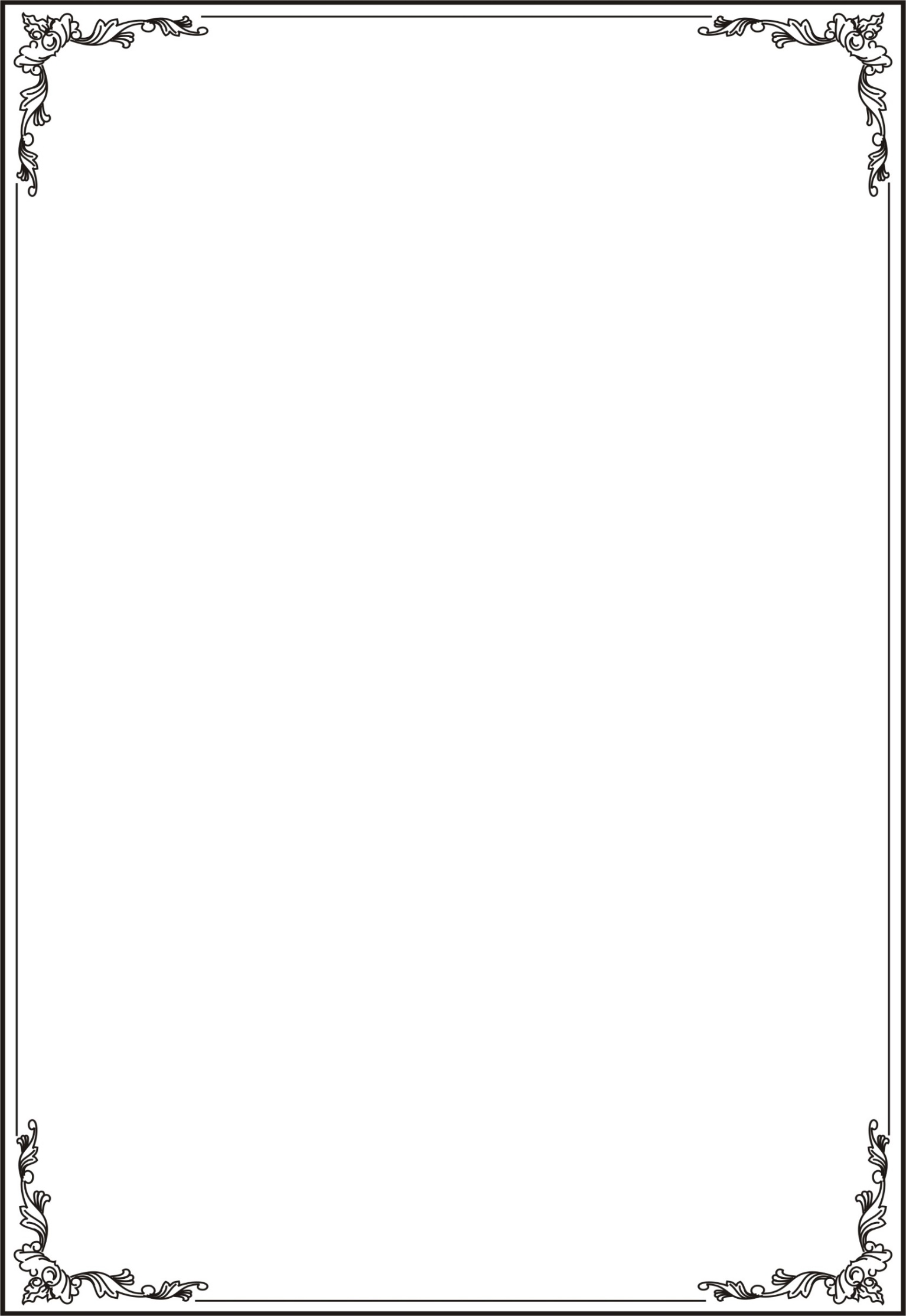
****TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KỲ**

**PHÁT TRIỂN ỨNG DỤNG**

**VIẾT NHẬT KÝ**

*Người hướng dẫn*: **LÊ VĂN VANG**

*Người thực hiện*: **HOÀNG DIỆP NGUYÊN KIỆT– 51503626**

**NGUYỄN HỮU HIỀN VINH- 51503635**

Lớp **: 15050310**

Khoá  **: 19**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

LỜI CẢM ƠN

Trong suốt thời gian qua, nhờ sự giảng dạy tận tâm của quý Thầy Cô Khoa Công Nghệ Thông Tin, trường Đại học Tôn Đức Thắng, em đã học hỏi được rất nhiều điều bổ ích và tích lũy cho mình một số kiến thức để hoàn thành bài báo cáo này. Em xin chân thành cảm ơn.

Em xin cảm ơn Thầy Lê Văn Vang đã tận tình chỉ bảo chúng em qua những buổi học tại lớp, Thầy đã chỉ em cách thức làm bài, chỉ điểm những chỗ còn sai sót chưa phù hợp cũng như phải làm sao để trình bày bố cục đẹp. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì bài thu hoạch của em cũng rất khó để hoàn thiện. Một lần nữa em xin chân thành cảm ơn Thầy.

Bước đầu đi vào thực tế với nền kiến thức mở rộng, kiến thức của em còn hạn chế và nhiều bỡ ngỡ. Vì thế, trong quá trình biên soạn khó tránh khỏi những sai sót, em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy/Cô và Các bạn để bài báo cáo hoàn thiện hơn.

**BÀI TẬP ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là bài tập của riêng tôi và được sự hướng dẫn của thầy Lê Văn Vang. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài tập này còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 17 tháng* 2 *năm* 2017

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Hoàng Diệp Nguyên Kiệt*

*Nguyễn Hữu Hiền Vinh*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

MỤC LỤC

1. Giới thiệu
   1. Giới thiệu tổng quan
   2. Khảo sát các ứng dụng liên quan
   3. Mục tiêu của đề tài
2. Cơ sở lý thuyết

2.1 Android

2.2 SQLite trong Android

2.3 Intent

2.4 Dialog

2.5Permissions (quyền)

2.6 Mã hóa một chiều MD5

1. Đặc tả hệ thống và Thiết kế hệ thống.

3.1 giới thiệu chi tiết từng chức năng

3.1.1. Chức năng tạo mật khẩu.

3.1.2 Chức năng tạo nhật ký

3.1.3 Cập nhật ngày tháng hiện tại

3.1.4 Chức năng Lưu nhật ký

3.1.5 Chức năng xóa nhật ký

3.1.6 Chức năng hỏi người dùng trước khi thoát.

3.2 Thiết kế hệ thống

3.2.1 Thiết kế DataBase

3.2.2 Lấy dữ liệu từ Database

3.2.3 Tạo và showDialog().

3.2.4 Tạo cơ sở dữ liệu ngày tháng.

3.2.5 Cập nhật ngày tháng

3.2.6 Cập nhật dữ liệu

3.2.7 Một số Intent qua lại giữa các Activity

1. Kết quả thực nghiệm
2. Kết luận và hướng phát triển

**1 Giới thiệu tổng quan**

**1.1 Nhật ký**

Nhật ký là loại văn xuôi ghi chép những sinh hoạt thường ngày hoặc cảm xúc riêng tư không thể chia sẻ.

Chúng ta thường ghi lại những sự kiện đáng nhớ vào nhật ký. Nhưng khi cuốn Nhật ký không ở bên cạnh hoặc cuốn Nhật ký trông rườm rà khó khăn thì chính ngay lúc này đây chúng ta cần ứng dụng My Diary.

My Diary là một cuốn nhật ký mà bạn có thể mang theo bất kỳ nơi đâu và bất cứ nơi nào bạn muốn.

Khi cần viết nhật ký, ta chỉ việt mở app lên và viết thôi chứ không cần cuốn sổ và cây bút như xưa nữa.

