản. Thời gian thiết lập kết nối khá lâu vì thủ tục rườm rà.

* NFC
* Ưu điểm:

Dễ dàng truy cập nội dung số VD: thấy ap-phích quảng cáo hỗ trợ nfc thì dí điện thoại nfc dô thì các thông tin thêm về sản phẩm sẽ hiện lên màn hình điện thoại). Thời gian thiết lập kết nối nhanh. Không bị ảnh hưởng và không gây ảnh hưởng tới wifi, bluetooth... do nfc hoạt động trên tầng số 13.56 MHz

* Nhược điểm:

Hiện tại vẫn còn khá ít thiết bị đầu cuối. Tốc độ tối đa hiện nay là 424 Kbps/s. Chưa hỗ trợ mã hoá cao (chỉ khi có thông tin nhạy cảm thì chỉ mới dừng lại ở 2 chuẩn mã hoá là chuẩn mã hóa tiên tiến (Advanced Encryption Standard – AES) là thuật toán mã hóa khối được Mỹ áp dụng làm tiêu chuẩn mã hóa; và chuẩn mã hóa dữ liệu Triple DES (Data Encryption Standard) nghĩa là một thông tin được mã hóa DES 3 lần với 3 khóa khác nhau, do đó chiều dài mã hóa sẽ lớn hơn và an toàn hơn.

* Kết luận:

Nếu muốn chia sẻ dữ liệu đơn giản như file nhạc, video bình thường thì bluetooth là tốt nhất vì giá bình dân, đủ cho nhu cầu bình thường nhưng bạn nên xem xét lại khi dùng bluetooth vì nó khá bị ảnh hưởng bởi nhiễu và đặc biệt cực kém khi ở nơi có nhiều tường hoặc vật cản. Nếu muốn chia sẻ tốc độ cao, ở môi trường bảo mật nhiều và đặc biệt là chia sẻ nhưng file nặng hoặc cực nặng và bạn không quan tâm tới thiết bị đầu cuối hỗ trợ thì Wifi Direct sẽ là lựa chọn tối ưu cho bạn. Còn đối với NFC thì là một công nghệ mang tính di động và tiện lợi cực cao và bạn không quan tâm tới mã hoá và tốc độ vì đơn giản bạn có thể check-in, xem áp-phích, thanh toán điện tử..., những công việc ít khi mã hoá và truyền tải một lượng thông tin khá ít, vừa đủ cho NFC.

### Cách hoạt động

Wi-Fi Direct sẽ sử dụng nhiều chuẩn công nghệ khác để thực hiện nhiệm vụ của mình:

* Wi-Fi: Wi-Fi Direct sử dụng công nghệ Wi-Fi quen thuộc, vốn thường được sử dụng để kết nối máy vi tính và các thiết bị di động với các bộ định tuyến. Nếu có hỗ trợ Wi-Fi Direct, thiết bị của bạn sẽ có thể hoạt động như một bộ định tuyến thực thụ: Các thiết bị khác có thể kết nối trực tiếp tới mạng Wi-Fi do thiết bị này “làm chủ”. Bạn có thể thực hiện kết nối ngang hàng (ad-hoc) giữa các thiết bị có hỗ trợ Wi-Fi với nhau, song kết nối ad-hoc tương đối phức tạp và khó sử dụng so với Wi-Fi Direct.
* Wi-Fi Direct Device and Service Discovery (Giao thức Phát hiện Thiết bị và Dịch vụ qua Wi-Fi Direct): giao thức này cho phép các thiết bị tìm và nhận diện các dịch vụ mạng và các thiết bị khác có hỗ trợ Wi-Fi Direct. Ví dụ, smartphone Galaxy S4 của bạn có thể tìm thấy danh sách tất cả các thiết bị thông minh có hỗ trợ Wi-Fi Direct ở gần, tất cả các máy in có hỗ trợ Wi-Fi v…v…
* Wi-Fi Protected Setup: Khi 2 thiết bị kết nối với nhau, chúng sẽ trải qua quá trình cài đặt có tên gọi WPS (Wi-Fi Protected Setup).
* WPA2: Wi-Fi Direct sử dụng mã hóa WPA2, loại mã hóa Wi-Fi mạnh nhất.
* Wi-Fi Direct cũng có thể được gọi tên Wi-Fi P2P (peer-to-peer), do đây là một kết nối ngang hàng. 2 thiết bị có hỗ trợ Wi-Fi Direct sẽ kết nối không cần thông qua router.

# CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ ỨNG DỤNG

## Khảo sát hệ thống

Hiện nay công nghệ phát triển các nhà sản xuất điện thoại ngày càng cho ra nhiều mẫu điện thoại đẹp hơn và nhiều tính năng hơn. Vì vậy nhu cầu nâng cấp điện thoại cũ lên điện thoại mới cũng ngày càng nhiều. Việc chuyển dữ liệu trên thiết bị cũ và thiết bị mới là rất cần thiết. Nó giúp người dùng sử dụng điện thoại mới giống như sử dụng trên chiếc điện thoại mới mà không cần phải tải và cài đặt gì nhiều.

