TRƯỜNG ĐẠI HỌC NAM CẦN THƠ

KHOA KỸ THUẬT CÔNG NGHỆ



BÁO CÁO

MÔN:

LẬP TRÌNH THIẾT BỊ DI ĐỘNG

Tên đề tài:

APP GHI CHÚ

Giảng viên: Mai Yến Trinh

Thành viên nhóm 12:

Nguyễn Trung Nghĩa

Mssv: 188680

Lê Nguyễn Thúy Vi

Mssv: 188642

LỜI CẢM ƠN!

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép chúng em được bày tỏ lòng biết ơn đến tất cả các bạn và thầy cô đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè.

Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa Kỹ thuật công nghệ đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, em xin chân thành cảm ơn cô Mai Yến Trinh– người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này trong thời gian qua.

NGƯỜI THỰC HIỆN

PHỤ LỤC

[Chương 1: GIỚI THIỆU 3](#_Toc92260228)

[I. Lí do chọn đề tài 3](#_Toc92260229)

[II. Sơ lược về đề tài 4](#_Toc92260230)

[1. Mục đích nghiên cứu 4](#_Toc92260231)

[2. Đối tượng nghiên cứu 4](#_Toc92260232)

[III. Ý nghĩa của đề tài 4](#_Toc92260233)

[Chương 2: NỘI DUNG 4](#_Toc92260234)

[I. Cơ sở lí thuyết 4](#_Toc92260235)

[1. Lịch sử phát triển của Android 4](#_Toc92260236)

[2. Kiến trúc Android 5](#_Toc92260237)

[3. Các thành phần cơ bản của Android 8](#_Toc92260238)

[4. Hệ quản trị cơ sở dữ liệu SQLITE 9](#_Toc92260239)

[II. Giao diện 10](#_Toc92260240)

[1. Hướng dẫn sử dụng app 10](#_Toc92260241)

[Chương 3: KẾT LUẬN 13](#_Toc92260242)

# Chương 1: GIỚI THIỆU

1. Lí do chọn đề tài

Khi mà thời đại công nghệ phát triển để bắt kịp xu hướng ở Việt Nam hiện nay, thì công nghệ thông tin cũng đã được nghiên cứu và ứng dụng rộng rãi trong nhiều lĩnh vực, đặt biệt là lĩnh vực phần mềm hay lập trình ứng dụng, tin học đã trở thành một công cụ không thể thiếu và là lĩnh vực hàng đầu trên thế giới.

Gần đây điện thoại thông minh đã trở thành “vật bất li thân” với tất cả mọi người. Chính vì như vậy chúng em quyết định viết phần mền tiện ích nâng cao giá trị sử dụng của chúng cũng đã và đang trở thành trào lưu cho các công ty phát triển phần mềm, các giới lập trình viên chuyên nghiệp cũng như không chuyên. Đặt biệt là những ứng dụng dành cho máy chạy trên hệ điều hành Android.

Sau quá trình nghiên cứu và tìm hiểu về hệ điều hành này thì em nhận thấy có thể phát triển một chương trình giúp người dùng lập kế hoạch và quản lí nó một cách dễ dàng, hiệu và và mang đến chất lượng. Xuất phát từ nhận định trên em đã xây dựng ứng dụng Note với hi vọng mang lại một ứng dụng thuận tiện, hữu ích cho tất cả kế hoạch, lịch trình, thời gian biểu cũng như sắp xếp công việc hàng ngày.

1. Sơ lược về đề tài
2. Mục đích nghiên cứu

* Nghiên cứu kiến trúc Android và các thành phần cơ bản của Android.
* Nghiên cứu cách phát triển ứng dụng trên Android.
* Xây dựng ứng dụng Note trên nền tảng Android.