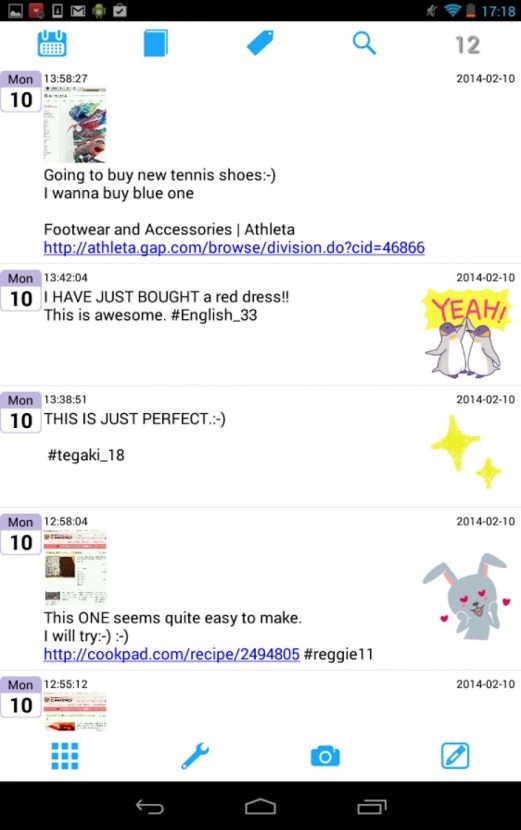
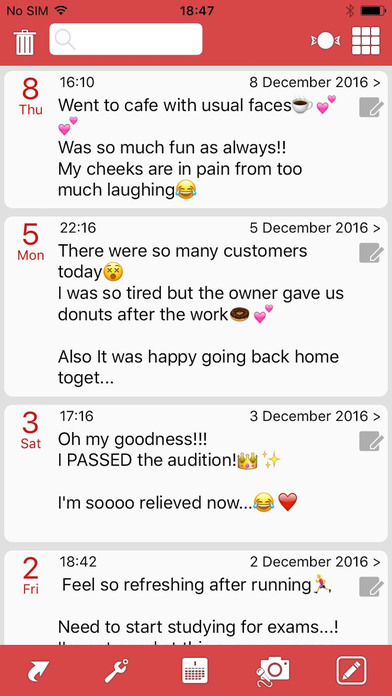
Ngoài ra ta còn có thể share lên mạng xã hội, hay lưu trữ nhật ký trên Google Drive.

Bạn sẽ có thật nhiều cảm hứng khi viết nhật ký vì được viết trên một ứng dụng cực xinh xắn cùng với nhiều tính năng, phông chữ, màu nền cùng với nhiều icon đẹp mắt.

Ứng dụng này rất hữu ích phải không ạ ☺

**1.2 Khảo sát các ứng dụng liên quan**

*Ứng dụng :**Nhật ký khoảnh khắc (Moment Diary)*



\_Bạn có thể viết những ghi chú, những sự kiện kèm theo ngày tháng năm..

\_Đưa ghi chú lên email của bạn.

\_Có chức năng sao lưu, phục hồi, đồng bộ hóa dữ liệu trên mọi thiết bị.

\_Khóa mật mã ( được hỗ trợ như chức năng bảo mật, bạn có thể dễ dàng đổi mật khẩu, bật/tắt khóa,.. Khi quên mật khẩu, có thể đổi lại thông qua số điện thoại).

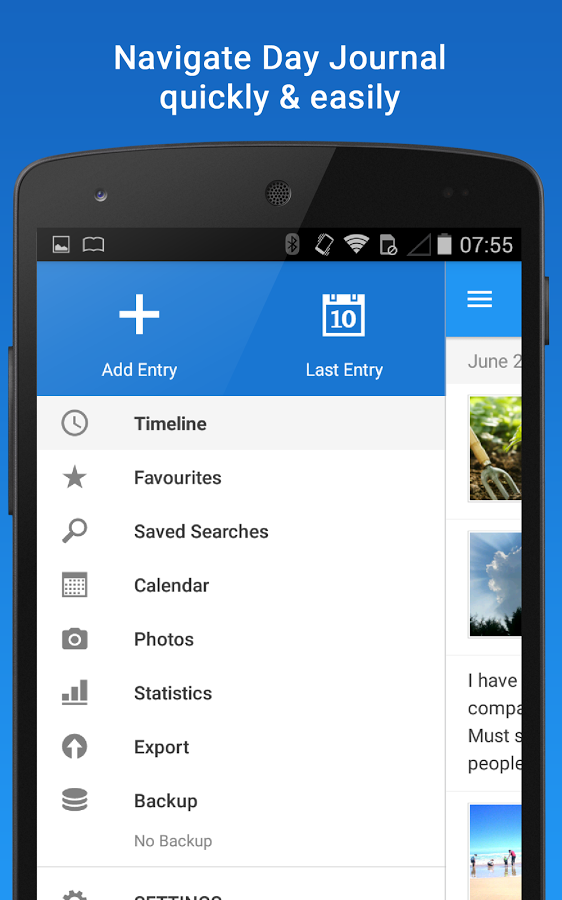
\_Bạn có thể thêm hình ảnh vào trong ghi chú.

\_Thay đổi màu nền.

\_Thay đổi phông chữ.

\_Thêm biểu tượng ưa thích vào trong nhật ký.

### Ứng dụng *:* ***Day Journal.***



\_Xem TimeLine.

\_Mục yêu thích.

\_Mục tìm kiếm.

\_Bảo vệ nhật ký của bạn với mật khẩu.

\_Sao lưu dữ liệu và phục hồi.

\_Tự động ghi nhận vị trí và thời tiết của mình.

\_Biểu tượng cảm xúc để diễn tả cảm xúc của bạn.

\_Chia sẻ mục qua Mail, Facebook, SMS, Twitter, …

\_Thay đổi giao diện màu.

… còn nhiều hơn nữa.

**1.3 Mục tiêu đề tài**

\_Thiết kế ứng dụng viết nhật ký có nhiều chức năng hấp dẫn và tiện ích.

\_Ứng dụng đơn giản, dễ sử dụng.

\_Ứng dụng chạy được trên nhiều máy khác nhau trên hệ điều hành Android.

\_Tạo nguồn cảm hứng cho người dùng khi viết nhật ký.

\_Ứng dụng không bị bug, lag. Không chiếm nhiều tài nguyên.

**2. Cơ sở lý thuyết**

**2.1 Android**

**2.1.1 giới thiệu**

Android là hệ điều hành giành cho thiết bị di động, được phát triển bởi Google và ngày càng trở nên phổ biến với các hãng điện thoại.

Android được xây dựng trên nhân Linux và được phân phối miễn phí. Mọi ứng dụng Android điều được viết trên cùng một tập API, thế nên không có sự phân biệt giữa các ứng dụng mặc định và các ứng dụng bên thứ ba. Người dùng hoàn toàn có thể thay thế mọi ứng dụng mặc định bằng mọi ứng dụng ưa thích của mình, thậm chí ngay cả màn hình thực hiện cuộc gọi và màn hình nhà (home screen)

Các nhà phát triển viết ứng dụng Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đích tạo nên một chuẩn mở cho điện thoại di động trong tương lai.

Google công bố hầu hết các mã nguồn của Android theo bản cấp Apache.



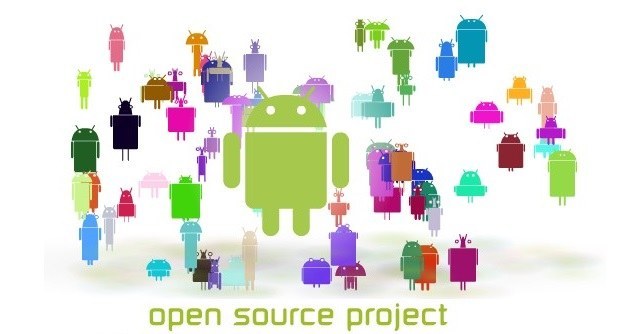
*Hình 1.1 Giao diện màn hình android*

**2.1.2 Ưu điểm và khuyết điểm**

a Ưu điểm:

* Hệ điều hành mở:

Điều làm cho Android được chú ý nhất chính vì nó là mã nguồn mở. Android đã thu hút rất nhiều lập trình viên và nhà phát triển từ tất cả các tầng lớp xã hội khi Google phân phối miễn phí Android và các đoạn mã có bản quyền khác cho bất cứ ai muốn tham gia phát triển phần mềm cho Google. Vì thế, trong một thời gian ngắn, các lập trình viên cũng như các nhà phát triển có thể nhanh chóng đưa ra phiên bản Android đươc tùy chỉnh của riêng mình.



