**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA ĐIỆN – ĐIỆN TỬ**

**-----🙞🙜🕮🙞🙜-----**

**Logo

Description automatically generated**

**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG**

**TODO&REMINDER**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện** | **:** | **Nguyễn Thiện Hùng** |
| **Mã sinh viên** | **:** | **181411580** |
| **Lớp** | **:** | **Kỹ thuật điện tử & Tin học công nghiệp K59-02** |
| **Giảng viên hướng dẫn** | **:** | **Ts. Trần Mạnh Cường** |

**Hà Nội, Năm 202****2**

|  |  |
| --- | --- |
| KHOA ĐIỆN – ĐIỆN TỬ  BỘ MÔN KỸ THUẬT ĐIỆN TỬ | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập – Tự do – Hạnh phúc |

**BẢNG DUYỆT ĐỒ ÁN VÀ XÁC NHẬN**

**CHO SINH VIÊN NỘP ĐỒ ÁN TỐT NGHIỆP**

**Họ và tên sinh viên:Nguyễn Thiện Hùng** **Mã SV:181411580**

**Tên đề tài:ToDo&Reminder App**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Hạng mục** | **Nội dung** | | **Đánh giá** | |
| **ĐẠT** | **CHƯA ĐẠT** |
| **1** | **Bố cục đồ án** | Bố trí trang bìa, trang lót, nhiệm vụ thiêt kế TN, đúng mẫu quy định | |  |  |
| Có lời mở đầu, lời cảm ơn | |  |  |
| Mục lục chính xác, đúng mẫu | |  |  |
| Danh mục bảng biểu, hình vẽ đầy đủ, chính xác | |  |  |
| Danh mục cụm từ viết tắt đầy đủ, chính xác | |  |  |
| Bố cục các chương đúng mẫu, có kết luận của mỗi chương | |  |  |
| Phụ lục trình bày hợp lý (nếu có) | |  |  |
| Danh mục tài liệu tham khảo đủ | |  |  |
| **2** | **Nội dung đồ án** | Nội dung cúa đồ án đã được GVHD duyệt, và kết luận:  **◻** Đồng ý cho SV nộp đồ án cho Bộ môn để bảo vệ  **◻** Không đồng ý cho SV nộp đồ án. | | | |
| **3** | **Điểm hướng dẫn đồ án tốt nghiệp:** |  | | | |
|  | | | ***Hà Nội , ngày … tháng … năm 20…***  **Giáo viên hướng dẫn**  *(Ký và ghi rõ họ tên)* | | | |

# LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

*Hà Nội, ngày tháng năm 2022*

Nguyễn Thiện Hùng

# LỜI CẢM ƠN

Để hoàn thành chuyên đề báo cáo đồ án tốt nghiệp này trước hết em xin gửi đến quý thầy, cô trong khoa Điện- Điện tử lời cảm ơn chân thành.

Em xin được gửi lời cảm ơn đến người anh mentor của em tại SamSung SRV, anh Nguyễn Văn Vinh đã hướng dẫn em trong suốt 3 tháng thực tập và làm việc. Anh là người đã dạy tôi những điều đầu tiên về Android,... và cũng là người hướng dẫn em học tập và làm việc có hiệu quả

Đặc biệt, em xin gửi đến thầy TS.Trần Mạnh Cường người đã tận tình hướng dẫn, giúp đỡ em hoàn thành chuyên đề báo cáo đồ án tốt nghiệp này lời cảm ơn sâu sắc nhất.

Với điều kiện thời gian cũng như kiến thức và hiểu biết còn hạn chế của một sinh viên, bài báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn cho đề tài đồ án phát triển sau này.

Em xin chân thành cảm ơn!

*Hà Nội, ngày tháng năm 2022*

*Sinh viên thực hiện*

Nguyễn Thiện Hùng

# TÓM TẮT

Todo&Reminder app là một yêu cầu thiết yếu của mỗi cá nhân trong một xã hội phát triển. Thời đại công nghệ thông tin phát triển đòi hỏi mỗi người phải làm việc với cường dộ cao và tập chung, để công việc và mọi hoạt động được sắp xếp khoa học và hiệu quả không bị chồng chéo, đúng thời gian, tiến độ, đảm bảo yêu cầu mà mỗi người vẫn thấy được sự thoải mái và niềm vui của cuộc sống. Nhưng chủ yếu sự sắp xếp đó là do mỗi cá nhân nhưng nhờ có sự phát triển của công nghệ thông tin và ứng dụng vô cùng phong phú và hiệu quả trong cuộc sống đã tạo ra những tiện ích của nó giúp con người giải quyết những khó khăn. Từ yêu cầu thực tế ma một phần mềm chuyên dụng lập danh sách nhiệm vụ cá nhân cho mọi người là một yêu cầu chính đáng và thiết thực.

Hiện tại có nhiều phần mềm lập nhiệm vụ khá phong phú đáp ứng được một số yêu cầu của người dùng khi lên lịch cá nhân của mình cho mọi hoạt động. Nó có nhiều chức năng nhưng đa số chưa thực sự chuyên dụng tới đa số mọi người. Yêu cầu đặt ra là có một phần mềm chuyên dụng hơn ,sát với nhiều đối tượng người dùng hơn ,chỉ chuyên dụng là lập nhiệm vụ làm việc, hỗ trợ người dùng lập ra một lịch biểu khoa học hiệu quả và nhắc nhở người dùng thực hiện mọi công việc. Để có thể hỗ trợ người dùng thì đòi hỏi người dùng dành khoảng thời gian nhất định để nhập thông tin liên quan ,sửa đổi ,cập nhập ,tra cứu và phần mềm dựa trên sự cập nhập của bạn sẽ nhắc việc cho bạn và thống kê cho bạn những công việc ở mức độ hoàn thành như thế nào .

Trên cơ sở nghiên cứu đề tài, khảo sát thực tế các phần mềm đã có như OUTLOOK trong bộ OFFICE, phần mềm lập lịch CORE DestopSetupFull, lịch cá nhân Sunbird Portable (Mozilla Sunbird), phần mềm nhắc lịch bằng tiếng nói YOUR VOICE REMINDER, phần mềm nhắc tiếng Rmp2Full và một số phần mềm khác. Mỗi phần mềm có một đặc điểm riêng và mặt mạnh riêng song đa số đối với người dùng vẫn còn khoảng cách đó là tất cả các mục lục đều ghi bằng ngôn ngữ tiếng anh nên không phù hợp với đa số người Việt. Đó là một hạn chế lớn nhất đối với người dùng vì khi một người dùng muốn sử dụng hiệu quả thì họ phải biết Tiếng Anh một yêu cầu khá phức tạp. Qua đó chúng tôi xin nói về phần mềm này chỉ mang tính hỗ trợ trên cơ sở phân tích các chức năng chủ yếu và yêu cầu thực tế người dùng. Sau đây là bản phân tích thiết kế của chúng em rất mong được sự đóng góp của quý thầy cô và các bạn sinh viên

# MỤC LỤC

[LỜI CAM ĐOAN iii](#_Toc123582200)

[LỜI CẢM ƠN iv](#_Toc123582201)

[TÓM TẮT v](#_Toc123582202)

[MỤC LỤC vii](#_Toc123582203)

[DANH MỤC VIẾT TẮT ix](#_Toc123582204)

[DANH MỤC HÌNH ẢNH x](#_Toc123582205)

[DANH MỤC BẢNG BIỂU xii](#_Toc123582206)

[LỜI MỞ ĐẦU xiii](#_Toc123582207)

[NỘI DUNG NGHIÊN CỨU 1](#_Toc123582208)

[CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDRIOD 1](#_Toc123582209)

[1.1 Tìm hiểu chung về hệ điều hành Android 1](#_Toc123582210)

[1.1.1 Khái niệm Android 1](#_Toc123582211)

[1.1.2 Lịch sử Android 1](#_Toc123582212)

[1.1.3 Một số đặc trưng của hệ điều hành Android 18](#_Toc123582213)

[1.1.4 Một số tính năng hỗ trợ sẵn trong hệ điều hành Android 19](#_Toc123582214)

[1.1.5 Cấu trúc của hệ điều hành Android 19](#_Toc123582215)

[1.1.6 Máy ảo Dalvik 24](#_Toc123582216)

[1.1.7 Các thành phần trong một ứng dụng Android 25](#_Toc123582217)

[1.1.8 Công cụ phát triển hệ điều hành Android 29](#_Toc123582218)

[1.2 Quản trị CSDL với SQLite 31](#_Toc123582219)

[1.2.1 Quản trị cơ sở dữ liệu 31](#_Toc123582220)

[1.2.2 Thiết kế của SQLite 31](#_Toc123582221)

[CHƯƠNG 2: XÁC ĐỊNH YÊU CẦU VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG 35](#_Toc123582222)

[2.1 Mục tiêu và phạm vi của hệ thống 35](#_Toc123582223)

[2.1.1 Phạm vi ứng dụng 35](#_Toc123582224)

[2.1.2. Mục tiêu của ứng dụng 35](#_Toc123582225)

[2.2 Phân tích thiết kế hệ thống 35](#_Toc123582226)

[2.2.1 Biểu đồ tổng quan chức năng của hệ thống 35](#_Toc123582227)

[2.2.2 Biểu đồ ca sử dụng 36](#_Toc123582228)

[2.3 Thiết kế cơ sở dữ liệu 48](#_Toc123582229)

[2.3.1 Cấu trúc bảng cơ sở dữ liệu 48](#_Toc123582230)

[CHƯƠNG 3: THIẾT KẾ DỮ LIỆU VÀ XÂY DỰNG CHƯƠNG TRÌNH 49](#_Toc123582231)

[3.1 Thiết kế MODUL 49](#_Toc123582232)

[3.1.1 Công cụ sử dụng 49](#_Toc123582233)

[3.1.2 Thông tin phần mềm 49](#_Toc123582234)

[3.1.3 Bảng một số modul chính trong chương trình 50](#_Toc123582235)

[3.2 Thiết kế giao diện chương trình 51](#_Toc123582236)

[3.2.1 Giao diện chính của chương trình 51](#_Toc123582237)

[3.2.2 Nội dung của một nhiệm vụ 52](#_Toc123582238)

[3.2.3 Thông báo của một nhiệm vụ 52](#_Toc123582239)

[3.2.4 Tìm kiếm nhiệm vụ 53](#_Toc123582240)

[3.2.5: Lọc các nhiệm vụ 54](#_Toc123582241)

[3.2.6: Thêm nhiệm vụ 55](#_Toc123582242)

[3.2.7: Đa dạng đầu vào 56](#_Toc123582243)

[3.2.8: Sửa nhiệm vụ 58](#_Toc123582244)

[KẾT LUẬN 59](#_Toc123582245)

[DANH MỤC TÀI LIỆU THAM KHẢO 60](#_Toc123582246)

[PHỤ LỤC 61](#_Toc123582247)

[Phụ lục code: 61](#_Toc123582248)

# DANH MỤC VIẾT TẮT

|  |  |
| --- | --- |
| **API** | Application Programming Interface |
| **SDK** | Software Development Kit |
| **PC** | Personal Computer |
| **OTA** | Over The Air |
| **CDMA** | Multiple Division Access Division |
| **HTML** | Hypertext Markup Language |
| **GPS** | Global Positioning System |
| **IBM** | International Business Machines |
| **ADT** | Android Development Tools |
| **JDT** | Java Development Tools |
| **WIFI** | Wireless Fidelity |
| **CSDL** | Cơ sở dữ liệu |

# DANH MỤC HÌNH ẢNH

[*Hình 1. 1 Minh họa Android. 1*](#_Toc122906162)

[*Hình 1. 2 Phiên bản Android 1.0 2*](#_Toc122906163)

[*Hình 1. 3 Giao diện được cải tiến dành riêng cho điện thoại di động 3*](#_Toc122906164)

[*Hình 1. 4 Minh họa giao diện Android 1.1 3*](#_Toc122906165)

[*Hình 1. 5 Phiên bản Android 1.5 4*](#_Toc122906166)

[*Hình 1. 6 Phiên bản Android 1.6 5*](#_Toc122906167)

[*Hình 1. 7 Minh họa giao diện Android 1.6 5*](#_Toc122906168)

[*Hình 1. 8 Phiên bản Android 2.0 6*](#_Toc122906169)

[*Hình 1. 9 Minh hoạn giao diện Android 2.0 7*](#_Toc122906170)

[*Hình 1. 10 Phiên bản Android 2.2 8*](#_Toc122906171)

[*Hình 1. 11 Phiên bản Andoid 2.3 9*](#_Toc122906172)

[*Hình 1. 12 Phiên bản Andoid 3.0 10*](#_Toc122906173)

[*Hình 1. 13 Phiên bản Andoid 4.0 11*](#_Toc122906174)

[*Hình 1. 14 Phiên bản Andoid 4.1 11*](#_Toc122906175)

[*Hình 1. 15 Minh họa phiên bản Android 4.3 Jelly Bean 13*](#_Toc122906176)

[*Hình 1. 16 Phiên bản Andoid 4.4 14*](#_Toc122906177)

[*Hình 1. 17 Minh họa giao diện Android 4.4 15*](#_Toc122906178)

[*Hình 1. 18 Phiên bản Andoid 5.0 16*](#_Toc122906179)

[*Hình 1. 19 Tỷ lệ sử dụng các phiên bản 18*](#_Toc122906180)

[*Hình 1. 20 Tầng Linux Kernel trong kiến trúc Android 19*](#_Toc122906181)

[*Hình 1. 21 Tầng Libraries trong kiến trúc Android 20*](#_Toc122906182)

[*Hình 1. 22 Tầng Android Runtime trong kiến trúc Android 21*](#_Toc122906183)