1. Đối tượng nghiên cứu

Với đề tài này em tập trung vào các nội dung sau:

* Tìm hiểu về Activity trong Android
* Tìm hiểu về Sevice trong Android
* Cách tạo giao diện trong Android
* Cách lưu trữ dữ liệu trong Android
* Xây dựng ứng dụng Note

1. Ý nghĩa của đề tài
2. Ý nghĩa lí luận: đề tài này tài liệu tham khảo rất hữu ích cho những ai muốn tìm hiểu về Android. Bên cạnh đó chưa có nhiều tài liệu tham khảo bằng tiếng Việt, do đó em muốn nghiên cứu để đóng góp những hiểu biết mà mình có để làm giàu thêm tài liệu cho đề tài này.
3. Ý nghĩa thực tiễn: hiện tại hệ điều hành Android đang được sử dụng rộng rãi trên các thiết bị di động. Nhu cầu thư giản, giải trí, quản lí công việc kế hoạch của người dùng trên điện thoại rất cao. Vì vậy ứng dụng Note là một ý tưởng không tồi giúp người dùng quản lí lịch trình, lên kế hoạch cho chính bản thân mình một cách hiệu quả, thuận tiện và chất lượng.

# Chương 2: NỘI DUNG

1. Cơ sở lí thuyết
2. Lịch sử phát triển của Android

Hệ điều hành Android có nguồn gốc tên từ Android, Inc. – chính công ty cha đẻ của Android. Công ty chính thức thành lập ở Palo Alto, California khoảng tháng 10 năm 2003 do Andy Rubin – đồng sáng lập công ty Danger, Rich Miner – đồng sáng lập công ty Viễn thông Wildfire, Nick Sears – cựu Phó giám đốc T-Mobile, và Chris White – trưởng thiết kế và giao diện tại WebTV.

Đến 17/08/2005 Google chính thức thu mua lại Android, Inc., từ đó Android chính thức thuộc quyền sở hữu của Google. Bộ phận cốt cán nhân sự chính của Android, Inc. có Rubin, Miner và White, vẫn duy trì chức vụ sau lần mua bán này. Vào thời điểm đó không có quá nhiều thông tin về công ty này nhưng ngầm dự báo về tương lai Google sẽ lấn sân sang thị trường điện thoại di động.

Phát triển tại Google, nhóm do Rubin đứng đầu chính thức cho thành lập và phát triển nền tảng thiết bị di động trên nền tảng Linux. Google mang tới lời hứa với nhà sản xuất điện thoại và các nhà mạng cho tương lai nâng cấp hệ thống và cho nền tảng tích hợp mươt mà.

1. Kiến trúc Android

2.1 Nhân Linux (Linux Kernel)

Nền tảng của kiến trúc hệ điều hành Android là nhân Linux. Ví dụ: Android Runtime (ART) dựa trên nhân Linux cho các chức năng cơ bản như phân luồng và quản lý bộ nhớ cấp thấp.

Bằng cách sử dụng nhân Linux, Android có thể tận dụng các tính năng bảo mật chính và cho phép các nhà sản xuất thiết bị phát triển trình điều khiển phần cứng.

2.2 Lớp trừu tượng phần cứng (Hardware Abstraction Layer – HAL)

Lớp trừu tượng phần cứng (Hardware Abstraction Layer – HAL) cung cấp các giao diện tiêu chuẩn thực hiện các khả năng của phần cứng thiết bị với khung API Java cấp cao hơn. HAL bao gồm nhiều mô-đun, mỗi mô-đun thực hiện một giao diện cho một loại thành phần phần cứng cụ thể, chẳng hạn như mô-đun camera hoặc bluetooth. Khi một API thực hiện truy cập phần cứng của thiết bị, hệ thống Android sẽ tải mô-đun cho thành phần phần cứng đó.

2.3 Thời gian chạy Android (Android Runtime)

Đối với các thiết bị chạy Android phiên bản 5.0 (API cấp 21) trở lên, mỗi ứng dụng chạy trong quy trình riêng và với phiên bản Android Runtime (ART) riêng. ART được viết để chạy nhiều máy ảo trên các thiết bị có bộ nhớ thấp bằng cách thực thi các tệp DEX, một định dạng bytecode được thiết kế đặc biệt cho Android được tối ưu hóa cho bộ nhớ tối thiểu.