* Đa nhiệm

Ngay từ phiên bản Android 1.0 đã có nhiều ứng dụng chạy cùng lúc(các ứng dụng hệ thống hoặc các ứng dụng từ Android Marketplace). Người dùng Android được hưởng lợi rất nhiều từ sự khác biệt này, vì họ có thể nhận thông báo, nghe nhạc và thậm chí ghi dữ liệu GPS mà không cần giữ ứng dụng luôn mở.

* Gía cả hợp lý:

Mức giá để sở hữu một thiết bị là vấn đề mà không ít người quan tâm đến. Với các thiết bị Android, trong khoảng từ 1 – 20 triệu đồng, bạn sẽ có muôn vàn những lựa chọn, từ những thiết bị bình dân vừa túi tiền đến những thiết bị cao cấp sở hữu vô số những tính năng hữu ích. Trên thị trường hiện nay có rất nhiều hãng điện tử như Samsung, LG, Sony, Lenovo, Oppo, Huawei … tung ra hàng loạt các sản phẩm đủ mọi kích cỡ, đủ mọi mức giá vì vậy người dùng sẽ có nhiều sự để lựa chọn hơn so với các thiết bị sử dụng hệ điều hành khác.

* Kho ứng dụng Google Play store đồ sộ:

Trước đây, khi mới ra mắt kho ứng dụng Google Play chỉ vỏn vẹn có một số ứng dụng. Nhưng đến nay, kho ứng dụng này đã đạt đến hơn 650.000 ứng dụng, trong đó có rất nhiều ứng dụng miễn phí mà vô cùng tiện dụng. Đây cũng là lý do để giải thích cho sự phát triển chóng mặt của hệ điều hành Android trong thế giới di động. Nếu bạn sử dụng một thiết bị Android, chắc hẳn bạn sẽ không phải bận tâm vì “thiếu ứng dụng để cài” mà cái bạn sẽ bận tâm đó là “không đủ bộ nhớ để cài”.



* Dễ sử dụng

Có lẽ hệ điều hành Android chưa chắc đã dễ sử dụng hơn các hệ điều hành khác (điều này tùy thuộc vào ý kiến chủ quan của mỗi người), nhưng vì nó đã quá phổ biến trên các thiết bị di động, nên chắc hẳn bạn không cảm thấy lạ lẫm gì với nó nữa. Bạn sẽ có 2 môi trường để làm việc trên các thiết bị Android, đó là màn hình chính (HomeScreen) và menu chính. Màn hình chủ giống như một ngôi nhà của bạn, gồm có các bức tường xung quanh mà bạn có thể thõa sức trang trí, treo đồng hồ, dán lịch, đặt máy nghe nhạc … Menu chính thì giống như trong hộc bàn của bạn, có đầy đủ những thứ bạn cần như: máy nhắn tin, máy nghe nhạc, điện thoại, máy chụp hình …



**b. Khuyết điểm**

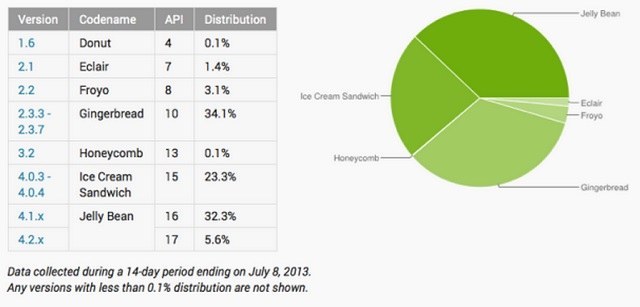
* Khả năng dọn rác kém

Sau một thời gian sử dụng thiết bị Android, các dữ liệu của ứng dụng đã được lưu trong bộ nhớ đệm, mà máy không có quá trình tự động để dọn dẹp chúng đi. Những dữ liệu này nếu không được dọn dẹp sẽ vẫn nằm ở đó và chiếm bộ nhớ của máy, đặc biệt là bộ nhớ RAM, vì vậy nếu sử dụng các thiết bị có bộ nhớ RAM thấp bạn sẽ thường xuyên gặp phải tình trạng giật, lag khi đang sử dụng. Tuy nhiên, nhược điểm này cũng có cách để khắc phục, nếu máy không tự “quét nhà” thì bạn sẽ phải “quét” bằng cách cài đặt các ứng dụng dọn dẹp như: History Eraser, Android System Cleaner, Clean MasterNhưng vấn đề giật, lag chỉ gặp ở những máy có cấu hình thấp, đối với những máy tầm trung hoặc cao cấp hiện nay thì hiện tượng đó rất hiếm.



* Qúa nhiều phiên bản dẫn đến việc phân mảnh

“Đem con bỏ chợ” là những gì mà đa phần người dùng than phiền Google khi sử dụng hệ điều hành Android phiên bản cũ và không được hỗ trợ nâng cấp lên phiên bản mới. Do tốc độ phát triển nhanh, kèm theo việc nâng cấp phiên bản thường xuyên dẫn đến việc có rất nhiều phiên bản Android trên thị trường. Thêm vào đó là rất nhiều hãng sử dụng hệ điều hành Android mà không thể nào hỗ trợ nâng cấp hết cho tất cả các thiết bị. Một phần cũng dễ hiểu vì họ luôn chạy theo và tập trung phát triển cho cái mới nên việc bỏ quên các thiết bị cũ là không hề tránh khỏi.



* Bảo mật

Do đặc điểm là mã nguồn mở, nên đây là điều kiện để những người có ý đồ xấu tìm hiểu và khai thác lỗ hổng bảo mật trên thiết bị Android. Ví dụ: trước đây có một ứng dụng tên là iCalendar là một ứng dụng lịch báo thông thường với những icon và hiệu ứng lạ mắt, nhưng ẩn bên dưới là những đoạn mã độc dùng để “móc túi” người dùng, những đoạn mã độc này sẽ tự động gửi tin nhắn về tổng đài đã được định sẵn khi chạm 5 lần trên màn hình mà người dùng không hề hay biết. Sau khi vụ việc được phát hiện thì ứng dụng đó đã bị gỡ bỏ khỏi Google Play. Dần dần qua các phiên bản, Google cũng đã cải thiện khả năng bảo mật, nhưng “liệu có còn tồn tại những lỗ hổng nữa không?” thì vẫn chưa thể trả lời chính xác được.



* Kho ứng dụng nhiều dẫn đến việc khó lựa chọn

Thật không dễ dàng để tìm kiểm một ứng dụng phù hợp trong kho lưu trữ khổng lồ Google Play Store. Khi bạn gõ một từ khóa tìm kiếm trên Play Store tiếp đó sẽ xuất hiện một danh sách thật dài các ứng dụng có cùng chức năng. Vì vậy việc lựa chọn không hề dễ dàng.Tuy nhiên, khuyết điểm này cũng có cách để khắc phục. Nếu bạn có thời gian, bạn có thể tự chọn lọc ứng dụng mình thích vì việc cài đặt và gỡ ra vô cùng dễ dàng, bạn chỉ việc kéo ứng dụng vào sọt rác là có thể gỡ nó ra nhanh chóng. Nếu không có thời gian bạn có thể tham khảo trên các diễn đàn công nghệ, có rất nhiều đề xuất về các ứng dụng hữu ích, chất lượng đã được “kiểm duyệt” bởi một số người dùng.



**2.2 SQLite trong Android**

**SQLite** là cơ sở dữ liệu mã nguồn mở, SQLite hỗ trợ chuẩn quan hệ cơ sở dữ liệu giống như hệ cơ sở dữ liệu SQL. SQLite yêu cầu giới hạn bộ nhớ trong thời gian thực thi là 250 kb. SQLite hỗ trợ chuẩn dữ liệu kiểu Text giống như kiểu dữ liệu String trong Java, Integer tương tự kiểu Long trong Java, và kiểu REAL tương tự kiểu Double trong Java. Tất cả các kiểu dữ liệu khác phải được chuyển đổi thành các kiểu dữ liệu này trước khi được lưu trong cớ sở dữ liệu SQLite không xác định các kiểu dữ liệu được ghi vào các cột thực tế của các kiểu được định nghĩa, có thể viết kiểu Integer trong cột String và ngược lại.

* **SQLite** được nhúng trong các thiết bị Android, sử dụng một cơ sở dữ liệu SQLite trong Android không yêu cầu thủ tục thiết lập hoặc quản lý cơ sở dữ liệu.

Bạn chỉ phải xác định các câu lệnh SQL để tạo và cập nhật cơ sở dữ liệu. Sau đó các cơ sở dữ liệu được quản lý tự động cho bạn bởi các nền tảng Android.