Trên Google Store có rất nhiều ứng dụng giúp người dùng chuyển dữ liệu giữ hai thiết bị. Những ứng dụng này chia là 2 nhóm:

Nhóm một ứng dụng của các hãng sãn xuất điện thoại. Hạn chế của ứng dụng là các nhà sản xuất chỉ cho phép truyền dữ liệu một chiều, tức là chuyển dữ liệu từ máy khác sang thiết bị do công ty họ sản xuất mà không cho phép điều ngược lại

Nhóm hai là ứng dụng không phải của các nhà sản xuất điện thoại mà do lập trình viên tạo nên. Những ứng dụng này đa số làm từ rất lâu nên giao diện đã lỗi thời. Một số ứng dụng đã ngừng cập cập nên nó không thể hoạt động trên hệ điều hành androi mới nhất.

Chính vì những lý do trên nên em đã xây dựng ứng dụng “Transfer Data”. Ứng dụng sẽ hỗ trợ người dùng truyền dữ liệu giữa 2 thiết bị một cách nhanh chóng và tiện lợi.

## Phân tích hệ thống

### 2.2.1 Mô tả bài toán

Khi người dùng mua một chiếc điện thoại mới và muốn lấy lại toàn bộ dữ liệu quan trọng trong chiếc điện thoại cũ của mình thì họ có rất nhiều cách để có thể lấy lại dữ liệu như: chuyển qua internet, đồng bộ... Nhưng những cách chuyển này đều phụ thuộc vào internet.

Hiện nay, có 3 loại kết nối giúp chuyển dữ liệu mà không cần đến internet:

* Bluetooth
* NFC
* Wifi Driect

Trong đó thì bluetooth và NFC tốc độ chậm và còn nhiều hạn chế hơn so với

Wifi direct. Vì vậy, em chọn kết nối Wifi để xẩy dựng ứng dụng chuyển dữ liệu.

Người dùng sẽ tải và cài đặt ứng dụng sau đó thực hiện kết nối 2 chiếc điện thoại cũ và mới với nhau. Sau khi kết nối xong, trên điện thoại cũ sẽ hiện ra danh sách các dữ liệu chuyển: danh bạ, tin nhắn... Ở danh sách này, người dùng có thể chọn dữ liệu cần chuyển bằng cách tích vào loại dữ liệu tương ứng và bấm Start để bắt đầu chuyển.

Ứng dụng sẽ lấy hết các dữ liệu mà người dùng đã chọn trên máy cũ ghi vào các file tương ứng với từng dữ liệu: message.xml, callog.xml... Sau đó, thực hiện mã hóa các file này bằng giải thuật AES và RSA rồi chuyển các file đó sang điện thoại mới qua Socket. Khi điện thoại mới nhận được các file dữ liệu sẽ thực hiện giải mã và restore vào máy.

### 2.2.2 Ý nghĩa của bài toán

Bài toán giúp cho người sử dụng tiết kiệm thời gian và công sức khi truyền sự liệu giữa hai thiết bị smartphone. Thay vì phải truyền dữ liệu thủ công như trước kia thì bây giờ mọi công việc thực hiện một cách tự động, các dữ liệu đều được mã hoá. Vì vậy người dùng không cần phải lo đến vấn đề bị dò rỉ thông tin trong quá trình truyền dữ liệu.

### 2.2.3 Tại sao đề tài này cần thiết

Vì các phương pháp truyền dữ liệu từ smartphone cũ sang smartphone mới hiện nay rất tốn thời gian và công sức của người dùng. Vì vậy em đã được hiện đề tài này nhắm tạo ra một phần mềm giúp cho việc chuyển dữ liệu tiện lợi và nhanh chóng hơn. Các tác vụ diễn ra hoàn toàn tự động sẽ giảm công sức và thời gian của người sử dụng

### 2.2.4 Xác định yêu cầu của hệ thống

Yêu cầu 1 :

* Ngôn ngữ sử dụng Java, máy ảo android, phần mềm android studio

Yêu cầu 2 : Phân tích yêu cầu và xây dựng úng dựng bao gồm

* Kết nối 2 điện thoại thông qua mạng ngang hàng p2p
* Backup dữ liệu của điện thoại cũ
* Mã hoá dữ liệu riêng tư
* Truyền dữ liệu sang điện thoại mới
* Restore dữ liệu

Yêu cầu 3 :

* Giao diện dễ sử dụng, có tính khoa học, thân thiện với người dùng
* Phối màu sắc , font chữ phù hợp
* Có tính đến ứng dụng rộng dãi

### 2.2.5 Xác định tác nhân của hệ thống

Người dùng: Sử dụng hệ thống để backup dữ kiệu và truyền dữ liệu đã backup sang thiết bị mới.