Một số tính năng chính của ART bao gồm:

– Biên dịch trước thời gian (AOT) và đúng lúc (JIT).

– Thu gom rác tối ưu hóa (GC).

– Trên Android 9 (API cấp 28) trở lên, chuyển đổi tệp định dạng Dalvik Executable (DEX) của gói ứng dụng thành mã máy nhỏ gọn hơn.

– Hỗ trợ gỡ lỗi tốt hơn, bao gồm trình biên dịch lấy mẫu chuyên dụng, các ngoại lệ chẩn đoán chi tiết và báo cáo sự cố cũng như khả năng thiết lập các điểm theo dõi để giám sát các trường cụ thể.

Trước phiên bản Android 5.0 (API cấp 21), Dalvik là Android runtime. Nếu ứng dụng của bạn chạy tốt trên ART thì ứng dụng đó cũng sẽ hoạt động trên Dalvik nhưng điều ngược lại chưa hẳn đã đúng.

Android cũng bao gồm một bộ thư viện cung cấp hầu hết các chức năng của ngôn ngữ lập trình Java, bao gồm một số tính năng của ngôn ngữ Java 8 mà khung API Java sử dụng.

2.4 Thư viện C/C++ gốc (Native C/C++ Libraries)

Nhiều thành phần và dịch vụ hệ thống cốt lõi của Android, chẳng hạn như ART và HAL, được xây dựng từ mã gốc được viết bằng C và C++. Nền tảng Android cung cấp các API khung Java để hiển thị chức năng của một số thư viện gốc này cho các ứng dụng. Ví dụ: Bạn có thể truy cập OpenGL ES thông qua API Java OpenGL để thêm hỗ trợ vẽ và thao tác đồ họa 2D và 3D trong ứng dụng của bạn.

Nếu bạn đang phát triển một ứng dụng yêu cầu mã C hoặc C++, bạn có thể sử dụng Android NDK để truy cập trực tiếp vào một số thư viện nền tảng gốc này từ mã gốc của mình.

2.5 Khung API Java (Java API Framework)

Toàn bộ tính năng có sẵn của hệ điều hành Android thông qua các API được viết bằng ngôn ngữ Java. Các API này tạo thành các khối xây dựng mà bạn cần để tạo ứng dụng Android bằng cách đơn giản hóa việc sử dụng lại các thành phần và dịch vụ hệ thống mô-đun, cốt lõi, bao gồm những điều sau:

– Hệ thống xem phong phú và có thể mở rộng mà bạn có thể sử dụng để xây dựng giao diện người dùng của ứng dụng.

– Quản lý tài nguyên, cung cấp quyền truy cập vào tài nguyên không code.

– Quản lý thông báo cho phép tất cả các ứng dụng để cảnh báo hiển thị tùy chỉnh trong thanh trạng thái.

– Hoạt động quản lý để quản lý vòng đời của ứng dụng và cung cấp chuyển hướng trở lại màn hình.

– Nhà cung cấp nội dung cho phép ứng dụng truy cập dữ liệu từ các ứng dụng khác hoặc chia sẻ dữ liệu của riêng họ.

– Các nhà phát triển có toàn quyền truy cập vào cùng một API khung mà các ứng dụng hệ thống Android sử dụng.

2.6 Ứng dụng hệ thống (System Apps)

Android đi kèm với một tập hợp các ứng dụng cốt lõi cho email, nhắn tin SMS, lịch, trình duyệt internet, danh bạ… Các ứng dụng đi kèm với nền tảng không có trạng thái đặc biệt. Vì vậy, ứng dụng của bên thứ ba có thể trở thành trình duyệt web mặc định của người dùng, trình nhắn tin SMS hoặc thậm chí là bàn phím mặc định của người dùng.