Truy cập vào một cơ sở dữ liệu SQLite liên quan đến việc truy cập vào hệ thống tập tin. Điều này có thể được làm chậm. Vì vậy nó được khuyến khích để thực hiện các hoạt động cơ sở dữ liệu không đồng bộ. Nếu ứng dụng của bạn tạo ra một cơ sở dữ liệu, cơ sở dữ liệu này là bởi mặc định được lưu trong thư mục DATA/data/APP\_NAME/databases/FILENAME.

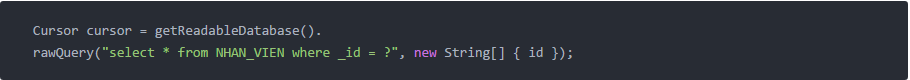
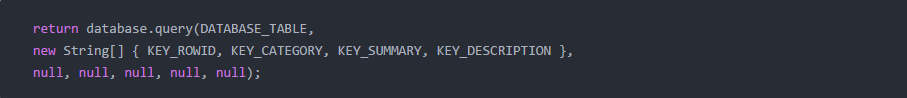
* **DATA:** Là đường dẫn mà bạn có thể get bởi câu lệnh Environment.getDataDirectory().
* **APP\_NAME:** Là tên ứng dụng của bạn.
* **FILENAME:** Là tên cơ dở dữ liệu mà bạn tạo trong ứng dụng

**2.2.1 Kiến trúc SQLite**

Để tạo và nâng cấp cơ sở dữ liệu trong ứng dụng Android của bạn, bạn tạo ra một lớp con của lớp SQLiteOpenHelper. Trong hàm khởi tạo của lớp con, bạn gọi super() phương thức của SQLiteOpenHelper, ghi tên cơ sở dữ liệu và phiên bản cơ sở dữ liệu hiện hành.Trong lớp này chúng ta cần ghi đè phương thức tạo và cập nhật cơ sở dữ liệu. Chúng ta sẽ có hai phương thức được ghi đè là:

* **onCreate():** Phương thức này được gọi bởi framework, cơ sở dữ liệu đã được truy cập nhưng chưa được tạo.
* **onUpgrade():** Phương thức này cho phép bạn cập nhật phiên bản mới cơ sở dữ liệu của bạn hoặc xóa cơ sở dữ liệu đã tồn tại và cập nhật cơ sở dữ liệu mới thông qua phương thức onCreate(). Cả hai phương thức này đều nhận đối tượng SQLiteDatabase làm tham số.

**2.2.2 Thao tác cơ sở dữ liệu với SQLiteDatabase**

* **SQLiteDatabase** là lớp cơ sở để làm việc với cơ sở dữ liệu SQLite trong Android và nó cung cấp các phương thức mở, truy vấn, cập nhật và đóng cơ sở dữ liệu, ngoài ra còn cung cấp phương thức execSQL() để truy vấn trực tiếp đến cơ sở dữ liệu.
* Sử dụng đối tượng ContentValues để định nghĩa key/values. Key định danh cho cột của bảng và value là giá trị của bản ghi của cột này. ContentValues sử dụng để insert và cập nhật dữ liệu trong bản ghi. Ngoài ra để truy vấn cơ sở dữ liệu chúng ta sử dụng phương thức rawQuery() và query() thông qua lớp SQLiteQueryBuilder.
* **SQLiteQueryBuilder** là lớp rất thuận tiện giúp truy vấn cơ sở dữ liệu.
* **rawQuery()** chấp nhận một câu lệnh lựa chọn như một đầu vào. 
* **query()** cung cấp cấu trúc cho câu lệnh SQL đơn giản.
* Các tham số của phương thức query()
* **String dbName**: tên bảng cơ sở dữ liệu
* **String[] columnNames**: Trả về danh sách cột của bảng mà bạn muốn, nếu \* "null" sẽ trả về tất cả các cột của bảng.
* **String whereClause**: Điều kiện chọn dữ liệu trả về, nếu tham số này là "null" thì sẽ trả về tất cả các dữ liệu.
* **String[] selectionArgs**: Chọn lọc dữ liệu với điều kiện mảng.
* **String[] groupBy**: Sẽ nhóm các dòng được trả về, nếu tham số này là "null" thì các dòng trả về sẽ không được nhóm.
* **String[] having**: Lọc cho các nhóm, "null" sẽ không lọc nhóm.
* **String[] orderBy**: dữ liệu trả về sẽ được sắp xếp, nếu "null" sẽ không sắp xếp.

**2.2.3 Cursor**

* Một truy vấn cơ sở dữ liệu **SQLite** luôn trả về một đối tượng **Cursor**.

Một Cursor đại diện cho kết quả của một truy vấn, sử dụng Cursor Android có thể đệm kết quả một cách hiệu quả, mà không cần load tất cả dữ liệu trong bộ nhớ.

* Để lấy số phần tử của kết quả truy vấn sử dụng phương thức **getCount().**
* Để di chuyển dữ liệu giữa các dòng riêng biệt bạn có thể sử dụng phương thức **moveToFirst()** và **moveToNext()**
* isAfterLast() là phương thức dùng để kiểm tra xem đã kết thúc câu lệnh truy vấn đã đạt được.
* **Cursor** cung cấp phương thức **get()** kiểu dữ liệu như : **getLong(columnIndex),**  **getString(columnIndex)** để truy cập dữ liệu trong cột tại vị trí hiện tại, trong đó columnIndex là số cột bạn truy cập.
* **Cursor** cũng cung cấp các phương thức **getColumnIndexOrThrow (String)** để lấy chỉ số cột cho một tên cột của bảng, sau khi sử dụng Cursor phải đóng với phương thức close().

**2.3 Intent**

* + - Là những tin nhắn không đồng bộ cho phép các thành phần ứng dụng yêu cầu chức năng từ các thành phần Android khác.
    - **Intents** cho phép bạn tương tác với các thành phần từ các ứng dụng như nhau cũng như với các thành phần đóng gói trong một ứng dụng khác.

**VD**:một Activity có thể gọi một Activity bên ngoài để chụp ảnh.

* + - **Intent** là đối tượng của class **android.content.Intent**.
    - Mã của bạn có thể gửi **Intent** vào hệ thống Android với chỉ định thành phần mục tiêu gửi đến.

**VD:** thông qua phương thức ***startActivity()***, bạn có thể định nghĩa một **Intent** sử dụng để gọi chạy một **Activity** khác. Tại **Activity** mục tiêu, thông qua phương thức ***startActivity()*** bạn có thể xác định được ý định của người gửi đến để bắt đầu (start) **Activity** này.

* + - Một **Intent** có thể chứa dữ liệu thông qua một **Bundle**. Dữ liệu này có thể được sử dụng bởi các thành phần tiếp nhận.
    - Các thuộc tính của **Intent**:
      * **Action**: là hành động được thực hiện,

**VD : ACTION\_VIEW, ACTION\_MAIN**

* + - * **Data:** là dữ liệu sẽ được xử lý trong action, thường được diễn tả là một Uri (Uniform Resource Identifier).