[*Hình 1. 23 Tầng Application Framework trong kiến trúc Android 22*](#_Toc122906184)

[*Hình 1. 24 Tầng Applications trong kiến trúc Android 23*](#_Toc122906185)

[*Hình 1. 25 Cơ chế máy ảo Dalvik 25*](#_Toc122906186)

[*Hình 1. 26 Vòng đời của một Activity 26*](#_Toc122906187)

[*Hình 1. 27 Vòng đời của một Services 27*](#_Toc122906188)

[*Hình 1. 28 Vòng đời của một Fragment 28*](#_Toc122906189)

[*Hình 2. 1 Biểu đồ tổng quan hệ thống 36*](#_Toc122906662)

[*Hình 2. 2 Biểu đồ ca sử dụng hệ thống 36*](#_Toc122906663)

[*Hình 2. 3 Biểu đồ ca sử dụng quản lý nhiệm vụ 37*](#_Toc122906664)

[*Hình 2. 4 Biểu đồ ca sử dụng quản lý nhắc nhở 37*](#_Toc122906665)

[*Hình 2. 5 Biểu đồ hoạt động kịch bản hiển thị nhắc nhở 40*](#_Toc122906666)

[*Hình 2. 6 Biểu đồ trình tự kịch bản hiển thị nhắc nhở 41*](#_Toc122906667)

[*Hình 2. 7 Biểu đồ hoạt động kịch bản thêm nhắc nhở 41*](#_Toc122906668)

[*Hình 2. 8 Biểu đồ trình tự kịch bản thêm nhắc nhở 42*](#_Toc122906669)

[*Hình 2. 9 Biểu đồ hoạt động kịch bản sửa nhắc nhở 43*](#_Toc122906670)

[*Hình 2. 10 Biểu đồ trình tự kịch bản sửa nhắc nhở 44*](#_Toc122906671)

[*Hình 2. 11 Biểu đồ hoạt động kịch bản xóa nhắc nhở 44*](#_Toc122906672)

[*Hình 2. 12 Biểu đồ trình tự kịch bản xóa nhắc nhở 45*](#_Toc122906673)

[*Hình 2. 13 Biểu đồ hoạt động kịch bản hiển thị nhiệm vụ 47*](#_Toc122906674)

[*Hình 2. 14 Biểu đồ trình tự kịch bản hiển thị nhiệm vụ 47*](#_Toc122906675)

[*Hình 3. 1 Thực thi đề tài trên Android studio 49*](#_Toc122980707)

[*Hình 3. 2 Giao diện chính của chương trình 51*](#_Toc122980708)

[*Hình 3. 3 Nội dung của nhiệm vụ 52*](#_Toc122980709)

[*Hình 3. 4 Thông báo của nhiệm vụ 52*](#_Toc122980710)

[*Hình 3. 5 Thanh công cụ tìm kiếm 53*](#_Toc122980711)

[*Hình 3. 6 Lọc nhiệm vụ 54*](#_Toc122980712)

[*Hình 3. 7 Thêm nhiệm vụ 55*](#_Toc122980713)

[*Hình 3. 8 Speech To Text 56*](#_Toc122980714)

[*Hình 3. 9 Chữ viết tay 57*](#_Toc122980715)

[*Hình 3. 10 Sửa nhiệm vụ 58*](#_Toc122980716)

# DANH MỤC BẢNG BIỂU

[*Bảng 2. 1 Mô tả luồng sự kiện trong quản lý nhắc nhở 38*](#_Toc122981387)

[*Bảng 2. 2 Mô tả luồng sự kiện trong quản lý nhiệm vụ 46*](#_Toc122981388)

[*Bảng 2. 3 Bảng cấu trúc cơ sở dữ liệu 48*](#_Toc122981389)

[*Bảng 3. 1 Một số modul chính trong chương trình 50*](#_Toc122981630)

# LỜI MỞ ĐẦU

1. **Mục tiêu đề tài**

* Ứng dụng ToDo&Reminder giúp chúng ta lên kế hoạch việc cần làm, nhiệm vụ cần hoàn thành một cách dễ dàng.
* Giao diện đơn giản, dễ sự dụng.
* Đặt thông báo nhắc nhở thực hiện công việc.
* Hỗ trợ bạn đánh dấu hoặc xoá bỏ công việc đã hoàn thành.

1. **Giới hạn đề tài**

Đề tài có một số giới hạn sau:

* Tạo một kế hoạch hoặc một sự kiện .
* Người dùng có thể theo dõi các sự kiện và kế hoạch của mình để dễ dang thực hiện.
* Ứng dụng sẽ tạo một thông báo nhắc nhở người dùng về những sự kện hoặc kế hoạch chưa hoàn thành.

1. **Nội dung nghiên cứu**

Đề tài được thực hiện gồm có 3 chương chính như sau:

* **Chương 1:** Giới thiệu về Android.
* **Chương 2:** Xác định yêu cầu và phân tích thiết kế hệ thống.
* **Chương 3:** Thiết kế chương trình.

# NỘI DUNG NGHIÊN CỨU

# CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDRIOD

## 1.1 Tìm hiểu chung về hệ điều hành Android

### ****1.1.1 Khái niệm Android****

**Android** là hệ điều hành trên thiết bị di động (mobile, tablet và hiện nay là cả trên các thiết bị phát HD, PC) phát triển bởi Google và dựa trên nền tảng Linux.

Android là nền tảng cho thiết bị di động bao gồm một hệ điều hành, midware và một số ứng dụng chủ đạo. Bộ công cụ Android SDK cung cấp các công cụ và bộ thư viện các hàm API cần thiết để phát triển ứng dụng cho nền tảng Android sử dụng ngôn ngữ lập trình Java.



Hình 1. Minh họa Android.

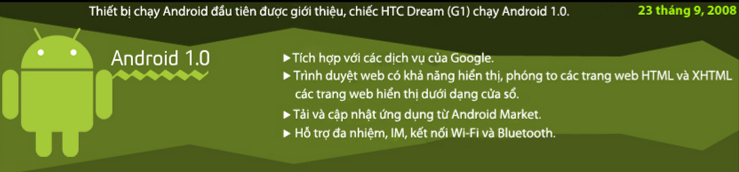
Ưu điểm chính của việc áp dụng Android là cung cấp một cách tiếp cận thống nhất để phát triển ứng dụng. Các nhà phát triển chỉ cần phát triển cho Android và các ứng dụng của họ có thể chạy trên nhiều thiết bị khác nhau, miễn là các thiết bị được hỗ trợ bằng cách sử dụng Android.

### 1.1.2 Lịch sử Android

Trước đây, Android được phát triển bởi công ty liên hợp Android (sau đó được Google mua lại vào năm 2005). Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở. Google công bố hầu hết các mã nguồn của Android theo bản cấp phép Apache.

* **Phiên bản Andoid 1.0**

Dường như ai dùng hệ điều hành Android thì cũng đều biết đều được biết rằng nó thường gắn với một cái tên nhất định. Tuy nhiên ở bản 1.0 thì Android xuất hiện thì nó chưa có một cái tên ý nghĩa mà nó chỉ mới được đặt tên bằng số. Sau đó Google phát hành bản cập nhật Android 1.1 dành riêng cho T-Mobile G1.



Hình 1. Phiên bản Android 1.0

* Hệ điều hành Android thời gian đầu ra mắt mang những đặc điểm, tính năng thú vị phải kể đến như:
  + Thanh thông báo kéo từ trên xuống cho phép người dùng xem nhanh các thông tin ngày giờ, tin nhắn, cuộc gọi…
  + Màn hình chính và Widget: màn hình chính gồm các biểu tượng chương trình người dùng hay truy cập và các Widget là các ứng dụng nhỏ trên màn hình chính, hoạt động và cung cấp thông tin liên tục.
  + Tích hợp chặt chẽ với Gmail



*Hình 1. 3* Giao diện được cải tiến dành riêng cho điện thoại di động

* **Phiên bản Android 1.1**

**

*Hình 1. 4* Minh họa giao diện Android 1.1

Phiên bản cập nhật đầu tiên của hệ điều hành Android giúp tinh chỉnh và sửa các lỗi gặp phải trên phiên bản 1.0. Tính năng cập nhật phần mềm từ động qua OTA được bổ sung và được xem như là một cải tiến đáng giá vì các hệ điều hành di động trước đó đều phải nhờ tới một chiếc máy tính để thực hiện việc này.

* **Phiên bản Android 1.5**

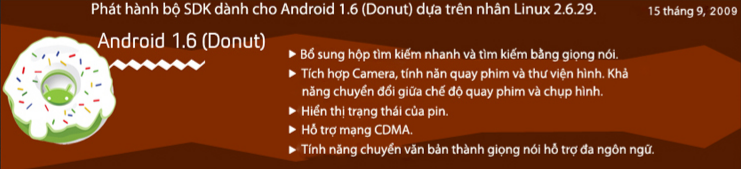
Từ phiên bản 1.5 Google chính thức đặt tên nó là một món tráng miệng được phát hành theo thứ tự bảng chữ cái thì người ta mới thực sự chú ý hơn với các tính năng nổi bật. Andoird 1.5 với cái tên là Cupcake (bánh nướng nhỏ), phiên bản dựa trên nhân Linux 2.6.27 phát hành vào tháng 4 năm 2009.



Hình 1. Phiên bản Android 1.5

* Về mặt giao diện, Android 1.5 không có nhiều điểm thay đổi so với người tiền nhiệm của mình. Google chỉ điểm thêm vài điểm để làm giao diện trông bóng bẩy, mượt mà hơn một tí, chẳng hạn như widget tìm kiếm có độ trong suốt nhẹ, biểu tượng app drawer có một số hoa văn nhỏ mới, v.v. Nói chung, giao diện không phải là một điểm nhấn của Android 1.5 mà người ta quan tâm nhiều hơn đến các tính năng mới mà nó mang lại, chẳng hạn như:
* Bàn phím ảo lần đầu tiên xuất hiện.
* Cải tiến Widget: kho ứng dụng Widget phong phú và đa dạng hơn.
* Hỗ trợ khả năng quay phim cho camera.
* Cải tiến Clipboard để nâng cao khả năng Coppy, Cut, Paste.
* **Phiên bản Android 1.6**

Tiếp đó, tháng 9 cùng năm Android 1.6 ra đời với cái tên là Donut. Donut là tên của một loại bánh rán hình vòng tròn



*Hình 1. 6 Phiên bản Android 1.6*

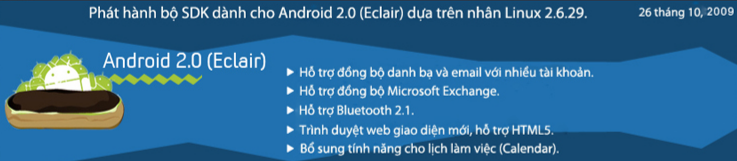
Phiên bản bánh Donut này, mặc dù chỉ thêm có 0.1 vào mã số của Android 1.5 nhưng nó cũng mang lại nhiều cải tiến đáng giá. Một vài điểm trong giao diện được cải thiện, vài tính năng nhỏ được thêm vào, cuối cùng là hỗ trợ cho mạng CDMA.



*Hình 1. 7 Minh họa giao diện Android 1.6*

* Bổ sung tính năng có thể chạy trên nhiều độ phân giải và tỉ lệ màn hình khác nhau, cho phép những thiết bị có nhiều độ phân giải hơn là 320 x 480. Hiện nay, chúng ta có những chiếc smartphone Android chạy ở độ phân giải QVGA, HVGA, WVGA, FWVGA, qHD, và 720p. Vài chiếc máy tính bảng còn đạt mức 1920 x 1080 nữa.
* Tính năng Quick Search Box, được biết nhiều hơn trong thế giới điện thoại với cái tên Universal Search, cũng là một điểm mà Android nhận được nhiều lời khen. Bạn có thể tìm kiếm danh bạ, ứng dụng, nhạc, tin nhắn,…, tất cả đều chỉ thao tác trong một hộp tìm kiếm mà thôi.
* Android 1.6 còn có gì mới? Đó là một Android Market với thiết kế mang tông màu xanh trắng đặc trưng của Android, có thể hiển thị các ứng dụng free và trả phí hàng đầu. Những ứng dụng bên thứ ba cũng bắt đầu xuất hiện nhiều hơn. Giao diện camera mới, tích hợp với trình xem ảnh tốt hơn, giảm thời gian chờ giữa hai lần chụp ảnh. Nhưng cũng thường thì các nhà sản xuất cũng thay thế bằng ứng dụng của riêng mình.
* **Phiên bản Android 2.0**

Chỉ một tháng tung ra phiên bản Donut thì Google đã tung ra phiên bản mới để sửa lỗi và cập nhật với mã là 2.0, 2.1 và được gọi với cái tên chung là Eclair, một loại bánh ngọt xuất xứ từ Pháp có hình dạng thon dài lại có nhân kem và phủ các loại kem hoặc đường đông bên trên.