1. Các thành phần cơ bản của Android

3.1 Activity

Trong ứng dụng Android, Activity đóng vai trò đặc biệt quan trọng, là nơi giúp người dùng tương tác trực tiếp với ứng dụng, ví dụ như gọi điện thoại, chụp ảnh, gửi e-mail hoặc xem bản đồ. Activity được coi là xương sống của một ứng dụng Android, một ứng dụng có thể có một hoặc nhiều Activity (bất kì ứng dụng nào cũng cần có ít nhất 1 Activity).

Activity có thể hiển thị ở chế độ toàn màn hình, dạng cửa sổ hoặc với một kích thước nhất định. Một Activity có thể gọi đến một Activity khác, Activity được gọi đến sẽ tương tác với người dùng tại thời điểm được gọi tới.

3.2 Sevice

Service là một thành phần ứng dụng chạy ngầm trên hệ điều hành ví dụ như nghe nhạc, hoặc tương tác với một content provider. Service không tương tác trực tiếp với người dùng, khi service chạy thì người dùng vẫn có thể tương tác với một thành phần khác trong ứng dụng hoặc có thể tương tác với một ứng dụng khác trong hệ thống.

3.3 Broadcast Receiver

Broadcast Receiver là một thành phần của ứng dụng giúp lắng nghe các sự kiện mà hệ thống phát ra thông qua Intent, hệ thống có thể truyền phát ngay cả khi app không chạy. Broadcast Receiver không có giao diện cụ thể nhưng nó có thể thực hiện thông báo thông qua thanh Notification. Có rất nhiều broadcast được phát ra từ hệ thống, chúng ta có thể lấy ví dụ như một broadcast thông báo rằng màn hình điện thoại đã tắt, hay điện thoại đang ở trạng thái “Battery Low”, “Power Connected”, “Power Disconnected” hoặc một bức ảnh đã được chụp. Cũng có những broadcast được phát ra từ ứng dụng như sau khi download một tệp, ví dụ: Sau khi hoàn thành download một tệp tin, ứng dụng A phát ra thông báo là dữ liệu đã download xong, tệp đã sẵn sàng cho các ứng dụng khác có thể sử dụng.

3.4 Content Provider

Content Provider là một thành phần giúp các một ứng dụng có thể đọc và ghi dữ liệu từ một file hoặc từ SQLite của một ứng dụng khác trong cùng một hệ thống. Bất kỳ ứng dụng nào có quyền (permisson) đều có thể truy xuất, chỉnh sửa dữ liệu của một ứng dụng khác.

Content Provider được chia thành 2 loại:

* Native Content Provider: Là những Content Provider có sẵn, được tạo ra bởi hệ thống, ví dụ như Contacts, Message, …
* Custom Content Provider: Bao gồm các Content Provider được tạo ra bởi các developer phụ thuộc vào đặc điểm của từng ứng dụng.

1. Hệ quản trị cơ sở dữ liệu SQLITE

4.1. SQLite là gì?

SQLite là một hệ quản trị cơ sở dữ liệu hay còn gọi là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, khác với các hệ quản trị khác như MySQL, SQL Server, Ocracle, PostgreSQL… SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine truyền thống, không cần mô hình client-server nên rất nhỏ gọn. SQLite được sử dụng vào rất nhiều chương trình từ desktop đến mobile hay là website.

4.2 Lịch sử hình thành SQLite như thế nào?

Năm 2000: D.Richard Hipp đã thiết kế SQLite dưới dạng thư viện bằng ngôn ngữ lập trình c với mục đích không cần quản trị để điều hành một chương trình.

Vào tháng 8, SQLite 1.0 được công bố với GNU Database Manager.

Năm 2011: Hipp bổ sung UNQI Interface cho SQLite DB và để phát triển UNQLite (là một Document Oriented Database).