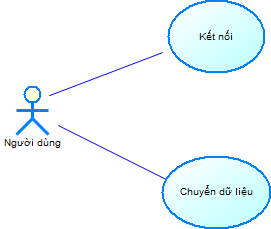
### 2.2.6 Chức năng chính của ứng dụng

* 1. **Tìm các thiết bị lân cận**
  + Chức năng này sẽ quét , tìm và hiển thị tên của tất cả thiết bị xung quanh có sử dụng kết nối p2p
  + Người dùng sẽ chọn tên thiết bị trong list để tiến hành kết nối 2 thiết bị với nhau
  1. **Kết nối 2 điện thoại**
  + Sau khi người dùng chọn tên thiết bị cần kết nói từ danh sách chức năng sẽ thực hiện kết nối 2 thiết bị với nhau thông qua mạng ngang hàng p2p
  1. **Chức năng lấy dữ trên điện thoại**
  + Chức năng sẽ lấy dữ liệu ( danh bạ , tin nhắn, nhật kí cuộc gọi, ứng dụng , file , ảnh , video)
  + Dữ liệu lấy xong sẽ được hiển thị lên cho người dùng lựa chọn dữ liệu cần chuyển
  1. **Chức năng mã hoá và giải mã**
  + Chức năng này sẽ mã hoá dữu liệu bằng thuật toán AES. Sau đó tiếp tục sử dụng thuật toán mã hoá RSA để mã hoá key AES
  1. **Chức năng truyền dữ liệu**
  + Chức năng này sẽ tạo kết nối socket giữa hai thiếu bị
  + Tất cả dữ liệu người dùng đã chọn sẽ được gửi qua thiết bị mới
  1. **Chức năng restore dữ liệu**
  + Sau khi nhận được dữ liệu từ điện thoại cũ chức năng sẽ tiến hành restore dữ liệu sang máy điện thoại mới .Sao cho dữ liệu còn nguyên vẹn và giống trên máy điện thoại cũ

## Thiết kế hệ thống

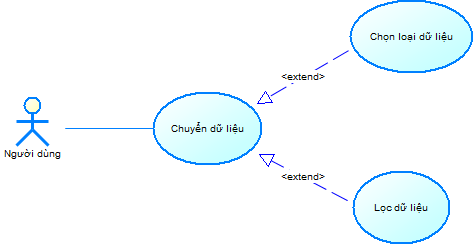
### Xây dựng biểu đồ use case

* **Biểu đồ Use Case cho tác nhân người dùng**

****

Hình 2.3.1‑1 Biểu đồ Use Case

* **Biều đồ phân rã Use Case cho tác nhân người dùng**

****

Hình 2.3.1‑2 Biểu đồ phân rã Use Case

Xây dựng kịch bản cho use case

* **Kịch bản cho UC “Kết nối”**

|  |  |
| --- | --- |
| Tên use case | Kết nối |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng phải cài đặt ứng dụng |
| Đảm bảo tối thiểu | Trở lại màn hình chờ kết nối |
| Đảm bảo thành công | Kết nối thành công 2 thiết bị |
| Kích hoạt | Bấm chọn loại thiết bị cũ hoặc mới |
| Chuỗi sự kiện chính:   1. Người dùng tải và cài đặt ứng dụng. 2. Người dùng chọn thiết bị cũ hoặc thiết bị mới. 3. Ứng dụng hiển thị danh sách các thiết bị xung quanh. 4. Người dùng chọn thiết bị cũ của mình. 5. Ứng dụng thông báo kết nối thành công trên thiết bị mới. 6. Ứng dụng hiển thị danh sách các dữ liệu có thể chuyển trên thiết bị cũ. | |

* **Kịch bản cho UC “Chuyển dữ liệu”**

|  |  |
| --- | --- |
| Tên use case | Chuyển dữ liệu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng phải kết nối thiết bị cũ và mới với nhau |
| Đảm bảo tối thiểu | Thông báo chuyển thất bại |
| Đảm bảo thành công | Chuyển dữ liệu từ thiết bị cũ sang thiết bị mới |
| Kích hoạt | Bấm nút “Start” để bắt đầu chuyển |
| Chuỗi sự kiện chính:   1. Người dùng kết nối thành công thiết bị cũ với thiết bị mới 2. Ứng dụng hiển thị danh sách các dữ liệu cần chuyển 3. Người dùng chọn nút “Start” trên màn hình. 4. Ứng dụng bắt đầu chuyển dữ liệu. 5. Ứng dụng restore dữ liệu trên máy mới. 6. Ứng dụng thông báo chuyển thành công. | |