*Hình 1. 8 Phiên bản Android 2.0*

*Gồm các tính năng mới sau:*

* Hỗ trợ nhiều tài khoản người dùng: lần đầu tiên, nhiều tài khoản Google có thể được đăng nhập trên cùng một thiết bị chạy Android. Tài khoản Microsoft Exchange cũng được hỗ trợ trong Eclair. Người dùng có thể duyệt qua danh bạ, email của từng tài khoản. Google giờ đây cho phép những nhà phát triển bên thứ ba tích hợp dịch vụ của họ vào trong mục Account này, đồng thời hỗ trợ tự động đồng bộ hóa.
* Quick Contact: khi chạm vào một số liên lạc nào đó, có một menu nhỏ sẽ xuất hiện để bạn tương tác theo nhiều cách: gửi email, gọi điện, nhắn tin và hơn thế nữa. Miễn là nơi nào có biểu tượng contact xuất hiện là nơi đó có thể dùng Quick Contact. Sau này có thêm Twitter, Facebook và nhiều dịch vụ đồng bộ khác cũng tích hợp tính năng riêng của mình vào Quick Contact.
* Cải tiến bàn phím ảo
* Trình duyệt mới: hỗ trợ HTML5, bao gồm luôn khả năng phát video ở chế độ toàn màn hình. Hộp địa chỉ kết hợp với thanh tìm kiếm (giống với trình duyệt Chrome) cũng lần đầu xuất hiện trên Android. Để bù lại cho tính năng cảm ứng đa điểm, trình duyệt mới hỗ trợ chạm hai lần để phóng to nội dung trên màn hình, kèm theo đó là hai nút Zoom in, Zoom out.
* Giao diện mới: các biểu tượng giờ đây đã đẹp hơn, sang trọng hơn, gọn gàng hơn nhiều so với trước. Widget cũng được thiết kế mới để tương thích với độ phân giải cao trên Droid. Giao diện menu cũng đẹp hơn nữa.



*Hình 1. 9 Minh hoạn giao diện Android 2.0*

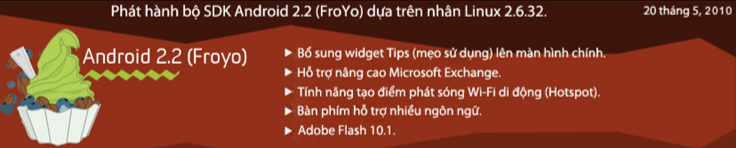
* **Phiên bản Android 2.1**

Ra mắt chủ yếu để sửa lỗi và thêm hàm API để lập trình viên can thiệp sâu hơn vào hệ thống nhưng nó đã hỗ trợ thêm vài tính năng lý thú như Live Wallpaper, chuyển giọng nói thành văn bản và một màn hình khóa mới. Android 2.1 được đánh dấu bởi sự ra đời của chiếc Nexus One do HTC sản xuất, chiếc điện thoại Nexus đầu tiên chạy Android chuẩn của Google.

* **Phiên bản Android 2.2**

Món tráng miệng tiếp theo được Google chọn ra là vào tháng 5 năm 2010, món sữa chua đá Froyo viết tắt của Frozen Yogurt (sữa chua đá tuyết), hay còn có tên với số hiệu là 2.2.

Nexus One là chiếc điện thoại đầu tiên được nâng cấp lên Android 2.2, sớm hơn nhiều so với tất cả các hãng khác. Giao diện màn hình chính đã được thay đổi, từ 3 màn hình chính tăng lên thành 5 màn hình. Dãy nút kích hoạt nhanh chế độ gọi điện, web và App Drawer cũng đã xuất hiện. Những chấm nhỏ ở góc trái, phải bên dưới của màn hình giúp người dùng biết mình đang xem đến màn hình nào. Google đã có nhiều cố gắng để giao diện Android được vui và đẹp hơn, dễ dùng hơn, bắt kịp phần nào với giao diện của bên thứ ba như HTC Sense chẳng hạn.



*Hình 1. 10 Phiên bản Android 2.2*

Froyo còn có một trình xem ảnh mới với khả năng hiển thị hình ảnh 3D: nghiêng máy thì dãy ảnh sẽ nghiên theo, đồng thời nó mang trong mình nhiều hiệu ứng chuyển động đẹp mắt khác. Tính năng trạm phát Wifi (Wifi Hotspot) cũng xuất hiện, cách copy, paste mới tốt hơn. Nhiều tính năng bảo mật mới cũng xuất hiện. Việc hỗ trợ duyệt web với Flash là điểm đáng quan tâm nhất. Hầu như người ta muốn lên FroYo là để tận hưởng được cảm giác duyệt web với Flash đầy đủ trên thiết bị di động. Android 2.2 bổ sung thêm tính năng di chuyển một phần ứng dụng từ bộ nhớ máy sang thẻ nhớ, giúp tiết kiệm dung lượng cho các điện thoại thời bấy giờ.

* **Phiên bản Android 2.3**

Một bản cập nhật tiếp mà hiện nay đang là bản Android phổ biến sau Jelly Bean đó là Android 2.3 (Gingerbread) cho suốt năm 2010 cho đến tận thời điểm này. Chính vì sự ổn định của nó mà tại thời điểm này người dùng vẫn ngại không nâng cấp lên phiên bản mới. Bánh gừng Gingerbread là tên gọi của phiên bản 2.3 xuất hiện vào cuối năm 2010, gần 6 tháng sau Froyo.

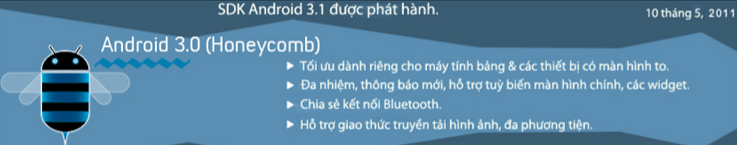


*Hình 1. 11 Phiên bản Andoid 2.3*

*Các tính năng mới:*

* Giao diện: được khoác giao diện màu đen tuyền kèm chút xanh đen mang đến vẻ sang trọng và chuyên nghiệp. Màu sắc này giúp các thiết bị có màn hình AMOLED tiết kiệm pin hơn.
* Hai thanh chặn khi chọn văn bản: Hồi Android 2.2, việc chọn văn bản khi đang soạn thảo hoặc khi đang duyệt web rất cực khổ. Nhờ có thanh chặn này mà người dùng có thể chọn từng kí tự một, tiện lợi hơn rất nhiều.
* Bàn phím được cải tiến: Google một lần nữa đầu tư cho bàn phím ảo, lần này có sự thay đổi lớn về tông màu của bàn phím. Nhìn các phím đẹp hơn, đen hơn và cả kí tự cũng được thay đổi so với hồi Android Cupcake.
* Công cụ quản lí pin và ứng dụng: vì Android cho phép các ứng dụng chạy nền thực thụ chứ không bị ngắt lại, vấn đề pin bị người dùng phàn nàn rất nhiều. Một công cụ mới đã ra mắt với biểu đồ nhằm theo dõi mức độ tiêu thụ pin cùng việc quản lí ứng dụng đang chạy sẽ giúp đỡ người dùng phần nào.
* Hỗ trợ máy ảnh trước: Lần đầu tiên một phiên bản của hệ điều hành Android hỗ trợ việc sử dụng máy ảnh phụ phía trước của thiết bị. Trên chiếc Nexus S có một máy ảnh trước, nhưng lúc thiết bị này mới ra mắt thì chúng ta không làm được gì nhiều ngoài việc chụp ảnh "tự sướng".
* Hỗ trợ kết nối NFC
* **Phiên bản Android 3.0**

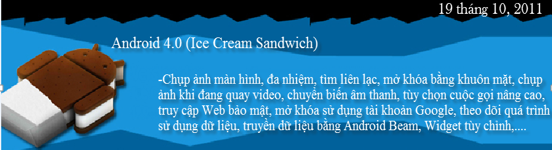
Sau khi tập trung vào mảng điện thoại di động thì Google cũng đã nhận ra tiềm năng của những chiếc máy tính bảng khi Ipad của Apple đang có sức thu hút lớn trên toàn thế giới. Và muốn chặn đứng lại trào lưu này, hệ điều hành Android 3.0 là câu trả lời đầu tiên của Google dành cho Apple. Phiên bản Android 3.0 lấy tên gọi từ bánh mật tổ ong, Honeycomb, được thiết kế lấy máy tính bảng làm trung tâm ra đời vào ngày 10 tháng 5 năm 2011.



*Hình 1. 12 Phiên bản Andoid 3.0*

* **Phiên bản Android 4.0**

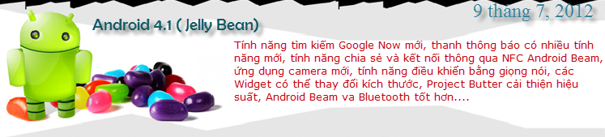
Ra mắt ngày 19 tháng 10 năm 2011, phiên bản 4.0 có một thiết kế hoàn toàn khác biệt trọng đó là nó đã gộp giao diện điện thoại di động với máy tính bảng làm một, cho thấy một điều máy tính bảng và điện thoại di động đã được tích hợp trong cùng một hệ điều hành Android. Ice Cream Sandwich là một chiếc sandwich kem lạnh, tên món tráng miệng của Android 4.0.



*Hình 1. 13 Phiên bản Andoid 4.0*

* **Phiên bản Android 4.1**

Tháng 7 năm 2012, một nhân vật mới trình làng Android có cái tên cũng như biểu tượng khá ngộ nghĩnh. Đây cũng là phiên bản ra sau Ice Cream Sandwich nhưng có lượng người dùng hiện đang là đông đảo nhất. Android 4.1 hay còn gọi là Jelly Bean, một loại kẹo dẻo hình hạt đậu với nhiều tính năng hấp dẫn.



*Hình 1. 14 Phiên bản Andoid 4.1*

* Về giao diện, không có nhiều thay đổi so với Android 4.0, vẫn là màn hình chính với thanh dock bên dưới quen thuộc. Giao diện kiểu này rất đơn giản nhưng lại đẹp và gọn gàng, giúp chúng ta dễ dàng theo dõi những thông tin xuất hiện trên màn hình, đặc biệt là khi điều khiển các ứng dụng. Ở màn hình Lockscreen, giờ đây vòng trượt mở khóa có đến ba biểu tượng để kích hoạt ba tính năng khác nhau chứ không chỉ có hai như trên Android 4.0. Trượt sang phải để mở khóa máy, sang trái để chạy ứng dụng camera, hai thứ này đã có từ trước. Nhưng nếu bạn kéo vòng trượt lên bên trên thì sẽ thấy một chữ "Google". Đây chính là cách thức để kích hoạt Google Now.
* Sự xuất hiện của Google Now cho thấy rằng Google đã bắt đầu bước chân vào việc cạnh tranh với Siri. Với việc nâng cao khả năng tìm kiếm, thay đổi giao diện, tự động theo dõi thói quen sử dụng để đưa ra những thông tin chính xác và thích hợp với người dùng hơn. Trên Android 4.1 bạn có thể tìm kiếm bằng giọng nói và các kết quả trả không chỉ đơn giản là những dòng tìm kiếm nữa mà nó được thiết kế theo dạng thẻ đồ họa, thông minh hơn, trực quan hơn. Không chỉ tìm kiếm theo yêu cầu, Google Now còn tự chủ động thu thập thông tin liên tục dựa vào lịch sử tìm kiếm của bạn (thói quen người dùng), dựa vào danh sách lịch hẹn để biết bạn sẽ đi đâu, cần thông tin gì.
* Nhưng có lẽ quan trọng hơn hết của Jelly Bean không phải là về giao diện hay ứng dụng mới mà về Project Butter giúp mang lại độ mượt chưa từng có cho Android.
* **Phiên bản Android 4.2 Jelly Bean**

Ngày 30/10/2012, Google chính thức tuyên bố cập nhật hệ điều hành Android của hãng lên phiên bản 4.2 và vẫn giữ nguyên tên gọi "Jelly Bean" (có nghĩa là kẹo dẻo đậu). Được hãng gọi là "một hương vị mới của Jelly Bean", Android 4.2 mang trong mình nhiều tính năng mới như hỗ trợ Miracast, bàn phím có thể nhập liệu bằng cách vẽ các đường nét từ kí tự này đến kí tự khác, chế độ chụp ảnh toàn cảnh Photo Sphere, ứng dụng Gmail mới và còn rất nhiều thứ khác nữa.

Trong quá trình giới thiệu Android 4.2, Google rất tập trung nói về Miracast, một chuẩn chia sẻ nội dung số thông qua kết nối Wi-Fi mà nhờ nó, những thiết bị mới có thể truyền hình ảnh và âm thanh đến các TV có hỗ trợ chuẩn này. Google cũng đã áp dụng một số kĩ thuật từ camera Street View của hãng cho tính năng chụp ảnh toàn cảnh mới có tên Photo Sphere. Khi chạy app Camera và kích hoạt Photo Sphere, máy sẽ yêu cầu chúng ta di chuyển và ngắm vào một số điểm trên màn hình, dần dần chụp qua hết tất cả các điểm này sẽ cho ra kết quả là một bức ảnh lớn có hình mặt cầu (hơi giống PhotoSynths của Microsoft).Một cải tiến lớn và quan trọng của Android 4.2 đó là việc hỗ trợ nhiều tài khoản người dùng trên máy tính bảng để có thể dễ dàng chia sẻ trong gia đình hoặc nơi làm việc. Mỗi tài khoản sẽ có dữ liệu app của riêng họ. Ví dụ, người dùng A cài game Angry Birds, người này chơi và đã ghi được một số điểm cũng như lên được màn cao hơn. Khi người B sử dụng máy bằng tài khoản khác, game này vẫn hiện diện trên máy nhưng dưới dạng một bản cài mới, không có sẵn data của người A.

* **Phiên bản Android 4.3 Jelly Bean**

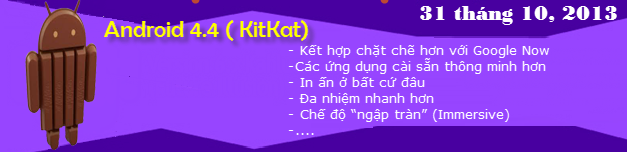
Lại thêm một thế hệ Jelly Bean nữa và lần này là Android 4.3. Ngày 24/7/2013, Google đã chính thức ra mắt hệ điều hành này song song với chiếc Nexus 7 (2013). Đây là phiên bản Android mới nhất đang có mặt trên thị trường và đi kèm những tính năng mới như hỗ trợ kết nối Bluetooth Smart, bộ API OpenGL ES 3.0, bổ sung tính năng sử dụng Wi-Fi để định vị ngay cả khi người dùng tắt kết nối này đi cùng nhiều thay đổi lớn nhỏ khác.