**VD**: **ACTION\_VIEW** content://contacts/people/1 - Hiển thị thông tin về người với mã danh 1.

**ACTION\_DIAL** content://contacts/people/1 - Hiển thị màn hình gọi đến người với mã danh 1.

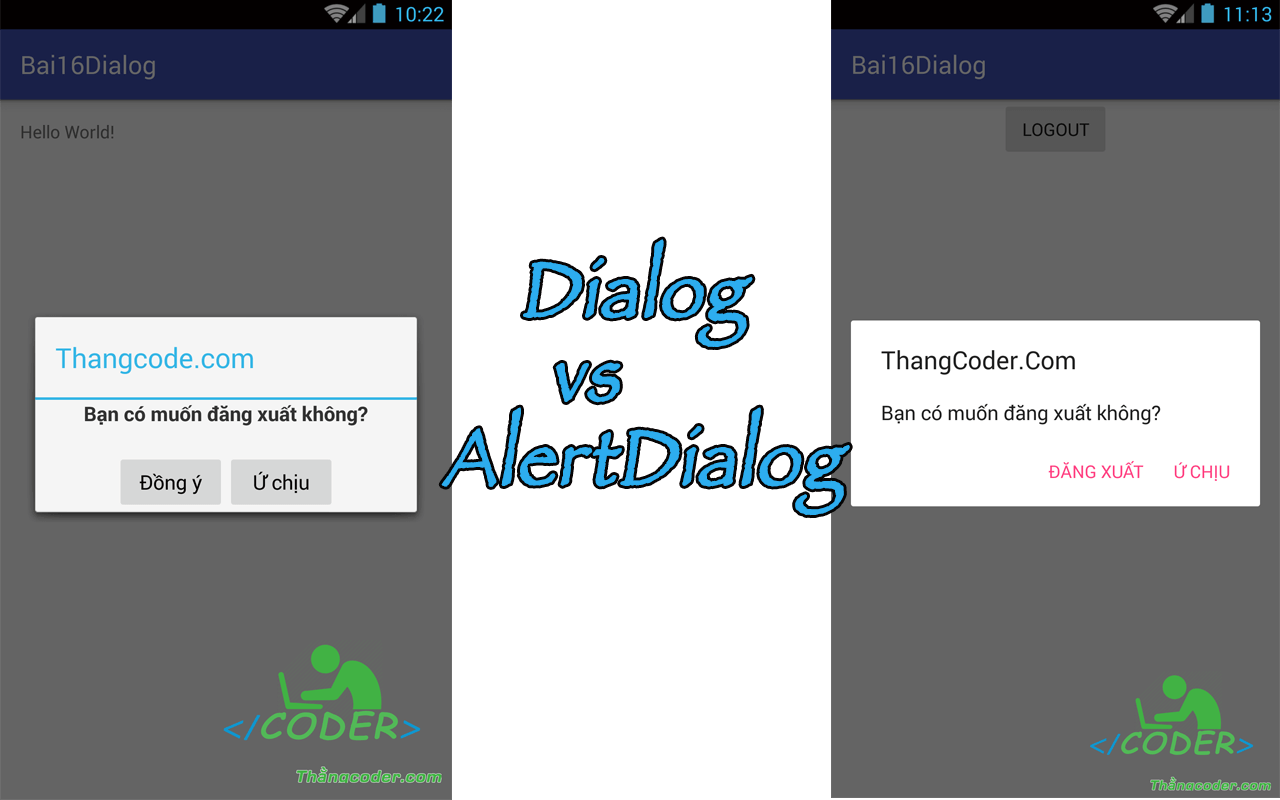
**ACTION\_DIAL** tel:123 - Hiển thị màn hình gọi với số gọi là 123.

* + - Ngoài ra còn có một số thuộc tính ta có thể thêm vào **Intent:**
      * **Category:** bổ sung thêm thông tin cho action của Intent. VD: **CATEGORY\_LAUNCHER** thông báo sẽ thêm vào Launcher như là một ứng dụng top-level.
      * **Type**: chỉ rõ kiểu của data.
      * **Component**: chỉ rõ thành phần sẽ nhận và xử lý **Intent**. Khi thuộc tính này được xác định thì các thuộc tính khác sẽ trở thành thuộc tính phụ.
      * **Extras**: mang theo đối tượng **Bundle** chứa các giá trị bổ sung.
    - **Intent** có thể sử dụng để :
      * Bắt đầu một Activity.
      * Bắt đầu một Activity con .
      * Bắt đầu một dịch vụ (Service).
    - **Các loại Intent phổ biến:**
* **Explicit Intents (Intent tường minh)**: **Intent** đã được xác định thuộc tính component, nghĩa là đã chỉ rõ thành phần sẽ nhận và xử lý intent. Thông thường **Intent** dạng này sẽ không bổ sung thêm các thuộc tính khác như action, data. **Explicit Intent** thương được sử dụng để khởi chạy các activity trong cùng 1 ứng dụng.
* **Implicit Intents (Intent không tường minh)**: **Intent** không chỉ rõ component xử lý, thay vào đó nó bổ sung thông tin trong các thuộc tính. Khi **Intent** được gửi đi, hệ thống sẽ dựa vào những thông tin này để quyết định component nào thích hợp nhất để xử lý nó.

**2.4 Dialog**

**Dialog** là gì?

Nếu bạn học bất kì ngôn ngữ nào thì chắc cũng đụng đến thằng **Dialog** cả rồi đúng không, xưa ở trường học VB hay C# gì đó thì gọi là **Box Message** còn bên Java thì hay gọi là Dialog.Gọi là gì thì ý nghĩa của nó cũng là show lên một thông báo trên màn hình, và thông báo này sẽ che mờ màn hình, bạn cũng có thể xử lý sự kiện trên Dialog này.



Dialog chính trong Android đó là **Dialog** và **AlertDialog**:

**Dialog**:

Đây là loại Dialog mà bạn phải tạo giao diện cho nó bằng xml chứ mặc định nó không có giao diện ngoài cái title, hay nói cách khác bạn phải custom nó từ đầu tới cuối, không giống AlertDialog bạn sẽ thêm được title, message, button…

* + - **dialog = new Dialog(MainActivity.this)**: đây là khởi tạo dialog
    - **dialog.setTitle(“Thangcode.com”):** ở đây bạn sẽ đặt tên tiêu đề cho Dialog nhé
    - **dialog.setContentView(R.layout.dialog):** đây chính là dòng quan trọng và quyết định đây, chắc bạn không thể nào quên phương thức setContentView() mà bất kì activity nào cũng có để tạo giao diện đó chứ.Ở đây có nghĩa là chúng ta sẽ set giao diện vào dialog, giao diện này bạn đã định nghĩa ở file layout dialog.xml trước đó rồi nhé.
    - **dialog.show():** tất cả đã xong thì bây giờ bạn chỉ cần gọi phương thức show() để dialog hiển thị lên là được

**AlertDialog**:

**AlertDialog** là một kiểu Dialog hay nó cách khác nó là lớp con kế thừa từ Dialog ra, nhưng có thể hiển thị nhiều các Button bên dưới mà chúng ta không cần phải định nghĩa layout cho nó.Ngoài ra bạn muốn hiển thị tiêu đề, nội dung bên trong thì **AlertDialog** đều hỗ trợ bạn nhanh chóng.

* + - **AlertDialog.Builder builder = new AlertDialog.Builder(this):** đây là hàm khởi tạo một AlertDialog.Builder, thằng này có tác dụng vẽ ra giao diện cho AlertDialog, mặc định hàm trên nó sẽ sử dụng theme dialog mặc định khi show và trong video bên dưới mình sẽ nói rõ hơn chỗ này, giúp bạn tuỳ chỉnh theme cho đẹp hơn.
    - **builder.setTitle():** ở đây là set tiêu đề cho AlertDialog, cái này tương tự như Dialog không có gì khác cả.
    - **builder.setMessage():** đây là set nội dung bên trong cho Alert, thằng Dialog thì không có phương thức này nhé.
    - **builder.setCancelable():** đây là thuộc tính nếu set false thì khi show dialog lên người dùng click ra bên ngoài dialog thì nó vẫn không bị mất, nếu set true thì sẽ mất khi click vào bất kì đâu ngoài dialog.

**2.5Permissions (quyền):**

* + - * Permission được dùng để giới hạn quyền truy xuất vào chương trình mà dữ liệu trên thiết bị. Những giới hạn này được áp đặt để bảo vệ dữ liệu nhạy cảm và các chương trình có thể gây lỗi hoặc cố ý dùng sai của người dùng có kinh nghiệm.
      * Mỗi quyền được xác định bởi 1 chuỗi duy nhất. Thường mỗi chuỗi xác định duy nhất 1 quyền. Dưới đây là một số quyền được mô tả bởi sẳn Android.
        + android.permission.CALL\_EMERGENCY\_NUMBERS
        + android.permission.READ\_OWNER\_DATA
        + android.permission.SET\_WALLPAPER
        + android.permission.DEVICE\_POWER
      * Mỗi tính năng của ứng dụng có thể được bảo vệ bằng một quyền.
      * Nếu ứng dụng cần truy xuất vào một tính năng được bảo vệ bằng quyền truy xuất, nó phải mô tả quyền cần để truy xuất trong tập tin manifest bằng thẻ <uses-permission>.
      * Khi ứng dụng được cài đặt vào thiết bị, trình cài đặt sẽ kiểm tra xem có thể cấp quyền được yêu cầu bằng cách kiểm tra chứng chỉ xác thực của ứng dụng, nếu không cấp quyền được, trình cài đặt có thể yêu cầu xác nhận của người dùng.
      * Nếu đã được ủy quyền, ứng dụng có thể truy xuất vào các tính năng được bảo vệ, nếu không được ủy quyền, khi ứng dụng sẽ không truy cập được các tính năng được bảo vệ mà không có bất kỳ thông báo nào với người dùng.
      * Ứng dụng có thể bảo vệ các component của nó (activities, services, broadcast receivers và content providers) bằng quyền.
      * Nó có thể sử dụng bất kỳ quyền nào đã được mô tả sẳn bởi Android hoặc được khai báo trong ứng dụng khác.
      * Hoặc nó cũng có thể tự mô tả quyền của mình. Một quyền mới sẽ được khai báo bằng thẻ <permission>.
      * Ví dụ: ứng dụng có thể tự bảo vệ activity của mình bằng quyền như sau:

<manifest . . . >

<permission android:name="com.example.project.DEBIT\_ACCT" . . . />

<uses-permission android:name="com.example.project.DEBIT\_ACCT" />

. . .