* **Kịch bản cho UC “Chọn loại dữ liệu”**

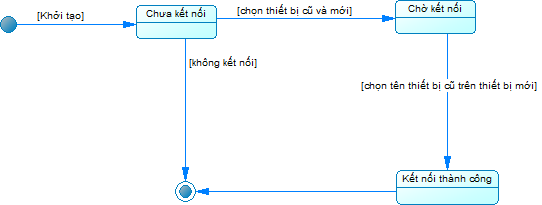
|  |  |
| --- | --- |
| Tên use case | Chọn loại dữ liệu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng phải kết nối thiết bị cũ và mới với nhau |
| Kích hoạt | Tích vào ô tương ứng từng loại dữ liệu |
| Chuỗi sự kiện chính:   1. Người dùng kết nối thành công thiết bị cũ với thiết bị mới 2. Ứng dụng hiển thị danh sách các dữ liệu cần chuyển 3. Người dùng tích vào ô tương ứng với từng loại dữ liệu trên màn hình. | |

* **Kịch bản cho UC “Lọc dữ liệu”**

|  |  |
| --- | --- |
| Tên use case | Lọc dữ liệu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng phải kết nối thiết bị cũ và mới với nhau |
| Kích hoạt | Bấm vào mũi tên sang phải tương ứng với từng loại dữ liệu |
| Chuỗi sự kiện chính:   1. Người dùng kết nối thành công thiết bị cũ với thiết bị mới 2. Ứng dụng hiển thị danh sách các dữ liệu cần chuyển 3. Người dùng bấm vào mũi tên sang phải tương ứng với từng loại dữ liệu trên màn hình. 4. Ứng dụng hiển thị các tiêu chí lọc. 5. Người dùng chọn một hoặc nhiều tiêu chú và bấm “Save”. 6. Ứng dụng lọc dữ liệu theo tiêu chí người dùng đã chọn. | |

### Xây dựng biểu đồ trạng thái

* **Kết nối**



Hình 2.3.2‑1 Biểu đồ trạng thái UC kết nối

* **Chuyển dữ liệu**

****

Hình 2.3.2‑2 Biểu đồ trạng thái UC chuyển dữ liệu

* **Chọn loại dữ liệu**

****

Hình 2.3.2‑3 Biểu đồ trạng thái UC chọn loại dữ liệu

* **Lọc dữ liệu**



Hình 2.3.2‑4 Biểu đồ trạng thái UC lọc dữ liệu

### Xây dựng biểu đồ trình tự

* + - * **Kết nối**

****

Hình 2.3.3‑1 Biểu đồ trình tự UC kết nối

* + - * **Chuyển dữ liệu**

****

Hình 2.3.3‑2 Biểu đồ trình tự UC truyền dữ liệu

* + - * **Chọn loại dữ liệu**

****

Hình 2.3.3‑3 Biểu đồ trình tự UC chọn loại dữ liệu

* + - * **Lọc dữ liệu**

****

Hình 2.3.3‑4 Biểu đồ trình tự UC lọc dữ liệu

### Xây dựng biểu đồ hoạt động

* + - * **Kết nối**

****

Hình 2.3.4‑1 Biểu đồ hoạt động UC kết nối

* + - * **Chuyển dữ liệu**

****

Hình 2.3.4‑2 Biểu đồ hoạt động UC chuyển dữ liệu

* **Chọn loại dữ liệu**

****

Hình 2.3.4‑3 Biểu đồ hoạt động UC chọn loại dữ liệu

* + - * **Lọc dữ liệu**

****

Hình 2.3.4‑4 Biểu đồ hoạt động UC lọc dữ liệu

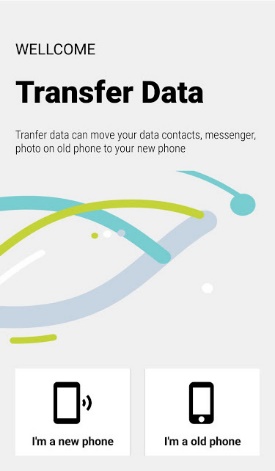
# CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG

## 3.1 Phương pháp xây dựng ứng dụng

* Kết nối thiết bị: Wifi direct
* Lấy dữ liệu trên thiết bị thì sử dụng các Provider:
* Danh bạ (Contact): ContactsConstract
* Tin nhắn (Message): Telephony
* Lịch sử cuộc gọi (CallLog): CallLog.Call
* Hình ảnh (Image): MediaStore.Images
* Video: MediaStore.Video
* Âm thanh (Audio): MediaStore.Audio.
* Tệp tin (File): MediaStore.Files
* Ứng dụng (App): getPackageManager();
* Mã hóa dữ liệu: sử dụng thuật toán mã hóa AES và RSA.