****

*Hình 1. 15 Minh họa phiên bản Android 4.3 Jelly Bean*

* Bên cạnh tính năng Multi User của Android 4.2 như đã nói ở trên, Google bổ sung thêm một tính năng mới cho Android 4.3 đó là Restricted Profile. Mỗi thành viên sẽ có một "hồ sơ" riêng của mình và chỉ có thể làm được những gì được chỉ định trong "hồ sơ". Ví dụ, cha mẹ có thể khóa tính năng in-app purchase của một ứng dụng nào đó để ngăn con trẻ vô tình mua hàng trăm USD. Ngoài ra, khi áp dụng vào môi trường doanh nghiệp thì nó sẽ cho phép người quản trị phân quyền cho nhân viên dễ dàng hơn.
* Chưa hết, Google còn làm mới trình gọi điện thoại mặc định để hỗ trợ việc tìm kiếm trong danh bạ khi người dùng bấm số, giống như kiểu bàn phím T9 trước đây. Google cũng có thêm vào Notification Access để cho phép các ứng dụng bên thứ ba truy cập và tương tác với những thông báo nằm trong notification hệ thống ngay khi chúng vừa xuất hiện.
* **Phiên bản Android 4.4**

Tháng 10 năm 2013, phát hành phiên bản 4.4 với tên gọi là KitKat. KitKat là phiên bản đầu tiên mà Google đã hợp tác với một thương hiệu bên ngoài để làm linh vật Android, thương hiệu đó là Nestle. Chiếc bánh sô cô la này đem đến những hương vị ngọt ngào bởi những tính năng tuyệt vời. Cách đặt tên này cũng hoàn toàn khớp với trình tự chữ cái mà các phiên bản Android sử dụng làm tên mã. Google tiết lộ thêm rằng "mục tiêu của chúng tôi với Android KitKat đó là mang trải nghiệm Android đáng kinh ngạc đến cho mọi người". Android Kitkat xuất hiện đầu tiên trên chiếc smartphone Nexus 5. Những cải tiến trên Android Kitkat bao gồm Chế độ toàn màn hình – Immersive Mode, Hiệu ứng chuyển cảnh màn hình -Transition Manager, Storage Access Framework, Chromium WebView, NFC,Cổng hồng ngoại – Infrared Blasters …



*Hình 1. 16 Phiên bản Andoid 4.4*

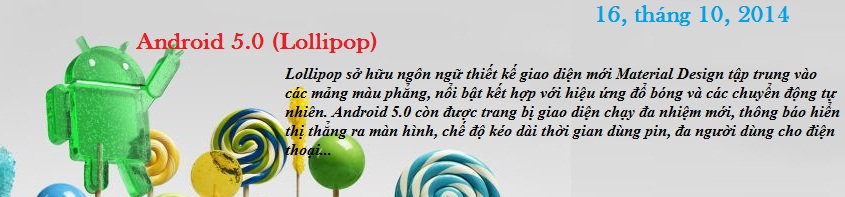
Giao diện: Tông màu chủ đạo của font chữ là xanh và đen được sử dụng trên Android Jelly Bean được Google thay đổi bằng tông màu đen và ghi trên Android 4.4 Kitkat vừa ra mắt, thể hiện rõ ở các dòng chữ trên thanh thông báo Notification. Bên cạnh đó, Android Kitkat cũng có bộ biểu tượng, folder mới, giao diện nhìn chung được làm phẳng bớt, loại bỏ các chi tiết đổ bóng và thêm hiệu ứng trong suốt vào nhiều phần, như ba phim ảo bên dưới hay thanh thông báo ở phía trên.



*Hình 1. 17 Minh họa giao diện Android 4.4*

* Phần mềm điện thoại mặc định của Kitkat được làm mới thông minh hơn, cho phép thao tác nhanh. Bên cạnh đó, Google cũng kết hợp dịch vụ tìm kiếm Google Search vào ứng dụng này, cho phép dò và tự động liên kết các danh bạ có sẵn trên Internet váo số điện thoại mới.
* Từ phiên bản 4.4, ứng dụng nhắn tin truyền thông Text Messages được gộp chung với ứng dụng nhắn tin và liên lạc online Hangouts. Đây là một sự thay đổi hợp lý giúp người dùng Android có được sự thuận tiện khi liên lạc trên điện thoại, giống như iMessage trên iOS.
* Bàn phím mặc định của Android được bổ sung thêm các biểu tượng Emoji.
* Google cũng cho phép người dùng thay đổi giao diện (laucher) trên Android 4.4 dễ dàng hơn các phiên bản trước khi bổ sung thêm mục cài đặt Home ở trong Setting, cho phép chọn lựa các Laucher khác làm giao diện chính của thiết bị thay thế giao diện mà Google cung cấp.
* Bên cạnh việc chụp ảnh màn hình, tính năng Screen Recording còn cho phép ghi lại các hoạt động đang diễn ra trên màn hình của Android 4.4 thành các đoạn video MP4.
* Chế độ định vị, phát nhạc trên Android 4.4 mới đều sẽ tiết kiệm năng lượng hơn. Ngay trên màn hình khóa, các bản nhạc hay video đang phát cũng có thể điều chỉnh được thời gian hay tạm dừng, không bắt buộc người dùng phải mở khóa và truy cập vào ứng dụng như cũ. Ngoài ra, phiên bản mới của Google cũng đưa ra các chế độ kết nối Bluetooth tiết kiệm năng lượng, cung cấp các hàm lập trình, điều khiển bằng hồng ngoại IR. Tính năng in dữ liệu qua kết nối không dây cũng được tích hợp sẵn vào Kitkat.
* Dù là phiên bản mới, Android 4.4 Kitkat lại không đòi hỏi cấu hình phần cứng mạnh hơn, thậm chí hỗ trợ tốt cả những thiết bị cũ với phần cứng không cao, như có RAM chỉ đạt dung lượng 512 MB. Hệ điều hành mới được Google tối ưu khả năng hoạt động, cho hiệu năng cao hơn tới 1,6 lần phiên bản trước.
* **Phiên bản Android 5.0**

Giữa tháng 10 năm nay Google chính thức ra mắt phiên bản Android 5.0 Lollipop (kẹo que, kẹo mút).



*Hình 1. 18 Phiên bản Andoid 5.0*

* Sau hơn một năm Android 4.0 KitKat ra mắt, Google đã chính thức trình làng phiên bản hậu duệ mới nhất với tên gọi Android Lollipop, được đánh giá là hệ điều hành có những thay đổi rõ rệt và đáng kể nhất trong lịch sử Android từ trước đến nay.
* Thiết kế "Material" mới: Ngôn ngữ đồ họa mới của Android được lấy cảm hứng từ thiên nhiên, vật lý và phong cách đậm, đổ bóng giống như in ấn. Nói cách khác, đó là một thiết kế dựa trên những đặc tính của giấy.
* Android trên mọi màn hình: Trên điện thoại, tablet, laptop, thậm chí là TV, ô tô, đồng hồ và đồ điện gia dụng, hệ điều hành Android mới học theo cách tiếp cận của Microsoft khi cố gắng hợp nhất mọi nền tảng về một mái nhà duy nhất mang tên Windows.
* Thiết kế báo nhắc hoàn toàn mới: Kẹo ngọt mang đến một phong cách báo nhắc thực sự khác biệt. Giờ đây, hệ thống báo nhắc cải tiến chỉ hiện ra những thông báo quan trọng nhất và tự động ẩn đi những thông báo không được ưu tiên. Khi bạn trượt tay xuống dưới, giống như lật một dãy bài, những báo nhắc kia mới lần lượt hiện ra.
* Một tính năng mới thú vị nữa trong hạng mục báo nhắc là cách chúng nhảy ra (pop up) mà không can thiệp vào những gì bạn đang xem hoặc đang thao tác trên màn hình. Bạn còn có thể phớt lờ chúng, hoặc xử lý chúng ngay ở cửa sổ pop up một cách dễ dàng. Hãy tưởng tượng thế nào: Bạn đang chơi game thì nhận được lời mời tham gia một cuộc chat video nhóm. Giờ đây, bạn không cần phải mở trung tâm báo nhắc nữa mà có thể chấp nhận hoặc từ chối lời mời này ngay tức khắc, chưa kể là cuộc chat còn không ảnh hưởng đến trò chơi bạn đang chơi nữa.
* Hỗ trợ chip 64 bit.
* Cải thiện thời lượng pin.
* Mở khóa dựa trên bối cảnh: Mở khóa riêng tư là một ý tưởng thông minh, cho phép người dùng mở khóa thiết bị bằng một [smartphone](https://www.thegioididong.com/dtdd) hoặc tablet đang kết nối Bluetooth với thiết bị gốc, bằng vị trí hiện tại của người dùng hay thậm chí là giọng nói của họ.
* Tách bạch công việc và giải trí: Google khẳng định rằng Lollipop sẽ có những tính năng giúp người dùng dễ dàng phân biệt dữ liệu cá nhân với dữ liệu công việc hơn.
* Tìm kiếm bối cảnh cải tiến: Google Search giờ đây sẽ nhận thức được tốt hơn về trạng thái hiện tại của người dùng, từ đó đưa ra những kết quả tìm kiếm phù hợp hơn với bối cảnh.

***Tỷ lệ sử dụng các phiên bản khác nhau thu thập trong 14 ngày đến ngày 4 tháng 3 năm 2013***



*Hình 1. 19 Tỷ lệ sử dụng các phiên bản*

### 1.1.3 Một số đặc trưng của hệ điều hành Android

* Application framework: cho phép sử dụng lại và thay thế các thành phần trong lập trình ứng dụng.
* Dalvik virtual machine: tối ưu hóa cho thiết bị di động.
* Intergrated browser: trình duyệt tích hợp, dựa trên cơ chế WebKit mã nguồn mở.
* SQLite: cơ sở dữ liệu trong môi trường di động.
* Media support: hỗ trợ các định dạng audio, video và hình ảnh thông dụng.
* GSM Telephony: mạng điện thoại di động (phụ thuộc vào phần cứng).
* Bluetooth, EDGE, 3G, và WiFi: các chuẩn kết nối dữ liệu (phụ thuộc vào phần cứng).
* Camera, GPS, la bàn, và gia tốc kế (phụ thuộc vào phần cứng).
* Môi trường phát triển phong phú: bao gồm thiết bị mô phỏng, công cụ cho việc dò tìm lỗi, bộ nhớ, định hình hiệu năng và một plugin cho Eclipse IDE.

### 1.1.4 Một số tính năng hỗ trợ sẵn trong hệ điều hành Android

* Storage: Sử dụng SQLite, một cơ sở dữ liệu quan hệ nhẹ cho việc lưu trữ dữ liệu.
* Connectivity: Hỗ trợ GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (bao gồm AD2P và AVRCP), Wifi, LTE và WiMAX.
* Web browser: Dựa trên mã nguồn mở Webkit, cùng với công nghệ JavaScript V8 của Chrome.
* Media support: Bao gồm hỗ trợ các phương tiện truyền thông sau: H.263, H.264 ( Trong 3GP hoặc MP4 container), MPEG-4 SP, AMP, AMR-WB (3GB container), AAC, HE-AAC (MP4 hoặc 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF và BMP.
* Hardware support: Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor và GPS.
* Multi-touch: Hỗ trợ màn hình cảm ứng đa điểm.
* Multi-tasking: Hỗ trợ ứng dụng đa tác vụ.
* Flash support: Android 2.3 hỗ trợ Flash 10.2
* Tethering: Hỗ trợ kết nối Internet không dây/có dây.

### 1.1.5 Cấu trúc của hệ điều hành Android

Các tầng cơ bản trong hệ điều hành Android: Linux Kernel, Android Runtime và Native Libraries, Application Framework, Application. Mỗi tầng làm việc đều nhờ sự giúp đỡ của tầng bên dưới.

* ***Tầng Linux Kernel***

Linux Kernel là tầng trong cùng đóng vai trò làm cơ sở, nền tảng phát triển cho hệ điều hành Android. Tất cả mọi hoạt động của điện thoại muốn thực thi đều phải thông qua tầng này, tầng này giúp ta kết nối đến tất cả các phần cứng của điện thoại.



Hình 1. 20 Tầng Linux Kernel trong kiến trúc Android

*Các thành phần chính*:

* + - Display Driver: điều khiển việc hiển thị lên màn hình và những điều khiển của người dùng lên trên màn hình (như di chuyển, cảm ứng...).
    - Camera Driver: Điều khiển hoạt động của Camera, nhận luồng dữ liệu từ Camera trả về.
    - Bluetooth Driver: Điều khiển thiết bị thu và phát sóng Bluetooth.
    - USB Driver: Quản lý hoạt động của các cổng giao tiếp USB.
    - Keypad Driver: Điều khiển bàn phím.
    - Wifi Driver: Chịu trách nhiệm về việc thu phát sóng wifi.
    - Audio Driver: Điều khiển các bộ thu phát âm thanh, giải mã các tín hiệu dạng audio thành tín hiệu số và ngược lại.
* ***Tầng Libraries và Android Runtime***
* Tầng Libraries:

Tầng libraries chứa tất cả các thư viện lõi chủ yếu được xây dựng dựa trên ngôn nhữ C/C++ của hệ điều hành Android để cung cấp cho nhà phát triển thiết thế chương trình.