<application . . .>

<activity android:name="com.example.project.FreneticActivity"

android:permission="com.example.project.DEBIT\_ACCT"

. . . >

. . .

</activity>

</application>

</manifest>

* + - * Chú ý rằng trong ví dụ trên, quyền **DEBIT\_ACCT** không chỉ được khai báo bằng thẻ <permission>, nó còn được sử dụng để yêu cầu quyền với thẻ <users-permission>.
      * Trong ví dụ trên, nếu thuộc tính permission dùng quyền khác, như: android.permission.CALL\_EMERGENCY\_NUMBERS, thì không không cần khai báo quyền này với thẻ <permission>. Chỉ cần khai khai báo quyền này trong thẻ <uses-permission>.
    - Library:
      * Mỗi ứng dụng sẽ cần liên kết với những thư viện có sẳn của Android, chứa các gói cơ bản cho việc biên dịch ứng dụng.
      * Tuy nhiên, nếu ứng dụng sử dụng những package nằm ngoài thư viện có sẳn của Android, bạn cần phải liên yêu cần được liên kết với chúng. Thẻ <uses-library> trong tập tin manifest sẽ thực hiện công việc này.

**2.6 Mã hóa một chiều MD5**

– Mã hóa một chiều hiểu đơn giản theo đúng nghĩa của nó, tức là một khi đã mã hóa rồi thì không thể giải ngược lại được nữa.

– Đây là phương thức giúp bảo vệ tuyệt đối dữ liệu.

– Dùng để lưu mật khẩu vào Database

- Ví dụ:

– Khi bạn tạo một password là 123123 chẳng hạn, mật khẩu này sẽ được mã hóa 1 chiều thành 1 đoạn code và lưu trong cơ sở dữ liệu.

– Khi bạn nhập mật khẩu vào để đăng nhập, mật khẩu sẽ tự động mã hóa và so sánh với mã hóa đã lưu trong cơ sở dữ liệu. Nếu giống nhau thì đăng nhập được.

– Trong trường hợp nếu database bị hack thì hacker cũng chỉ thấy một đoạn code đã được mã hóa mà không thể decode được.

**Mã hóa MD5**

– MD5 được dùng trong bảo mật và để kiểm tra độ toàn vẹn của tập tin.

– MD5 được thiết kế bởi ông Ronald Rivest năm 1991 mục đích là thay thế cho hàm băm MD4 trước đó. Vào năm 1996, một lỗ hổng trong MD5 đã được phát hiện; mặc dù không biết lỗ hổng đó có nghiêm trọng hay không nhưng một mã hóa khác đã được đề xuất, như SHA-1 (lúc đó cũng được coi là không an toàn). Đến năm 2004, nhiều lỗ hổng của hệ bảo mật này bị phát hiện khiến cho nó được đặt vào vòng nghi vấn.

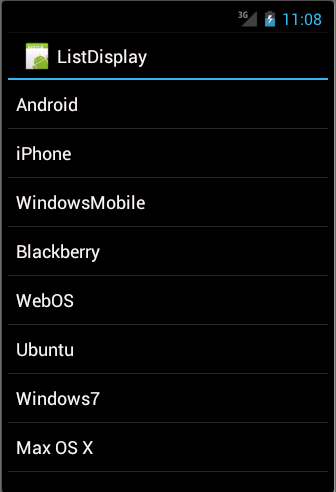
– Vì MD5 chỉ giúp dò qua dữ liệu 1 lần, nếu có hai tiền tố với cùng bảng băm được dựng nên, thì cùng 1 hậu tố có thể cùng được thêm vào một lúc để khiến cho đụng độ có thể dễ xảy ra. Tức là hai dữ liệu đầu vào (input) X và Y hoàn toàn khác nhau nhưng có thể đầu ra (output) cùng được một md5 hash như nhau . Tuy nhiên xác suất khiến điều này xảy ra khá nhỏ

– Vì những kỹ thuật tìm đụng độ hiện nay cho phép các trạng thái băm trước đó được xác định một cách ngẫu nhiên, có thể tìm thấy xung đột đối với bất kỳ tiền tố mong muốn nào; có nghĩa là, đối bất kỳ một chuỗi các ký tự X cho trước, hai tập tin đụng độ có thể được xác định mà đều bắt đầu với X.

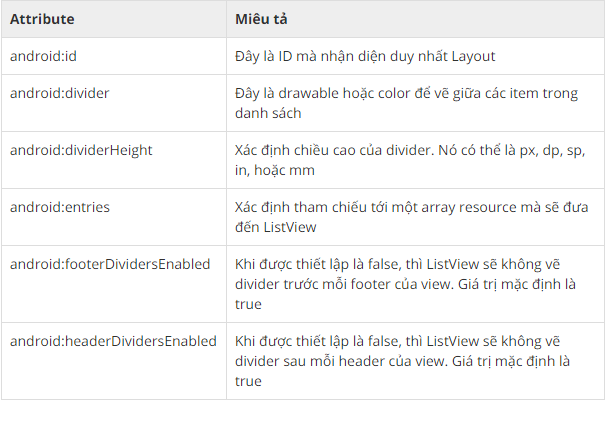
– Tất cả những gì cần để tạo ra hai tập tin đụng độ là một tập tin mẫu, với một khối dữ liệu 128 byte được xếp trên giới hạn 64 byte, có thể thay đổi tự do bằng giải thuật tìm va chạm.

**2.7 List View**

* ListView trong Android là một view mà nhóm một số item và hiển thị chúng trong danh sách theo chiều dọc có thể scroll. Các item của danh sách này được tự động chèn vào danh sách bởi sử dụng một Adapter mà kéo nội dung từ một nguồn như một mảng hoặc cơ sở dữ liệu.



* Một Adapter thực sự là cầu nối giữa các thành phần UI và nguồn dữ liệu mà điền dữ liệu vào trong thành phần UI. Adapter giữ dữ liệu và gửi dữ liệu tới Adapter View, từ đó view có thể lấy dữ liệu từ Adapter View đó và hiển thị dữ liệu trên các view khác nhau ở dạng Spinner, List View, Grid View, …
* ListViewvà GridView là các lớp con của AdapterView . Gắn kết chúng với mộtAdapter, mà thu nhận dữ liệu từ một nguồn ngoại vi và tạo một View để biểu diễn mỗi entry.
* Android cung cấp một số lớp con hữu ích của lớp Adapter để thu lấy các kiểu dữ liệu khác nhau và xây dựng các view cho một AdapterView (ví dụ ListView hoặc GridView). Các Adapter phổ biến làArrayAdapter,Base Adapter, CursorAdapter, SimpleCursorAdapter,SpinnerAdapter và WrapperListAdapter
* Các thuộc tính cơ bản của ListView



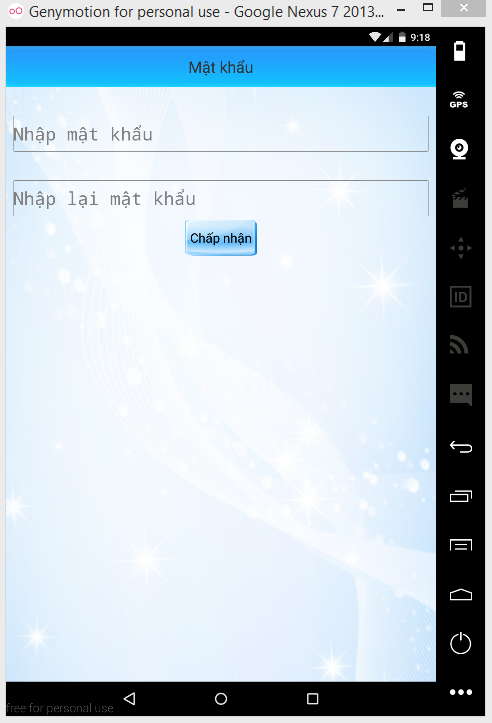
**ArrayAdapter trong Android**

Bạn có thể sử dụng Adapter này khi nguồn dữ liệu là một mảng. Theo mặc định, ArrayAdapter tạo một view cho mỗi item trong mảng bằng việc gọi toString() trên mỗi item và đặt nội dung vào trong một TextView. Giả sử bạn có một mảng các chuỗi và muốn hiển thị trong một ListView, bạn khởi tạo một ArrayAdapter mới sử dụng một Constructor để xác định Layout cho mỗi chuỗi và mảng chuỗi.