4.3 Tại sao sử dụng SQLite?

SQLite là mã nguồn mở, là một trong những Database Engine phát triển nhanh nhất, sự phát triển của nó là về mặt tính phổ biến. Ngoài những lý do trên thì không thể không kể đến những ưu điểm khi sử dụng SQLite, sau đây là phần ưu điểm của SQLite.

4.4 Ưu điểm của SQLite.

SQLite không cần mô hình client – server để hoạt động.

SQLite không cần phải cấu hình tức là bạn không cần phải cài đặt.

Với SQLite database được lưu trữ trên một tập tin duy nhất.

SQLite hỗ trợ hầu hết các tính năng của ngôn ngữ truy vấn SQL theo chuẩn SQL92.

SQLite rất nhỏ gọn bản đầy đủ các tính năng nhỏ hơn 500kb, và có thể nhỏ hơn nếu lược bớt một số tính năng.

Các thao tác dữ liệu trên SQLite chạy nhanh hơn so với các hệ quản trị cơ sở dữ liệu theo mô hình client – server.

SQLite rất đơn giản và dễ dàng sử dụng.

SQLite tuân thủ 4 tính chất ACID (là tính nguyên tổ (Atomic), tính nhất quán (Consistent), tính cô lập (Isolated), và tính bền vững (Durable).

Với đặc tính nhỏ gọn, truy xuất dữ liệu nhanh SQLite thường được sử dụng để nhúng vào các dự án.

4.5 Nhược điểm của SQLite

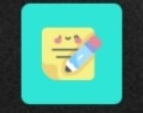
Ngoài những ưu điểm đã kể ra ở trên SQLite cũng có một số mặt hạn chế nếu đem so sánh với các hệ quản trị khác.

Do sử dụng cơ chế coarse-gained locking nên trong cùng một thời điểm SQLite có thể hỗ trợ nhiều người đọc dữ liệu, nhưng chỉ có 1 người có thể ghi dữ liệu.

SQLite không phải là lựa chọn hoàn hảo để đáp ứng các nhu cầu xử lý trên một khối lượng dữ liệu lớn, phát sinh liên tục.

1. Giao diện
2. Hướng dẫn sử dụng app

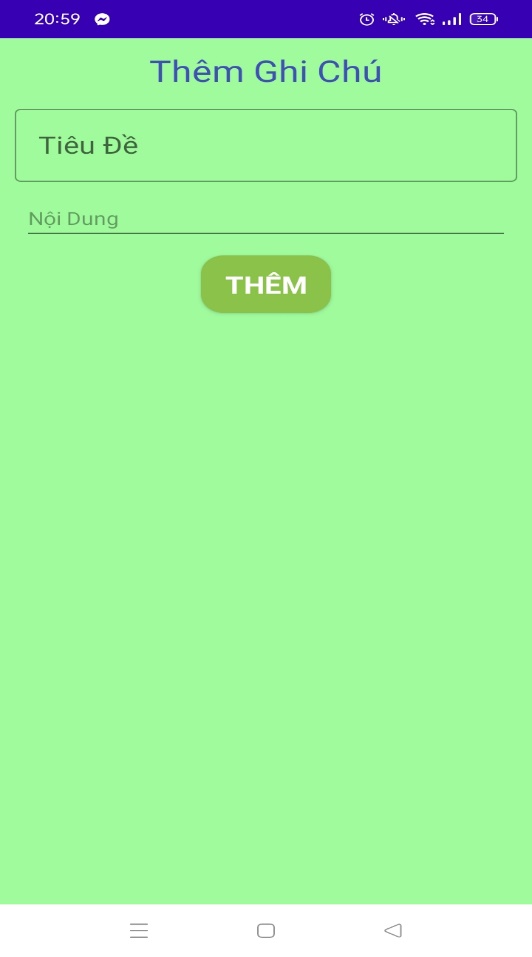
Sau khi cài đặt thành công ta chọn Icon của App Easy Notes hiển thị trên màn hình điện thoại