*Hình 1. 21 Tầng Libraries trong kiến trúc Android*

*Các thư viện chính:*

* Surface Manager: quản lý hiển thị, giúp cho nhà phát triển có thể soạn thảo graphic 2D và 3D để vẽ lên màn hình hiển thị.
* Media Framework: được cung cấp bởi gói video, là một phần của liên minh các thiết bị cầm tay. Media Framework cung cấp tất cả các thư viện để lập trình đa phương tiện.
* SQLite: cung cấp tất cả các đối tượng được sử dụng để quản lý cơ sở dữ liệu trong ứng dụng và chia sẻ dữ liệu với các ứng dụng khác.
* OPENGL ES: là thư viện đồ họa 3D.
* Free Type: giúp tùy chỉnh Fonts chữ trong ứng dụng.
* WebKit: là công cụ hỗ trợ lướt web.
* Android Runtime



*Hình 1. 22 Tầng Android Runtime trong kiến trúc Android*

* Libraries Core: là thư viện lõi được viết bằng ngôn ngữ java, chứa tất cả các class, io, tool để sử dụng trong ứng dụng.
* Daivik Virual Machine: là máy ảo được xây dựng trong Device để có thể chạy được những ứng dụng với file là apk.
* ***Tầng Application Framework***

Tầng này cung cấp một nền tảng phát triển ứng dụng mở và xây dựng bộ công cụ các phần tử ở mức cao qua đó cho phép nhà phát triển ứng dụng có thể nhanh chóng xây dựng ứng dụng và khả năng tạo ra các ừng dụng mới phong phú hơn.

Các lợi ích mà tầng này cung cấp cho nhà phát triển:

* *Với các nhà sản xuất phần cứng điện thoại:* có thể tùy biến hệ điều hành để phù hợp với cấu hình phần cứng điện thoại mà họ sản xuất cũng như để có nhiều mẫu mã điện thoại để phù hợp với thị hiếu của người tiêu dùng.
* *Với nhà phát triển ứng dụng:* Cho phép các lập trình viên có thể sử dụng các API ở tầng trên mà không cần phải hiển rõ cấu trúc bên dưới. Một tập hợp các API rất hữu ích được xây dựng sẵn như hệ thống định vị, các dịch vụ chạy nền, liên lạc giữa các ứng dụng, các thành phần giao diện, điều này tạo điều kiện thuận lợi cho các lâp trình viên làm việc vì họ chỉ cần quan tâm đến nội dung mà ứng dụng của họ mà không cần phải quan tâm đến những thứ liên quan khác.



*Hình 1. 23 Tầng Application Framework trong kiến trúc Android*

Các thành phần của Application Framework:

* ACTIVITY MANAGER: Activity Manager là những gì quản lý vòng đời của các ứng dụng. Nó cũng duy trì việc trở lại ngăn xếp chung vì vậy các ứng dụng mà đang chạy trong các quá trình khác nhau có thể có một trải nghiệm điều hướng tích hợp thông suốt.
* PACKAGE MANAGER: Package manager là những gì theo dõi các ứng dụng được cài đặt trên thiết bị của bạn. Vì vậy, nếu bản tải về các ứng dụng mới qua môi trường hoặc cài đặt các ứng dụng, đó là quản lý gói nó chịu trách nhiệm cho việc theo dõi những gì bạn có và những khả năng của mỗi ứng dụng của bạn là những gì.
* WINDOW MANAGER: Quản lý việc xây dựng và hiển thị các giao diện người dùng cũng như tổ chức quản lý các giao diện giữa các ứng dụng.
* TELEPHONY MANAGER: Telephony Manager có chứa các API mà chúng ta có thể sử dụng để xây dựng các ứng dụng điện thoại, cung các sự kiện liên quan đến điện thoại như gọi đến/gọi đi.
* LOCATION MANAGER: Cho phép xác định vị trí của điện thoại dựa vào hệ thống định vị toàn cầu GPS và Google Maps.
* NOTIFICATION MANAGER: Cho phép tất cả các ứng dụng hiển thị các thông báo tùy chỉnh trên thanh trạng thái.
* CONTENT PROVIDERS: Cho phép truy cập dữ liệu từ các ứng dụng khác (như tiếp xúc) quản lý danh bạ hoặc chia sẽ dữ liệu của riêng mình.
* RESOURCE MANAGER: Quản lý tất cả các tài nguyên tĩnh của các ứng dụng bao gồm các file hình ảnh, âm thanh, layout...tất cả những phần bên ngoài của một ứng dụng mà không phải code.
* VIEW SYSTEM: được dùng để xây dựng giao diện một ứng dụng bao gồm các văn bản, nút bấm và cả những web nhúng, kể cả những thứ như điều phối sự kiện, thiết kế bản vẽ.
* ***Tầng Applications***
* Đây là tầng ứng dụng giao tiếp với người dùng, bao gồm các ứng dụng cơ bản được cài đặt gắn liền với hệ điều hành như: phone, contacts, web, SMS, Calendar …
* Các ứng dụng trong tầng Application được phát triển bằng ngôn ngữ java, phần mở rộng là file apk.
* Khi một ứng dụng được chạy, nó có một phiên bản Virtual Machine được dựng lên để phục vụ cho nó. Nó có thể là một Active Program: chương trình có giao diện với người sử dụng hoặc là một Background: chương trình chạy nền hay là dịch vụ.



*Hình 1. 24 Tầng Applications trong kiến trúc Android*

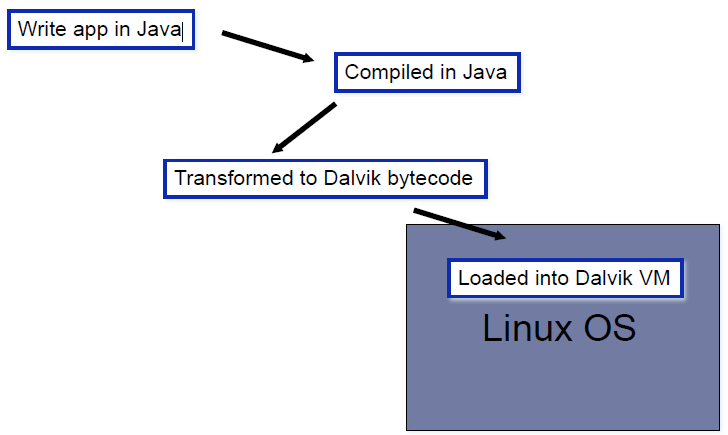
### 1.1.6 Máy ảo Dalvik

Máy ảo Dalvik (DalvikVM) được đặt bởi Bornstein theo tên một làng chài ở Iceland, nơi tổ tiên của ông sống, đây là một máy ảo register-based. Nói cách khác, “register-based” mà chúng ta đã nói ở đầu là khác so với máy ảo Java thông thường (trên các hệ thống Windows, Linux… của Sun).

Dalvik là tên của máy ảo Android. Nó chỉ là một máy ảo thông dịch thực thi các file ở định dạng thực thi của Dalvik (Dalvik Executable - \*.dex), một định dạng được tối ưu hóa cho lưu trữ và sử dụng bộ nhớ hiệu quả. Đây là một máy ảo register-based, và nó có thể chạy các class được biên dịch bởi một trình biên dich Java đã được chuyển thành định dạng tương thích với nó bởi công cụ “dx”. Máy ảo chạy ở mức trên cùng của nhân Linux 2.6 và dựa trên các chức năng phía dưới của hệ thống (giống như luồng và trình quản lý bộ nhớ mức thấp). DalvikVM cũng được tối ưu hóa để chạy đa thể hiện với một very low memory-footprint. Một vài máy ảo cung cấp cơ chế bảo vệ các ứng dụng của chúng không bị ảnh hưởng bởi một ứng dụng gặp lỗi khác.

Cơ chế máy ảo Dalvik?

* Ngay khi thực hiện ứng dụng, quá trình Linux kết hợp với các ứng dụng Android làm việc trong môi trường máy ảo riêng của mình.
* Khi một ứng dụng chạy trong quá trình Linux riêng của mình, hệ thống Android bắt đầu quá trình khi bất kỳ của các thành phần của ứng dụng cần được thực hiện và tắt quá trình khi nó không còn cần thiết.



Hình 1. 25 Cơ chế máy ảo Dalvik

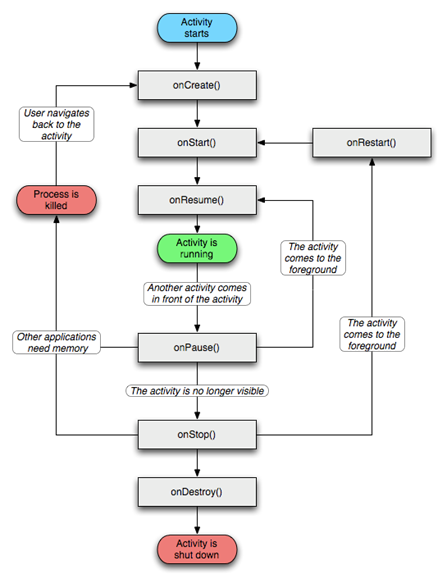
***Sự khác biệt với máy ảo Java thông thường?***

Máy ảo Java có thể tìm thấy trên hầu hết các máy tính ngày nay là stack-based VM. DalvikVM thì khác, là một register-based, bởi vì một vi xử lý di động được tối ưu hóa cho thao tác register-based. Máy ảo register-based cũng cho phép thời gian thực thi nhanh hơn.

### 1.1.7 Các thành phần trong một ứng dụng Android

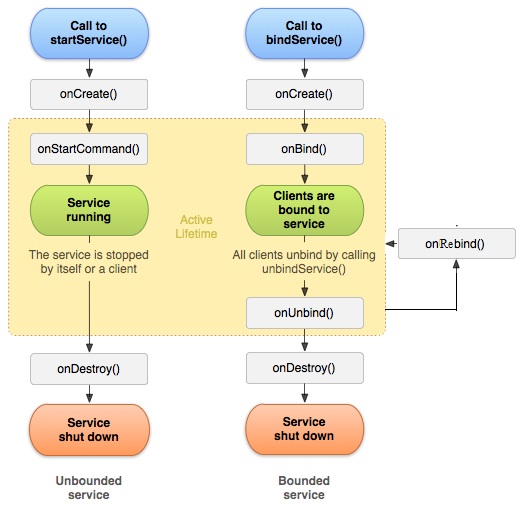
Trong một ứng dụng Android cơ bản thường có 4 loại thành phần khác nhau: Activity, Service, Broadcast Receiver và Content Provider. Đây là những thành phần chính cấu thành nên ứng dụng Android, bắt buộc phải khai báo trong AndroidManifest.

* ***Activity***
* Trong một ứng dụng Android thường có một hoặc nhiều Activity.
* Một Activity đại diện cho duy nhất một giao diện người dùng hiển thị lên màn hình. Một Activity sẽ có vòng đời riêng biệt, độc lập với tất cả các Activity khác trong một một ứng dụng Android.
* Vòng đời của một Activity:



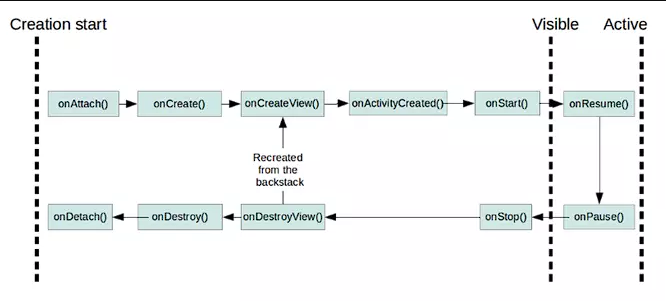
Hình 1. 26 Vòng đời của một Activity

* **onCreate():** là trạng thái được triệu gọi khi Activity được khởi tạo. Trạng thái này chỉ được triệu gọi duy nhất một lần trong xuất một vòng đời của Activity.
* **onStart():** được gọi đến trong hai trường hợp:
* Khi Activity được khởi tạo mới. Onstart được gọi sau khi onCreate được khởi tạo.
* Khi Activity được hiện thị trở lại (Activity từ trạng thái Background chuyển sang trạng thái Foreground)
* **onPause():** được gọi đến khi activity trở về trạng thái Background.
* **onResume():** cũng giống như trạng thái onStart nhưng nó được gọi đến sau khi onStart được khởi tạo.
* **onStop():** được gọi đến khi vòng đời của Activity kết thúc hoặc khi Activity trở về trạng thái Background (được triệu gọi sau onPause().
* **onDestroy():** là trạng thái được gọi đến sau cùng khi và chỉ khi Activity kết thúc vòng đời.
* ***Services***
* Services là một trong những thành phần chính trong hệ điều hành android. Services chạy dưới nền background thực hiện các công việc liên tục, lâu dài nhưng không ảnh hưởng đến giao diện người dùng.
* Vòng đời của services:



Hình 1. 27 Vòng đời của một Services

* Một vòng đời của services có thể được khởi tạo bằng 2 cách:
* **startService():** ta dùng phương thức này khi thực hiện một hành động đơn và hành động đó không trả về kết quả(ví dụ như chơi nhạc).
* **bindService():** cung cấp 1 interface dạng client-server cho phép dứng dụng có thể tương tác với service.
* ***Fragment***
* Trong các ứng dụng android, tại một thời điểm, chỉ có một Activity được hiển thị duy nhất trên màn hình. Chúng ta muốn chia màn hình ra nhiều phần để dễ sử dụng thì fragment đáp ứng điều đó
* Fragment là một thành phần android độc lập, được sử dụng bởi một activity, giống như một sub-activity. Fragment có vòng đời và UI riêng. Các Fragment thường có một file java đi kèm với file giao diện xml. Các fragment không có file giao diện xml thường được gọi là headless fragments.
* Fragment sử dụng method getActivity() để lấy ra Activity bố.
* Fragment được định nghĩa trong file xml của activity (static definition) hoặc có thể sửa đổi fragment khi đang chạy (dynamic definition)
* Vòng đời của một fragment



Hình 1. 28 Vòng đời của một Fragment

* **onAttach():** hàm này thực hiện tạo tham chiếu từ một fragment đến activity đã khởi tạo nó, và thực hiện một số bước trong quá trình khởi tạo.
* **onCreate():** thực hiện khởi tạo fragment.
* **onCreateView():** thực hiện tạo giao diện(view), trả về view là giao diện file xml tương ứng fragment. ko nên tương tác với activity trong hàm này bởi vì activity chưa được khởi tạo đầy đủ. Không cần thực hiện hàm này với các fragment không có header.
* **onActivityCreated():** thực hiện hoàn thành nốt việc khởi tạo activity và fragment. Trong bước này chúng ta có thể gọi findViewById().
* **onStart():** thực hiện việc hiển thị fragment lên màn hình.
* **onResume():** fragment chính thức hoạt động hoàn toàn.
* **onPause():** fragment bị tạm dừng hoạt động, nó vẫn có thể được nhìn thấy.
* **onStop():** fragment bị ẩn.
* **onDestroyView():** giao diện(view) của fragment bị hủy. Nếu nó được gọi quay lại, nó sẽ quay trở lại thực hiện hàm onCreateView().
* **onDestroy():** bị hủy.
* **onDetach():** bị hủy hoàn toàn.
* ***Content Providers***
* Content Providers là thành phần để chia sẻ dữ liệu giữa các ứng dụng với nhau.
* ***Broadcast Receivers***
* Broadcast Receivers được dùng để thu nhận các trạng thái từ hệ thống hoặc từ các ứng dụng.
* Vòng đời của Broadcast Receivers chỉ tồn tại trong khi onReciver được gọi, khi phương thức này kết thúc thì Receivers cũng kết thúc và không còn trong trạng thái hoạt động nữa.