**3.Đặc tả và thiết kế hệ thống.**

**3.1 giới thiệu chi tiết từng chức năng:**

**3.1.1. Chức năng tạo mật khẩu.**



Nhập mật khẩu mà bạn muốn tạo.

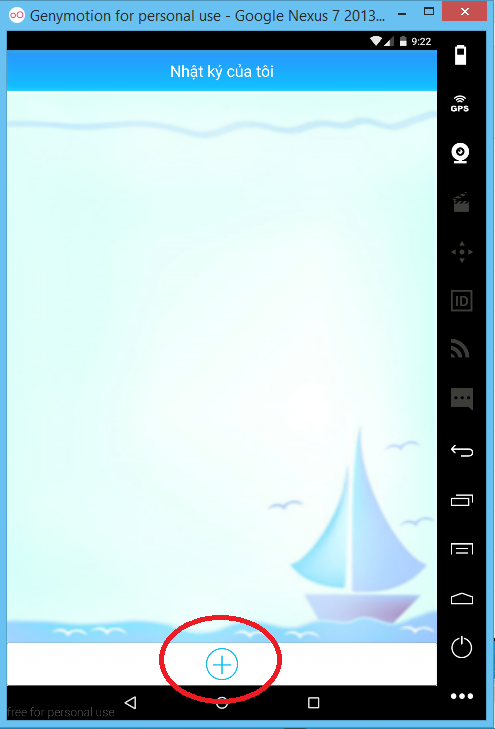
Nhập lại mật khẩu phải trùng khớp với mật khẩu đã nhập.

Mật khẩu được mã hóa MD5.

Khi chấp nhận, mật khẩu được lưu vào SQLite và database.

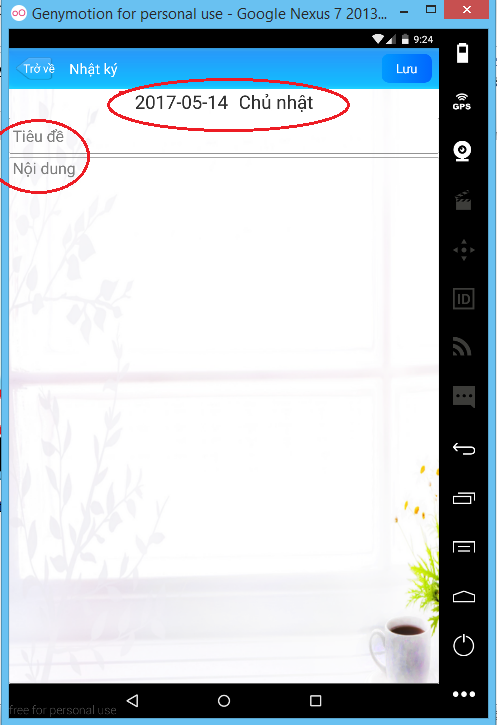
Khi truy cập lại ứng dụng, sẽ phải nhập lại mật khẩu:

**3.1.2 Chức năng tạo nhật ký**



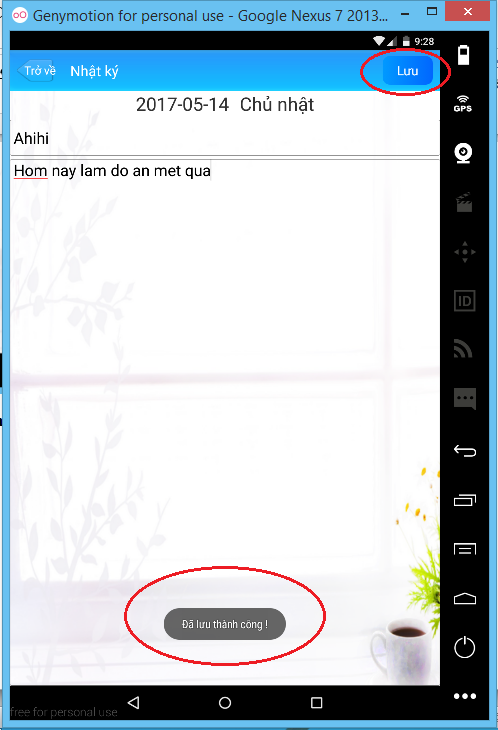
Khi nhấn vào Image Button sẽ tạo nhật ký mới:

**3.1.3 Cập nhật ngày tháng hiện tại**

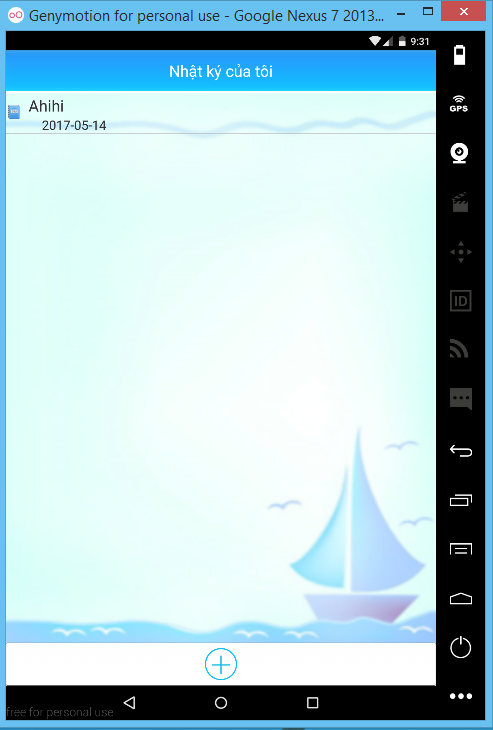


**3.1.4 Chức năng Lưu nhật ký**

Sau khi nhập Tiêu đề và nội dung. Bạn mới lưu được nhật ký.

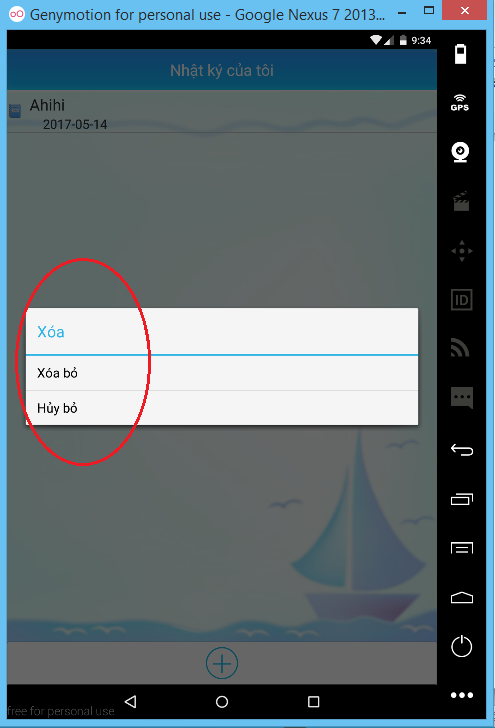
****

Sau khi lưu nhật ký bấm nút trở về nhật ký đã được lưu.