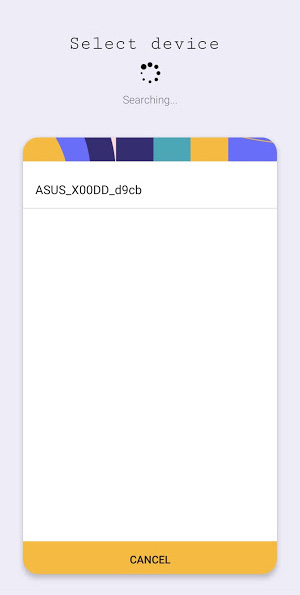
## 3.2 Xây dựng ứng dụng Transfer Data

### 3.2.1 Màn hình chính

****

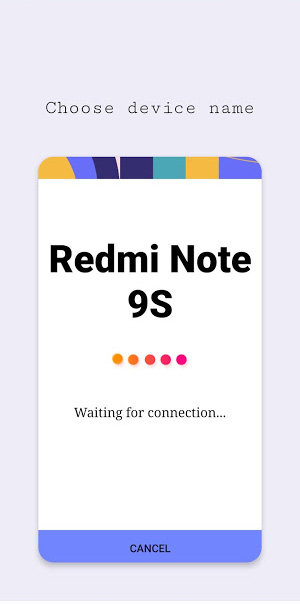
* Giao diện màn hình chính khi mở ứng dụng cho cả thiết bị cũ và thiết bị mới.
* Cho phép lựa chọn thiết bị nào là mấy nhận và máy gửi dữ liệu
* Nút “I’m a new phone”: Thiết bị nào bấm vào sẽ là máy nhận dữ liệu (server)
* Nút “I’m a old phone”:Thiết bị nào bấm và sẽ là máy gửi dữ liệu (client)

### 3.2.2 Màn hình tìm các thiết bị xung quanh

****

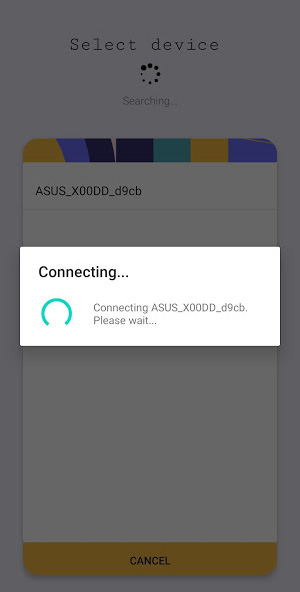
* Giao diện màn hình tìm các thiết bị xung quanh cho máy nhận (máy mới).
* Ứng dụng sẽ hiển thị ra danh cách các thiết bị xung quanh có thể kết nối.
* Người dùng bấm vào tên thiết bị hiển thị trên màn hình để kết nối đến thiết bị đó.
* Nút “CANCEL”: Tắt quá trình tìm thiết bị xung quanh và trở về màn hình chính.

### 3.2.3 Màn hình chờ kết nối

****

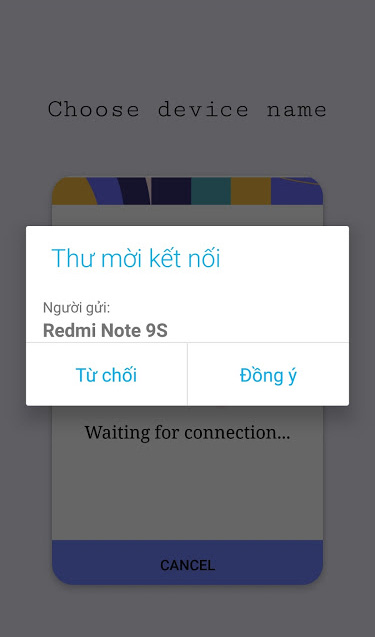
* Màn hình chờ kết nối trên máy gửi (máy cũ).
* Tên thiết bị hiện tại sẽ được hiển thị trên màn hình để người dùng có thể biết tên thiết bị cũ của mình là gì từ đó chọn đúng tên thiết bị trên máy nhận dữ liệu.
* Nút “CANCEL”: Quay trở về màn hình chính.

### 3.2.4 Popup chờ xác nhận kết nối

****

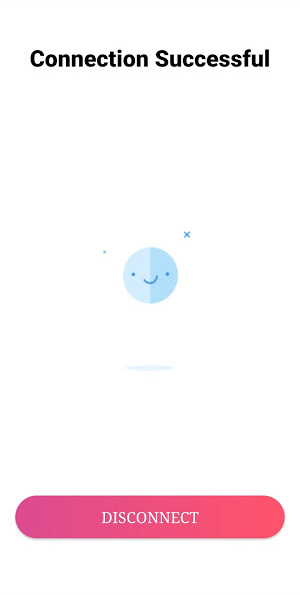
* Trên màn hình tìm các thiết bị xung quanh, khi người dùng bấm vào tên thiết bị mình cần kết nối, một popup “Connecting...” sẽ hiện nên màn hình để chờ máy cũ xác nhận đồng ý kết nối.

### 3.2.5 Popup xác nhận kết nối

****

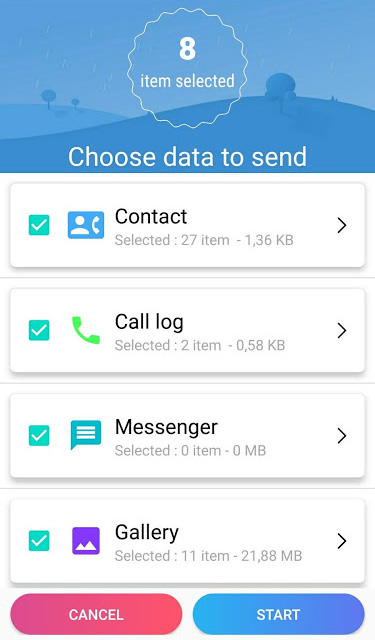
* Khi bên máy nhận (máy mới) bấm vào tên thiết bị cần kết nối thì bên máy gửi (máy cũ) sẽ hiện popup xác nhận kết nối đến thiết bị đó.
* Nút “Đồng ý”: Đồng ý kết nối đến thiết bị mới.
* Nút “Từ chối”: Từ chối kết nối.