Ví dụ: bạn viết một chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn cần một BR để nhận biết các Intent là các cuộc gọi tới.

### 1.1.8 Công cụ phát triển hệ điều hành Android

* ***Android SDK***

Công cụ phát triển ứng dụng Android (Android SDK) bao gồm một bộ công cụ phát triển toàn diện bao gồm một trình gỡ lỗi, các thư viện, thiết bị giả lập dựa trên QEMU, tài liệu, mẫu ví dụ tham khảo, và hướng dẫn. Hiện tại Android SDK được hỗ trợ các máy tính chạy hệ điều hành Linux, Mac OS X 10.5.8 hoặc mới hơn, Windows XP, Windows Vista, Windows 7 và mới hơn. Hỗ trợ chính thức môi trường phát triển tích hợp (IDE) Eclipse bằng cách sử dụng ADT Plugin, bên cạnh đó còn có IntelliJ IDEA IDE (tất cả các phiên bản) hỗ trợ đầy đủ lập trình Android, và NetBeans IDE cũng hỗ trợ phát triển Android thông qua một Plugin. Ngoài ra, các nhà phát triển có thể sử dụng trình soạn thảo bất kỳ để chỉnh sửa các tập tin Java và XML, sau đó sử dụng công cụ dòng lệnh (Java Development Kit và Apache Ant được yêu cầu) để tạo, biên dịch và gỡ lỗi các ứng dụng Android cũng như kiểm soát các thiết bị Android (ví dụ, kích hoạt khởi động lại máy, cài đặt phần mềm từ xa).

Cải tiến của Android SDK đi đôi với sự phát triển của toàn bộ nền tảng Android. SDK cũng hỗ trợ các phiên bản cũ hơn của nền tảng Android trong trường hợp các nhà phát triển muốn phát triển ứng dụng cho thiết bị cũ. Các thành phần của Android SDK dễ dàng tải xuống vì thế các phiên bản của Platform có thể tải về phiên bản mới nhất cũng như cũ hơn phục vụ quá trình kiểm thử.

Các ứng dụng Android được đóng gói thành tệp apk. Và dữ liệu được lưu trữ trong thưc mục */data/app* trên hệ điều hành Android. Trong tệp APK chứa tệp tin .dex (mã nguồn được dịch thành mã byte chạy trên bộ xử lý dalvik), và các tập tin tài nguyên, vv…

* ***Eclipse***

Eclipse là một nền tảng phát triển mã nguồn mở dựa trên Java có thể mở rộng được. Eclipse bao gồm một tập hợp các dịch vụ để xây dựng các ứng dụng từ các thành phần plug-in. Eclipse được tạo thành bởi một tập hợp các plug-in chuẩn JDT.

Dự án Eclipse được IBM bắt đầu vào tháng 11 năm 2001. Các dự án của Eclipse tập trung vào việc xây dựng nền tảng mã nguồn mở bao gồm các Framework có thể mở rộng, các tool và thư viện để xây dựng, triển khai và quản lý các phần mềm.

Eclipse là một tổ chức phi lợi nhuận, là nơi phát triển cộng đồng mã nguồn mở và hệ thống các sản phẩm, dịch vụ. Những người phát triển plug-in cho Eclipse hoặc những người sử dụng Eclipse như là nền tảng cho các ứng dụng phát triển phần mềm cần tuân thủ giấy phép EPL.

Kiến trúc Eclipse nền tảng bao gồm nhiều thành phần. Các chức năng của thành phần lõi tách biệt với các chức năng của phần giao diện.

Kiến trúc nền tảng Eclipse cung cấp các công cụ cùng cơ chế sử dụng và quy định để có thể tích hợp các công cụ một cách dễ dàng. Những cơ chế này được thể hiện thông qua giao diện API, lớp và phương thức. Ngoài ra, nền tảng Eclipse còn cung cấp các framework để dễ dàng phát triển các công cụ mới.

Eclipse SDK gồm Eclipse Platform, JDT và PDE. JDT và PDE là các plug-in được gắn vào Platform. Eclipse Platform được xây dựng dựa trên cơ chế “*phát hiện, tích hợp và chạy*”. Plug-in là đơn vị chức năng nhỏ nhất có thể được phát triển và phân phối một cách riêng biệt. Thông thường, công cụ nhỏ được viết trong một plug-in riêng biệt và công cụ phức tạp được tách ra thành nhiều plug-in.

**- SWT** (Standard Widget Toolkit) là một gói công cụ mã nguồn mở được phát triển bởi IBM, cung cấp cho các lập trình viên Java giải pháp để phát triển giao diện đồ họa người dùng.

**- JFace** là toolkit để xây dựng giao diện người dùng cấp cao, JFace là tầng trên cùng của SWT, cung cấp các lớp thuộc mô hình MVC để phát triển các ứng dụng đồ họa dễ dàng hơn.

Những người muốn phát triển Eclipse sẽ sử dụng PDE (Plug-in Development Environment) để bổ sung thêm các plug-in mới. [10]

## 1.2 Quản trị CSDL với SQLite

### 1.2.1 Quản trị cơ sở dữ liệu

SQLite là một hệ quản trị cơ sở dữ liệu quan hệ nhúng, chứa trong nó là một thư viện nhỏ bằng ngôn ngữ C. SQLite thực hiện hầu hết chuẩn SQL, sử dụng các cú pháp SQL linh hoạt nhưng không đảm bảo ràng buộc toàn vẹn. Hành động đọc có thể thực hiện đa nhiệm, nhưng hành động ghi thì thực hiện tuần tự.

Hệ cơ sở dữ liệu SQLite Database chính là hệ thống mã nguồn mở được sử dụng rộng rãi trong các ứng dụng Android.

Cơ sở dữ liệu được tạo cho một ứng dụng thì chỉ ứng dụng đó có quyền truy cập và sử dụng, các ứng dụng khác thì không. Khi đã được tạo, cơ sở dữ liệu SQLite được chứa trong thư mục /data/data/<package\_name>/database.

Khác với hầu hết các hệ quản trị cơ sở dữ liệu khác, SQLite được thiết kế theo tiêu chí đơn giản hóa tối đa:

* Đơn giản để quản trị.
* Đơn giản để hoạt động.
* Đơn giản để nhúng vào một chương trình lớn hơn.
* Đơn giản để bảo trì và tùy chỉnh.

SQLite là một lựa chọn phổ biến để lưu trữ dữ liệu cục bộ. Có thể kết hợp với nhiều ngôn ngữ lập trình khác nhau. Nó là công cụ cơ sở dữ liệu được sử dụng rộng rãi nhất trong các trình duyệt web, hệ điều hành, hệ thống nhúng và đặc biệt là trong các thiết bị di động.

### 1.2.2 Thiết kế của SQLite

Không giống như các hệ quản trị cơ sở dữ liệu client – server, SQLite không có các tiến trình giao tiếp ứng dụng client – server. Thay vào đó SQLite có các thư viện được liên kết và trở thành một phần không thể thiếu cho ứng dụng. Ứng dụng gọi những chức năng của SQLite trực tiếp thông qua lời gọi hàm đơn giản, làm giảm độ trễ của truy xuất dữ liệu. Do đó tiến trình gọi hàm để thực hiện các chức năng sẽ đơn giản và hiệu quả hơn quá trình giao tiếp mà các hệ quản trị cơ sở dữ liệu client – server thực hiện. SQLite lưu trữ toàn bộ cơ sở dữ liệu chỉ trong một tập tin. Có thể sử dụng SQLite để lưu trữ dữ liệu ứng dụng tốt hơn XML vì không cần phải phân tích cú pháp và ghi tập tin. Sử dụng SQLite dễ dàng truy cập các thao tác trên dữ liệu và độc lập nền tảng.

* ***Đặc tính của SQLite***
* Giao tác là nguyên tử, nhất quán, độc lập và bền vững ngay cả sau khi hệ thống bị treo hoặc mất điện.
* Không cần cài đặt hay quản trị phức tạp.
* Đáp ứng hầu hết chuẩn SQL92
* Toàn bộ cơ sở dữ liệu chứa trong một tập tin không phụ thuộc nền tảng.
* Hỗ trợ cơ sở dữ liệu có kích thước terabyte và chuỗi kích thước gigabyte.
* Mã nguồn nhỏ, khoảng 350KB với cấu hình đầy đủ tính năng và 200KB với các tính năng tùy chọn.
* Nhanh hơn các hệ quản trị cơ sở dữ liệu client – server ở hầu hết các thao tác.
* API đơn giản, dễ sử dụng.
* Được viết bằng ANSI-C kết hợp với nhiều ngôn ngữ riêng biệt.
* Có sẵn tập tin mã nguồn để dễ dàng đưa vào một dự án.
* Độc lập, không phụ thuộc bên ngoài.
* Độc lập nền tảng: UNIX (Linux, Mac OS-X, Android, iOS) và Windows (Win32, WinCE, WinRT) được hỗ trợ. Dễ dàng chuyển đổi để sử dụng trên nhiều hệ thống khác nhau.
* Mã nguồn mở và có thể được sử dụng với mọi mục đích.
* Giao diện quản trị bằng dòng lệnh.
* Kiểu dữ liệu: NULL, INTEGER, REAL, TEXT, BLOG.
* ***Sử dụng SQLite:***
* Trường hợp nên sử dụng SQLite:
  + - * Sử dụng cho các thiết bị nhúng và các ứng dụng cho các thiết bị nhúng.
      * Thích hợp cho các website có lượng truy cập trung bình.
      * Thích hợp với cơ sở dữ liệu nội bộ và tạm thời.
      * Cơ sở dữ liệu phục vụ cho việc giảng dạy.
* Trường hợp không nên sử dụng SQLite:
  + - * Ứng dụng client – server.
      * Sử dụng cho các trang web lớn, lượng truy cập cao.
      * Sử dụng cho các tập tin lớn.
* ***Đối tượng trong SQLite***
* Đối tượng kết nối SQLiteOpenHelper:
  + - Vai trò như một đối tượng kết nối.
    - Khởi tạo database và quản lý phiên bản.
    - Cần tham chiếu đến một đối tượng Context.
    - Phương thức:
* **onCreate(SqliteDatabase db):** Tạo bảng và khởi tạo giá trị cho các bảng.
* **onUpgrade(SqliteDatabase db, int oldVer, int newVer):** Xóa bảng cũ và tạo lại.
* **onOpen(SqliteDatabase db):** Mở một SqliteDatabase – ít sử dụng.
* **close():** Đóng đối tượng SQLiteOpenHelper hiện tại tới DB.
* **getReadableDatabase():** Trả về đối tượng SqliteDatabase chỉ đọc.
* **getWritableDatabase():** Trả về đối tượng SqliteDatabase đọc và ghi.
* Đối tượng thực thi SqliteDatabase:
  + - Common DML (Data Modification Language) database command.
    - Đối tượng trực tiếp sẽ thực thi các câu lệnh SQL.
    - Phương thức:
* **execSQL(String query):** Thực hiện câu lệnh sql thông thường không trả và kết quả.
* **delete(String table, String where, String argWhere):** Xóa dữ liệu trên một bảng.
* **rawQuery(String sql, String[] selectionArgs):** Thực thi câu sql với tham số nếu có, trả về một Cursor.
* **query(…):** Thực hiện câu lệnh trên một table và trả về đối tượng Cursor.
* **close():** Đóng – kết thúc đối tượng kết nối DB hiện tại.
* Đối tượng kết xuất dữ liệu Cursor:
  + - Đối tượng trả về từ một phương thức họ query.
    - Chưa chỉ tới đâu cả hoặc null nếu không có.
    - Phương thức:
* **moveToFirst():** Chuyển tới bản ghi đầu tiên.
* **moveToNext():** Chuyển tới bản ghi tiếp.
* int columnIndex = getColumnIndex(String columnName)
* **String data = getString(columIndex):** Lấy dữ liệu ở cột columnIndex dạng String.
* **int data = getInt(columnIndex):** Lấy dữ liệu ở cột columnIndex dạng int ...