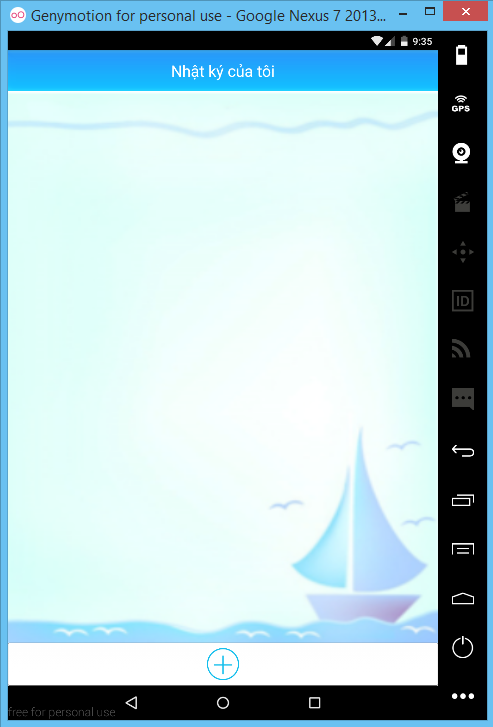
**3.1.5 Chức năng xóa nhật ký:**

Nhấp đè vào nhật ký, bạn sẽ thấy dialog thông báo có muốn xóa nhật ký.



Nhấp xóa, nhật ký sẽ biến mất.

Đây là kết quả sau khi xóa.

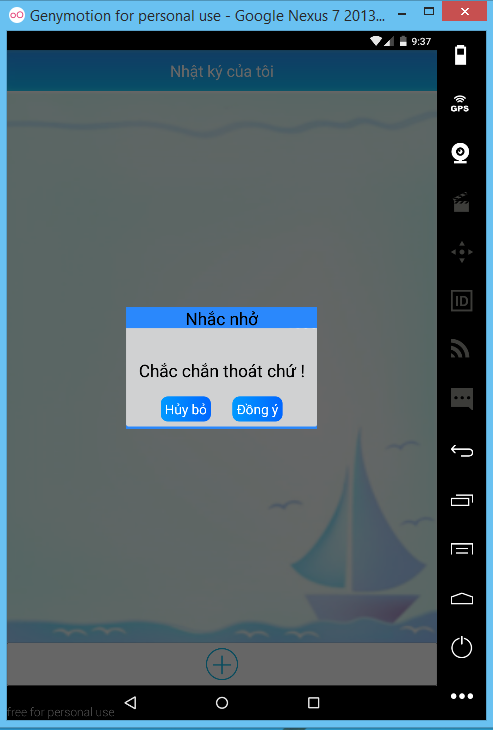


**3.1.6 Chức năng hỏi người dùng trước khi thoát.**

Tạo dialog thông báo khiến màn hình sẫm màu lại.

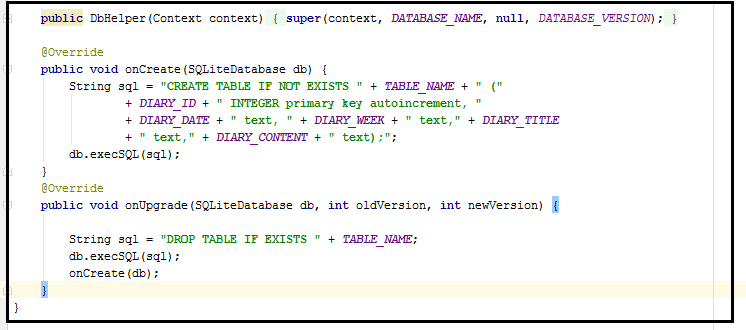
Khi nhấn **đồng ý**, thì thoát khỏi ứng dụng và lưu lại toàn bộ dữ liệu trong database.

Khi nhấn **hủy bỏ**, quay lại màn hình chính của ứng dụng.

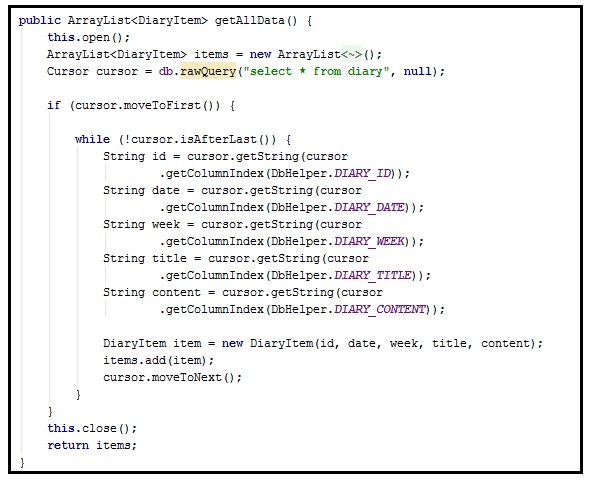


**3.2 Thiết kế hệ thống**

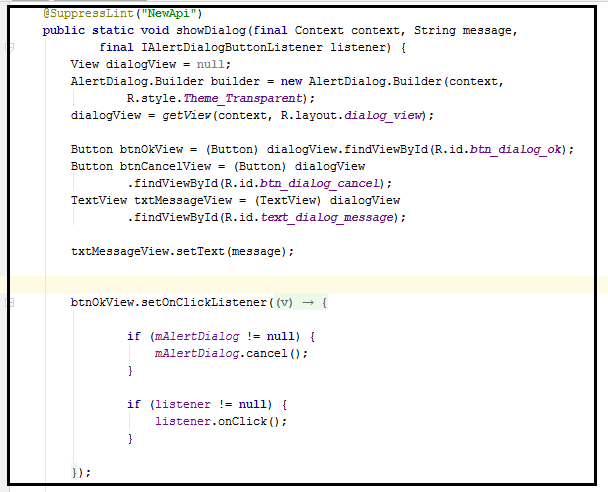
**3.2.1 Thiết kế DataBase**



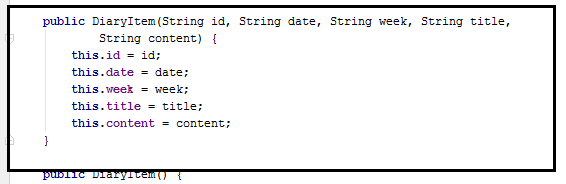
**3.2.2 Lấy dữ liệu từ Database**



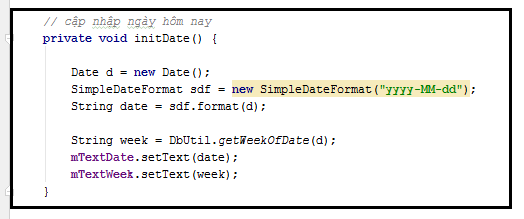
**3.2.3 Tạo và showDialog().**



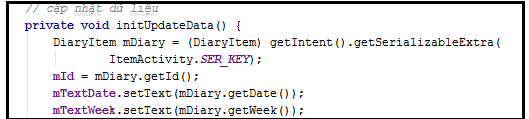
**3.2.4 Tạo cơ sở dữ liệu ngày tháng.**



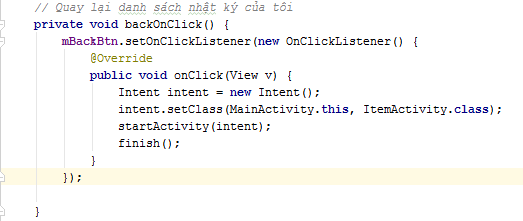
**3.2.5 Cập nhật ngày tháng**

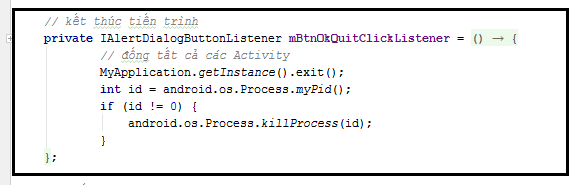


**3.2.6 Cập nhật dữ liệu**



**3.2.7 Một số Intent qua lại giữa các Activity**







**4. Kết quả thực nghiệm :**

Chạy được. Không có bug. Ứng dụng mượt mà không giật. Đạt được số chức năng như phần đặc tả hệ thống.

**5. Kết luận và hướng phát triển.**

**5.1 Đã làm được:**

Tuy ứng dụng còn hạn chế về chức năng nhưng vẫn hoàn thành được chức năng chính của nhật ký:

\_Tạo mật khẩu bảo mật.

\_Lưu nhật ký vào database.

\_Thêm, xóa, sửa nhật ký.

5.2 Chưa làm được:

\_Icon vui nhộn.

\_Màu, font chữ.

Hướng phát triển :

Cập nhật các chức năng vui nhộn khác như icon, đổi màu, phong chữ, vv