### 3.2.6 Màn hình chờ gửi dữ liệu

****

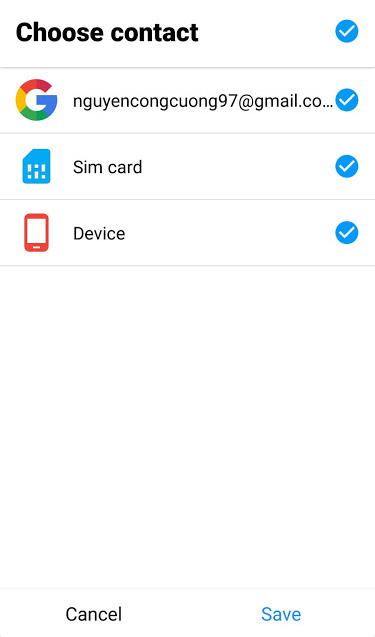
* Khi kết nối thành công, máy nhận sẽ hiển thị màn hình kết nối thành công và chờ máy gửi gửi dữ liệu tới.
* Nút “DISCONECT”: Hủy kết nối với máy gửi dữ liệu và trở về màn hình chính.

### 3.2.7 Màn hình hiển thị danh sách các dữ liệu gửi

****

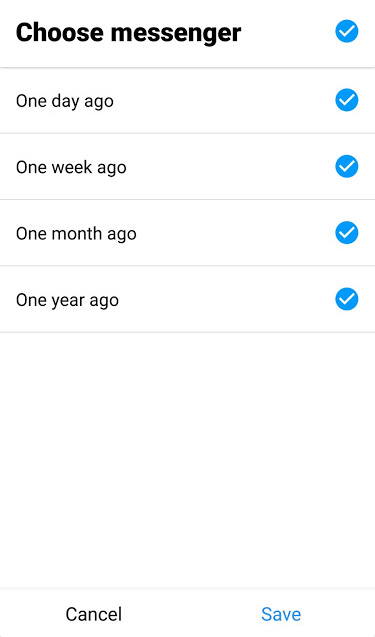
* Màn hình hiển thị danh sách các dữ liệu gửi bên máy cũ (máy gửi).
* Danh sách dữ liệu bao gồm: Danh bạ (contact), lịch sử cuộc gọi (call log), tin nhắn (message), ảnh (gallery), video, âm thanh (audio), tệp tin (file), ứng dụng (app).
* Người dùng có thể chọn hoặc bỏ chọn dữ liệu cần chuyển bằng cách tích và ô checkbox **** tương ứng với từng dữ liệu**.**
* **** Hiển thị số lượng loại dữ liệu đã được chọn để gửi sang máy nhận.
* Mỗi loại dữ liệu sẽ hiển thị số lượng hiện có có và dung lượng.
* ****Chuyển sang màn hình lọc dữ liệu theo các tiêu chí khác nhau ứng với từng loại dữ liệu.
* Nút “CANCEL”: Hủy bỏ quá trình chuyển dữ liệu.
* Nút “START”: Bắt đầu chuyển dữ liệu từ máy cũ sang máy mới.

### 3.2.8 Màn hình lọc danh bạ

****

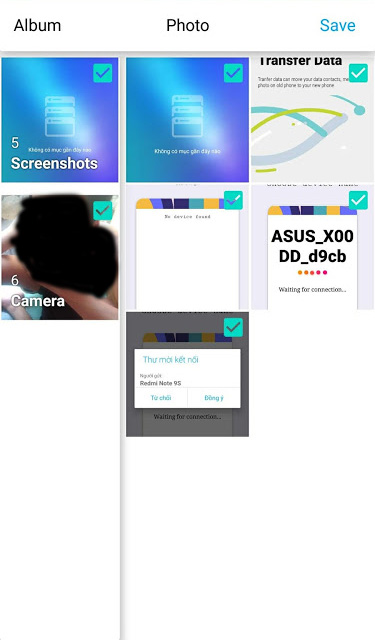
* + Màn hình hiển thị các tiêu chí lọc danh bạ trên máy gửi.
  + Có thể lọc danh bạ theo các tiêu chí:
* Lọc theo tài khoản Google.
* Lọc danh bạ trong máy.
* Lọc danh bạ trong sim.
* Người dùng muốn lọc theo tiêu chú nào thì tích vào ô checkbox  bên phải các tiêu chí .
* Nút “Cancel”: Hủy chọn tiêu chí.
* Nút “Save”: Lưu lại các tiêu chí đã chọn.