# CHƯƠNG 2: XÁC ĐỊNH YÊU CẦU VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG

**Nội dung của chương:**

**Mục đích và phạm vi hệ thống:**Trước tiên xác định phạm vi hệ thống, đặt ra các mục tiêu cần đạt được của hệ thống:

**Yêu cầu hệ thống:**Xác định yêu cầu của hệ thống và các chức năng của hệ thống

**Phân tích hệ thống:**Xác định các tiến trình và xây dựng biều đồ dữ liệu liên quan.

## 2.1 Mục tiêu và phạm vi của hệ thống

### 2.1.1 Phạm vi ứng dụng

- Tên ứng dụng: SVMC

- Phạm vi sử dụng: Áp dụng cho thiết bị di động trên hệ điều hành Android.

- Đối tượng sử dụng: Chỉ cần một thiết bị di động sử dụng hệ điều hành Android, bất cứ ai cũng có thể sử dụng ứng dụng này.

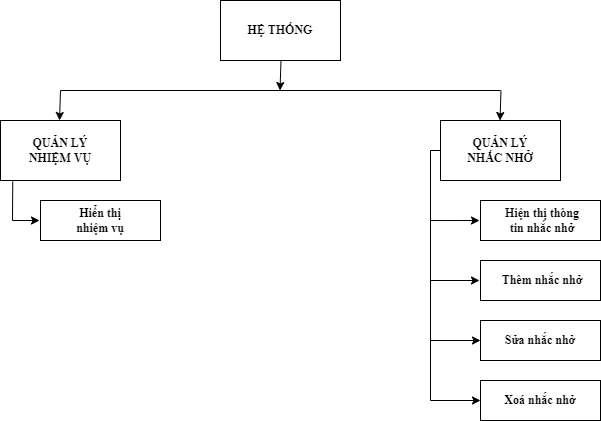
### 2.1.2. Mục tiêu của ứng dụng

- Ứng dụng trực quan, dễ thao tác, sử dụng với người dùng

- Cơ động có thể dễ dàng quản lý được thông tin các ngày nhắc nhở

## 2.2 Phân tích thiết kế hệ thống

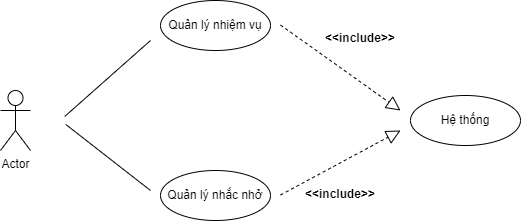
### 2.2.1 Biểu đồ tổng quan chức năng của hệ thống



Hình 2. 1 Biểu đồ tổng quan hệ thống

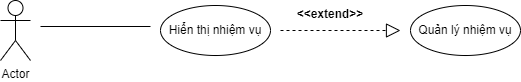
### 2.2.2 Biểu đồ ca sử dụng

***a. Biểu đồ ca sử dụng chính***



Hình 2. 2 Biểu đồ ca sử dụng hệ thống

***b. Biểu đồ ca sử dụng chức năng quản lý nhiệm vụ***

**

*Hình 2. 3 Biểu đồ ca sử dụng quản lý nhiệm vụ*

***c.*** ***Biểu đồ ca sử dụng chức năng quản lý nhắc nhở***



Hình 2. 4 Biểu đồ ca sử dụng quản lý nhắc nhở

***2.2.2 Mô tả chi tiết chức năng quản lý nhắc nhở***

|  |  |
| --- | --- |
| Tên use case | Quản lý nhắc nhở |
| Tác nhân chính | Quản trị hệ thống |
| Mức | 2 |
| Tiền điều kiện | Đã khởi động app |
| Hậu điều kện | Hệ thống cập nhập lại dữ liệu |
| Kích hoạt | Người dùng chọn các chức năng quản lý nhắc nhở |
| Chuỗi sự kiện chính:   1. Hệ thống yêu cầu lựa chọn chức năng: thêm, sửa, xóa 2. Người dùng lựa chọn chức năng 3. Hệ thống hiển thị giao diện nhập thông tin cần thay đổi 4. Người dùng nhập thông tin 5. Người dùng chọn đồng ý 6. Hệ thông kiểm tra thông tin 7. Hệ thống cập nhập thông tin vào cơ sở dữ liệu 8. Hệ thống thông báo thành công và quay lại giao diện chính | |
| Ngoại lệ:   1. Hệ thống thông báo thông tin không hợp lệ 2. Hệ thống yêu cầu nhập lại thông tin 3. Người dùng nhập lại hoặc thoát | |

Bảng 2. 1 Mô tả luồng sự kiện trong quản lý nhắc nhở

***Luồng sự kiện***

***Tiền điều kiện***

* Người dùng sử dụng ứng dụng

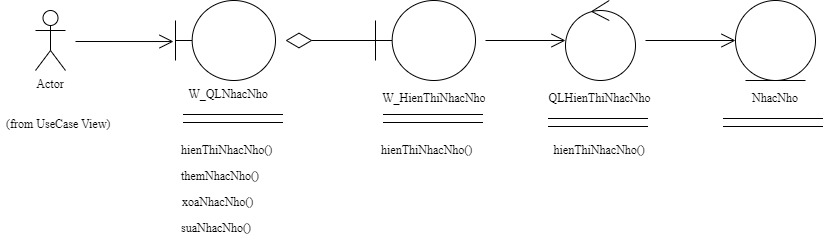
***Luồng sự kiện chính***

* Ca sử dụng bắt đầu khi người dùng kích hoạt ứng dụng
* Hệ thống đưa ra lựa chọn: thêm thông tin, sửa thông tin, xóa thông tin, hiển thị thông tin, thoát
* Nếu người dùng chọn thêm thông tin: thực hiện luồng sự kiện S1
* Nếu người dùng chọn sửa thông tin : thực hiện luồng sự kiện S2
* Nếu người dùng chọn xóa thông tin : thực hiện luồng sự kiện S3
* Nếu người dùng chọn hiển thị thông tin : thực hiện luồng sự kiện S4
* Nếu người dùng chọn thoát: ca sử dụng kết thúc

***Luồng sự kiện phụ***

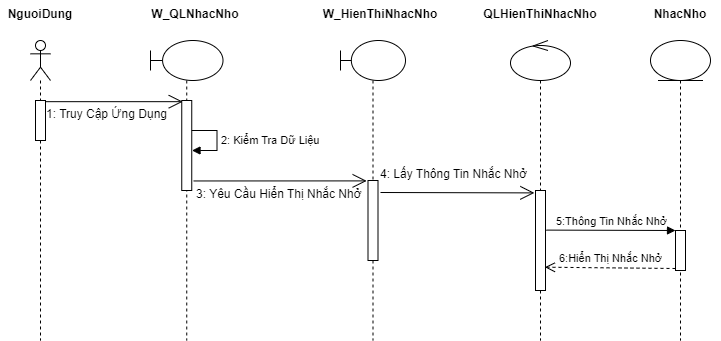
* S1: thêm thông tin
* Hệ thống hiển thị giao diện thêm thông tin mới
* Hệ thống yêu cầu nhập thông tin
* Người dùng nhập thông tin và chọn “✓” để hoàn thành
* Hệ thống kiểm tra thông tin và cập nhập thông tin vao cơ sở dữ liệu
* Ca sử dụng bắt đầu lại.
* S2: sửa thông tin
* Hệ thống yêu cầu người dùng chọn nhắc nhở cần thay đổi thông tin
* Người dùng chọn nhắc nhở cần sửa
* Hệ thống hiển thị giao diện thay đổi thông tin
* Người dùng nhập thông tin cần thay đổi và chọn “✓”
* Hệ thống kiểm tra thông tin và cập nhập vào cơ sở dữ liệu
* Ca sử dụng bắt đầu lại.
* S3: xóa thông tin
* Hệ thống yêu cầu người dùng chọn nhắc nhở muốn xóa
* Người dùng chọn nhắc nhở và chọn “Xóa”
* Hệ thống kiểm tra thông tin
* Hệ thống yêu cầu người dùng xác nhận có chắc chắc xóa hay không
* Người dùng xác nhận
* Hệ thống thực hiện xóa và cập nhập lại cơ sở dữ liệu
* Ca sử dụng bắt đầu lại.
* S4: Hiển thị thông tin
* Hệ thống yêu cầu người dùng chọn hiển thị
* Người dùng chọn hiển thị nhắc nhở
* Hệ thống kiểm tra danh sách nhắc nhở
* Hệ thống hiển thị danh sách nhắc nhở
* Ca sử đụng bắt đầu lại

**a. Kịch bản hiển thị nhắc nhở**

**

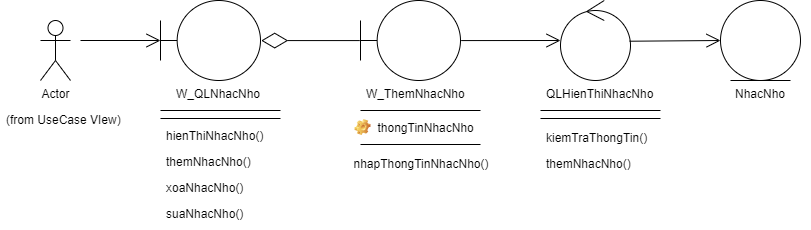
*Hình 2. 5 Biểu đồ hoạt động kịch bản hiển thị nhắc nhở*

* Lớp biên: W\_QLNhacNho, W\_HienThiNhacNho
* W\_ QLNhacNho:
* Thao tác
* Hiển thị nhắc nhở
* Thêm nhắc nhở
* Sửa nhắc nhở
* Xóa nhắc nhở
* W\_ HienThiNhacNho:
* Thao tác
* Hiển thị nhắc nhở
* Lớp điều khiển: QLHienThi
* QLHienThi:
* Thao tác
* Hiển thị nhắc nhở
* Lớp thực thể: NhacNho



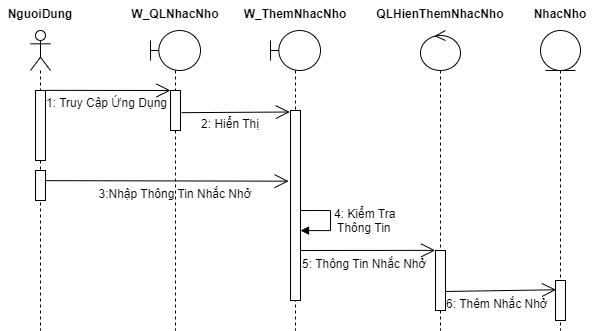
Hình 2. 6 Biểu đồ trình tự kịch bản hiển thị nhắc nhở

**b. Kịch bản thêm nhắc nhở**



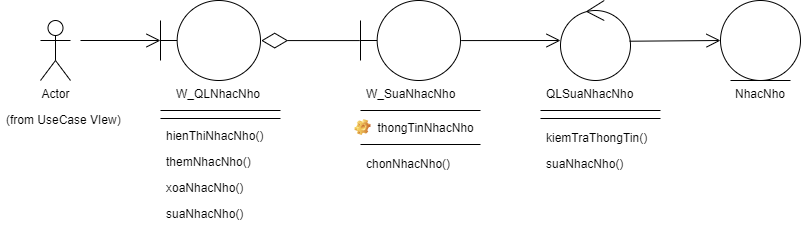
*Hình 2. 7 Biểu đồ hoạt động kịch bản thêm nhắc nhở*

* Lớp biên: W\_QLNhacNho, W\_ThemNhacNho
* W\_ QLNhacNho:
* Thao tác
* Hiển thị nhắc nhở
* Thêm nhắc nhở
* Sửa nhắc nhở
* Xóa nhắc nhở
* W\_ ThemNhacNho:
* Thuộc tính
* Thông tin nhắc nhở
* Thao tác
* Nhập thông tin nhắc nhở
* Lớp điều khiển: QLThemNhacNho
* QLHienThi:
* Thao tác
* Kiểm tra thông tin
* Thêm thông tin
* Lớp thực thể: NhacNho

****

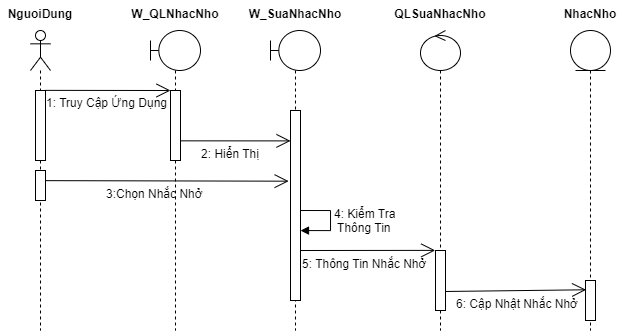
*Hình 2. 8 Biểu đồ trình tự kịch bản thêm nhắc nhở*

c. Kịch bản sửa nhắc nhở



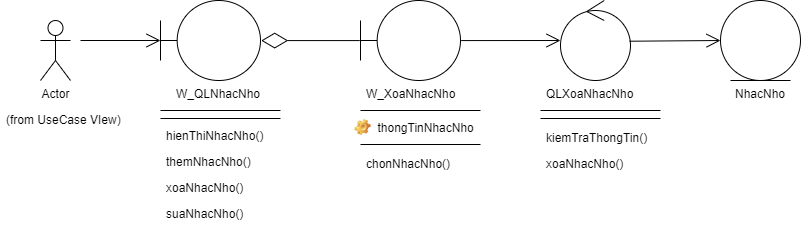
*Hình 2. 9 Biểu đồ hoạt động kịch bản sửa nhắc nhở*