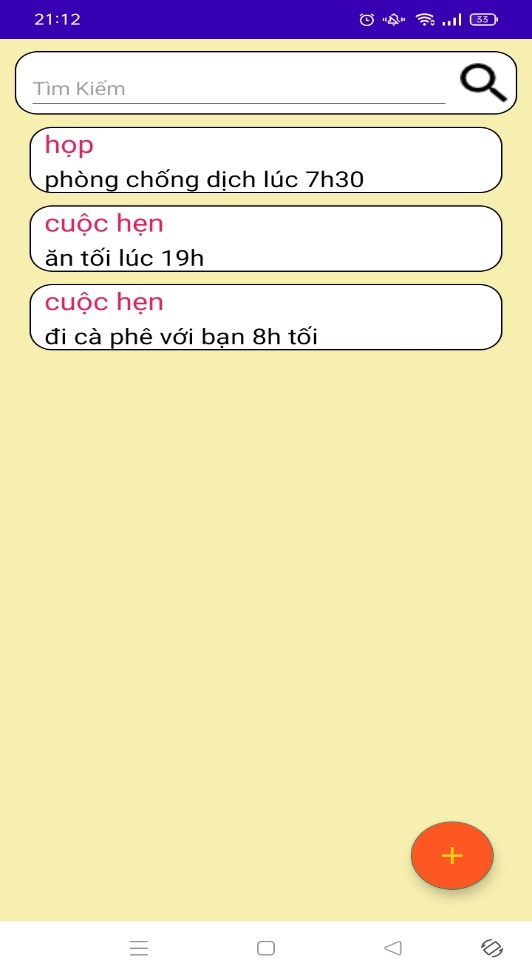
Ta thấy giao diện app hiện lên như sau:



Nhấn vào dấu cộng ở góc phải để thêm và nhập thông tin:



Giao diện khi thêm thành công:



Giao diện khi muốn chỉnh sửa ghi chú:



Giao diện khi muốn tìm ghi chú:



# Chương 3: KẾT LUẬN

Thời buổi công nghệ hiện nay, hầu như ai cũng sở hữu cho mình những chiếc smartphone, tablet, laptop để phục vụ nhu cầu học tập và công việc. Chính vì vậy, việc ghi chú nhanh những thông tin cần thiết cũng thay đổi theo thời gian. Nếu phương pháp ghi chép truyền thống bằng giấy bút có thể sẽ thất lạc sau một thời gian thì nay chúng ta đã có những ứng dụng tuyệt vời để "take note" các thông tin quan trọng hiệu quả hơn. Sử dụng ghi chú, việc cần làm và lịch biểu tại cùng một nơi để giảm bớt sự xao nhãng và làm được nhiều việc hơn, dù là ở cơ quan, ở nhà hay ở bất cứ đâu.

Phần mềm còn một số hạn chế trong quá trình lập trình còn chưa làm được chắc chắn chúng em sẽ cải thiện.

Lần nữa chúng em xin chân thành cảm ơn cô đã tận tình giảng dạy, trang bị cho chúng em những kiến thức bổ ích và quý báu trong suốt thời gian thực hiện đề tài này cũng như đã tạo điều kiện cho chúng em được thực hiện đề tài này một cách tốt nhất.

Cảm ơn các bạn trong lớp đã ủng hộ, giúp đỡ và động viên trong quá trình làm đề tài. Đó là những nguồn khích lệ, động viên to lớn giúp em có thêm nghị lực để hoàn thành tốt đề tài này.

Mặc dù em đã cố gắng, nỗ lực rất nhiều nhưng chắc hẳn đề tài vẫn còn nhiều thiếu sót mong các thầy cô thông cảm và châm chế cho những lỗi nhỏ chúng em mắc phải. Chúng em rất mong nhận được sự góp ý của quý thầy cô và các bạn để đề tài của em được hoàn thiện hơn và tốt hơn trong những lần tới.

XIN CHÂN THÀNH CẢM ƠN!