### 3.2.9 Màn hình lọc tin nhắn

****

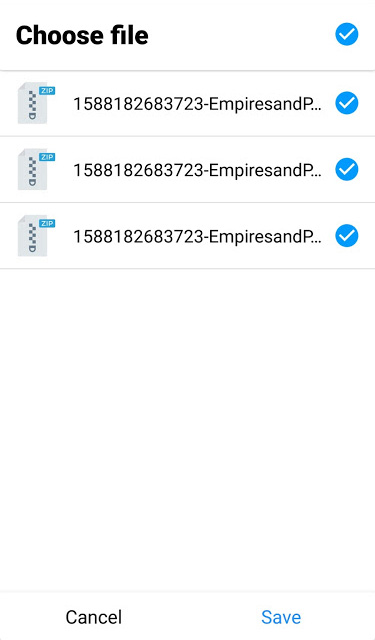
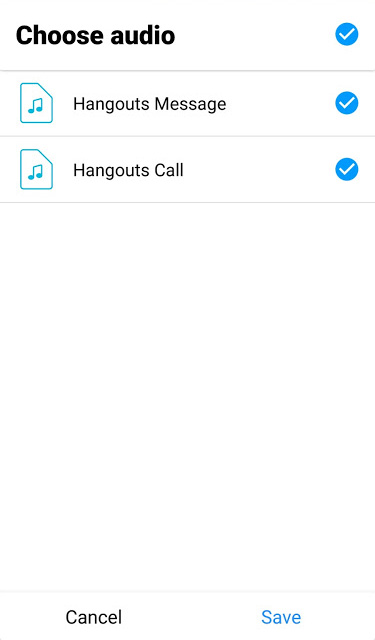
* + Màn hình hiển thị danh sách các tiêu chí lọc tin nhắn.
  + Có thể lọc tin nhắn theo các thiêu chí:
* Tin nhắn một ngày trước.
* Tin nhắn một tuần trước.
* Tin nhắn một tháng trước.
* Tin nhắn một năm trước.
* Người dùng muốn lọc theo tiêu chú nào thì tích vào ô checkbox  bên phải các tiêu chí .
* Nút “Cancel”: Hủy chọn tiêu chí.
* Nút “Save”: Lưu lại các tiêu chí đã chọn.

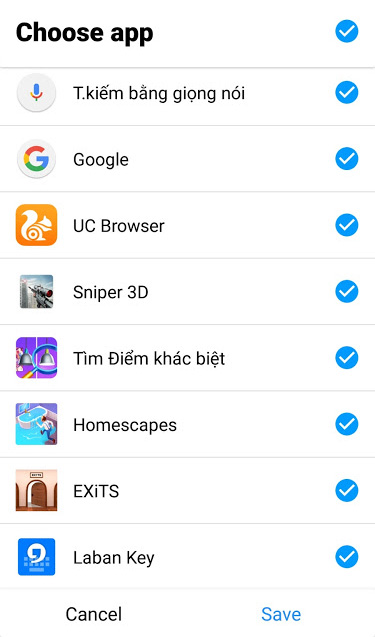
### 3.2.10 Màn hình chọn hình ảnh



* Màn hình hiển thị danh sách các hình ảnh trên máy gửi.
* Người dùng có thể chọn ảnh nào cần gửi bằng cách tích vào checkbox trên ảnh.
* Nút “Save”: Lưu hình ảnh đã chọn.

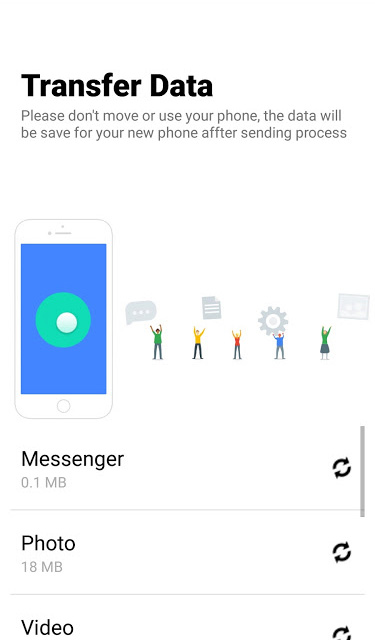
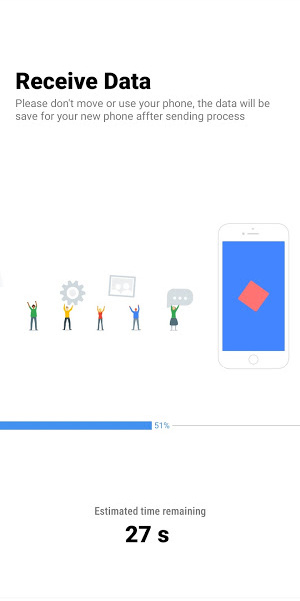
### 3.2.11 Màn hình chọn âm thanh và tệp tin





* Màn hình hiển thị danh sách các âm thanh, tệp tin hoặc ứng dụng trên máy gửi.
* Người dùng có thể chọn file nào cần gửi bằng cách tích vào checkbox bên phải từng file.
* Nút “Cancel”: Hủy chọn.
* Nút “Save”: Lưu file đã chọn.

### 3.2.12 Màn hình hiển thị quá trình chuyển dữ liệu trên máy gửi và nhận, restore dữ liệu trên máy nhận

** **

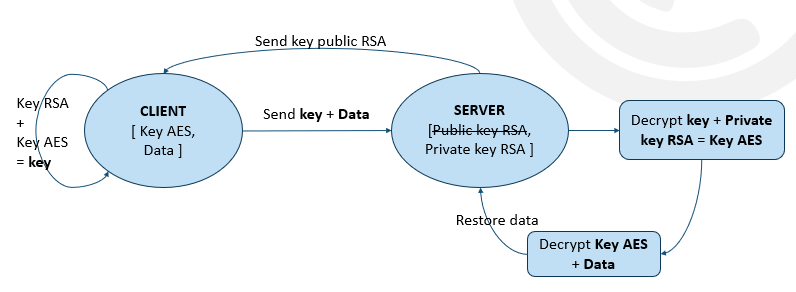
* Trên máy gửi sẽ hiển thị danh sách các loại dữ liệu đang gửi với biểu tượng **** biểu thị cho việc dữ liệu đang được chuyển.
* Trên máy nhận sẽ hiển thị thời gian hoàn thành và tiến độ hoàn thành việc chuyển dữ liệu.

### 3.2.13 Mã hoá và giải mã giữ liệu

**Quy trình mã hoá**

* Sử dụng một thuật toán mã hóa, cộng với khóa của mình để mã hóa file (cách tạo khóa dùng các giải thuật tạo ngẫu nhiên).
* Client nhận khóa công khai từ server rồi mã hóa khóa của thuật toán AES, sau đó gửi khóa và tập tin đã được mã hóa AES sang cho Server
* Khi server nhận tập tin, sẽ dùng khóa bí mật của thuật toán RSA này để giải mã khóa và dùng khóa này để giải mã AES ra tập tin gốc có thể đọc được.

**Sơ đồ mã hoá và giải mã:**



# KẾT LUẬN

Sau một thời gian tìm hiểu, nghiên cứu và làm báo cáo cho đề tài của mình “*Xây dựng ứng dụng chuyển dữ liệu từ smartphone cũ sang smartphone mới*”, với sự cố gắng của bản thân cùng sự hướng dẫn nhiệt tình của thầy giáo hướng dẫn ThS. **Nguyễn Hồng Tân** em đã đạt được nhiều tiến bộ cả về mặt tìm hiểu, nghiên cứu lý thuyết lẫn kỹ năng lập trình. Có thể nói thông qua đồ án tốt nghiệp, em đã đạt được:

1. **Về kiến thức**

* Hiểu biết nhiều hơn về các kỹ thuật trong lập trình android.
* Phân tích thiết kế theo hướng chuyên nghiệp hơn.
* Hiểu hơn về các công nghê truyền file và bảo mật.
* Nâng cao tình thần tự học, tự nghiên cứu.

1. **Về chương trình**

Xây dựng được phần mềm TransferData thực hiện được các yêu cầu đề ra của đề tài tốt nghiệp.

**Ưu điểm:**

* Giao diện thân thiện với người dùng, hiển thị phù hợp với các thiết bị có tỉ lệ màn hình phổ biến hiện nay
* Đáp ứng đủ các yêu cầu cơ bản của một ứng dụng truyền dữ liệu từ thiết bị cũ sang thiết bị mới

**Nhược điểm:**

* Chưa hỗ trợ backup các ứng dụng hệ thống
* Phần restore danh bạ chưa tưng thích hết với tất cả mẫu điện thoại

**Hướng phát triển**

* Phát triển thêm truyền dữ liệu thông qua kết nối cáp USB
* Phát triển thêm tính năng đồng bộ bản backup lên cloud

# TÀI LIỆU THAM KHẢO

[1] Đặng Văn Đức, *Phân tích thiết kế hướng đối tượng*, Nhà xuất bản Viện công nghệ thông tin, 2002.

[2] Đoàn Văn Ban*, Phân tích thiết kế hướng đối tượng bằng UML*, NXB KHKT, 2001

[3] Website tham khảo:

* <https://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager#discoverServices(android.net.wifi.p2p.WifiP2pManager.Channel,%2520android.net.wifi.p2p.WifiP2pManager.ActionListener)>
* <https://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager#discoverPeers(android.net.wifi.p2p.WifiP2pManager.Channel,%20android.net.wifi.p2p.WifiP2pManager.ActionListener)>
* <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct>
* <https://www.tutorialspoint.com/android/index.htm>

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

*Thái nguyên, ngày… tháng…năm 2020*

**Giảng viên hướng dẫn**