* Lớp biên: W\_QLNhacNho, W\_SuaNhacNho
* W\_ QLNhacNho:
* Thao tác
* Hiển thị nhắc nhở
* Thêm nhắc nhở
* Sửa nhắc nhở
* Xóa nhắc nhở
* W\_ SuaNhacNho:
* Thuộc tính
* Thông tin nhắc nhở
* Thao tác
* Chọn nhắc nhở
* Lớp điều khiển: QLSuaNhacNho
* QLSuaNhacNho:
* Thao tác
* Kiểm tra thông tin
* Sửa nhắc nhở
* Lớp thực thể: NhacNho



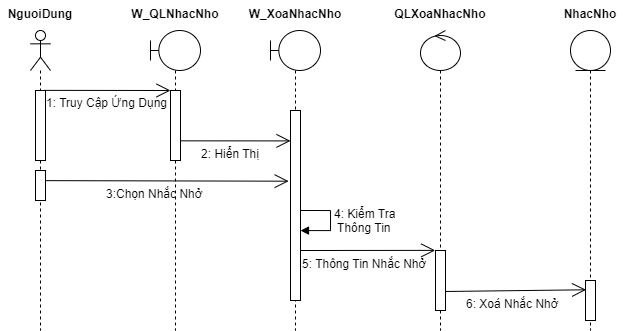
*Hình 2. 10 Biểu đồ trình tự kịch bản sửa nhắc nhở*

d. Kịch bản xóa nhắc nhở

**

*Hình 2. 11 Biểu đồ hoạt động kịch bản xóa nhắc nhở*

* Lớp biên: W\_QLNhacNho, W\_XoaNhacNho
* W\_ QLNhacNho:
* Thao tác
* Hiển thị nhắc nhở
* Thêm nhắc nhở
* Sửa nhắc nhở
* Xóa nhắc nhở
* W\_ XoaNhacNho:
* Thuộc tính
* Thông tin nhắc nhở
* Thao tác
* Chọn nhắc nhở
* Lớp điều khiển: QLXoaNhacNho
* QLXoaNhacNho:
* Thao tác
* Kiểm tra thông tin
* Xóa nhắc nhở
* Lớp thực thể: NhacNho

**

Hình 2. 12 Biểu đồ trình tự kịch bản xóa nhắc nhở

***2.2.4 Mô tả chức năng quản lý nhiệm vụ***

|  |  |
| --- | --- |
| Tên use case | Quản lý nhiệm vụ |
| Tác nhân chính | Quản trị hệ thống |
| Mức | 2 |
| Tiền điều kiện | Đã khởi động ứng dụng |
| Hậu điều kện | Hệ thống hiển thị nhiệm vụ |
| Kích hoạt | Người dùng khởi động ứng dụng |
| Chuỗi sự kiện chính:  Hệ thống hiển thị thông tin nhiệm vụ: Tên nhiệm vụ, Ngày thực hiện, Thời Gian thực hiện | |
| Ngoại lệ:  Thông tin nhiệm vụ không hợp lệ | |

Bảng 2. 2 Mô tả luồng sự kiện trong quản lý nhiệm vụ

***Luồng sự kiện***

***Tiền điều kiện***

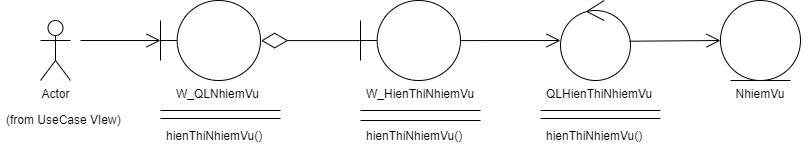
* Người dùng sử dụng ứng dụng

***Luồng sự kiện chính***

* Ca sử dụng bắt đầu khi người dùng kích hoạt ứng dụng
* Hệ thống tự động hiển thị nhiệm vụ
* Nếu người dùng chọn thoát: ca sử dụng kết thúc

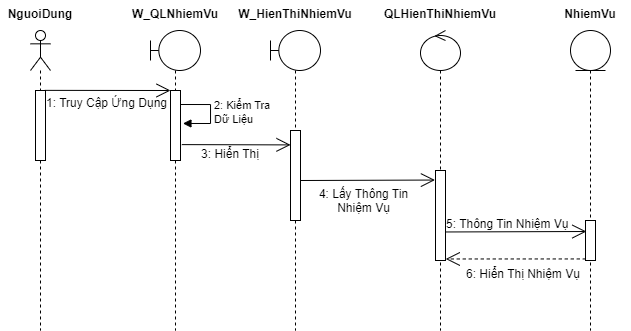
***Luồng sự kiện phụ***

* S1: Xem thông tin các nhiệm vụ
* Hệ thống hiển thị giao diện nhiệm vụ
* Người dùng chọn một trong hai mục “HÔM NAY” và “TẤT CẢ” để xem các nhiệm vụ
* Hệ thống kiểm tra thông tin và hiển thị lại nhiệm vụ
* Ca sử dụng bắt đầu lại.

**

Hình 2. 13 Biểu đồ hoạt động kịch bản hiển thị nhiệm vụ

* Lớp biên: W\_QLNhiemVu, W\_HienThiNhiemVu
* W\_ QLNhiemVu:
* Thao tác
* Hiển thị nhiệm vụ
* W\_ HienThiNhiemVu:
* Thao tác
* Hiển thị nhiệm vụ
* Lớp điều khiển: QLHienThiNhiemVu
* QLHienThiNhiemVu:
* Thao tác
* Hiển thị nhiệm vụ
* Lớp thực thể: NhiemVu

**

*Hình 2. 14 Biểu đồ trình tự kịch bản hiển thị nhiệm vụ*

## 2.3 Thiết kế cơ sở dữ liệu

***2.3.1*** ***Cấu trúc bảng cơ sở dữ liệu***

|  |  |
| --- | --- |
| **Nhiệm vụ** | |
| **idNV** | int |
| name | text |
| time | text |
| repeat | int |
| date | text |
| trangthai | int |

Bảng 2. 3 Bảng cấu trúc cơ sở dữ liệu

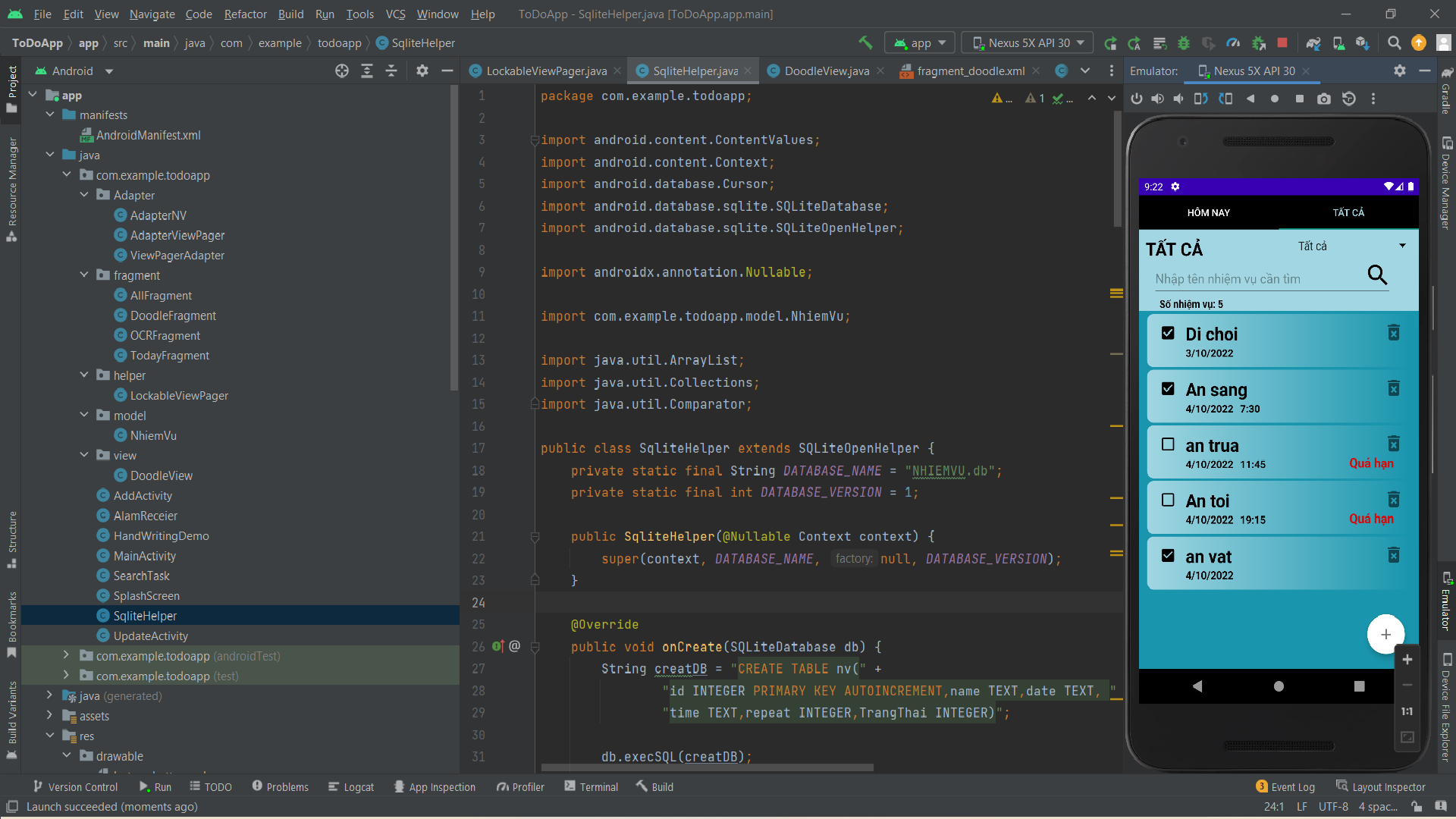
# CHƯƠNG 3: THIẾT KẾ DỮ LIỆU VÀ XÂY DỰNG CHƯƠNG TRÌNH

## 3.1 Thiết kế MODUL

### 3.1.1 Công cụ sử dụng

* Android studio
* CSDL: SQLiteHelper

### 3.1.2 Thông tin phần mềm

**

Hình 3. 1 Thực thi đề tài trên Android studio

* Ngôn ngữ sử dụng: Android
* Tên phần mềm: SVMC
* Version: 1

Android SDK version requirement: 28

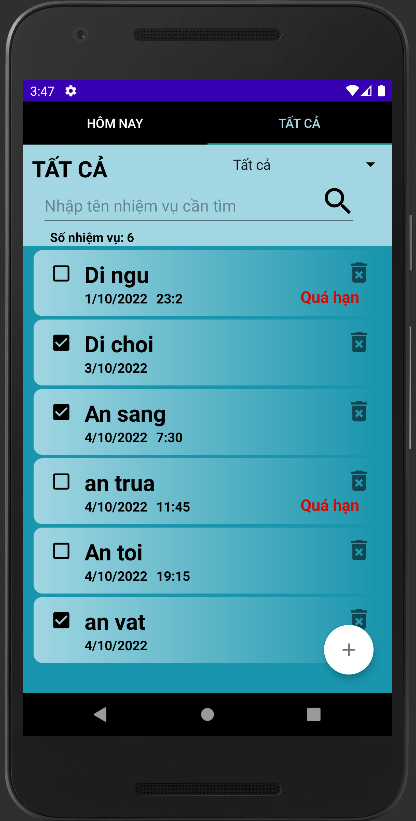
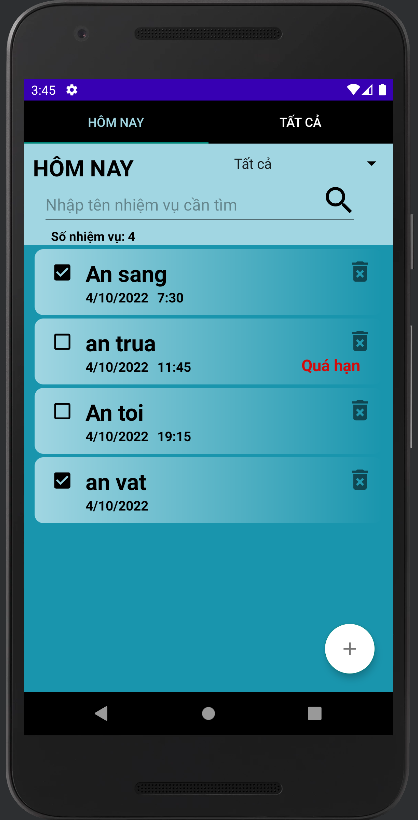
### 3.1.3 Bảng một số modul chính trong chương trình

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên modun** | **Lớp liên quan** | **Mô tả** |
| 1 | Activity | - MainActivity  - AddActivity  - UpdateActivity  - HandWriting | Hiển thị giao diện chính của chương trình: Hiển thị nhiệm vụ, cập nhập nhắc nhở, thêm nhắc nhở và viết tay |
| 2 | database | - SqliteHelper | Thực hiện quản lý dữ liệu các nhắc nhở và đưa vào database. |
| 3 | adapter | - AdapterNV  - AdapterViewPage  - ViewPageAdapter | Tùy chỉnh giao diện các item của listview. |
| 4 | fragment | * AllFragment   - DoodleFragment  - OCRFragment  - TodayFragment | Chia nhỏ các giao diện  Thuận lợi cho việc kiểm soát code |
| 5 | service | * AlamReceier   - SearchTask | Dịch vụ chạy ngầm để kiểm tra thời gian báo thức và tìm kiếm |

Bảng 3. 1 Một số modul chính trong chương trình

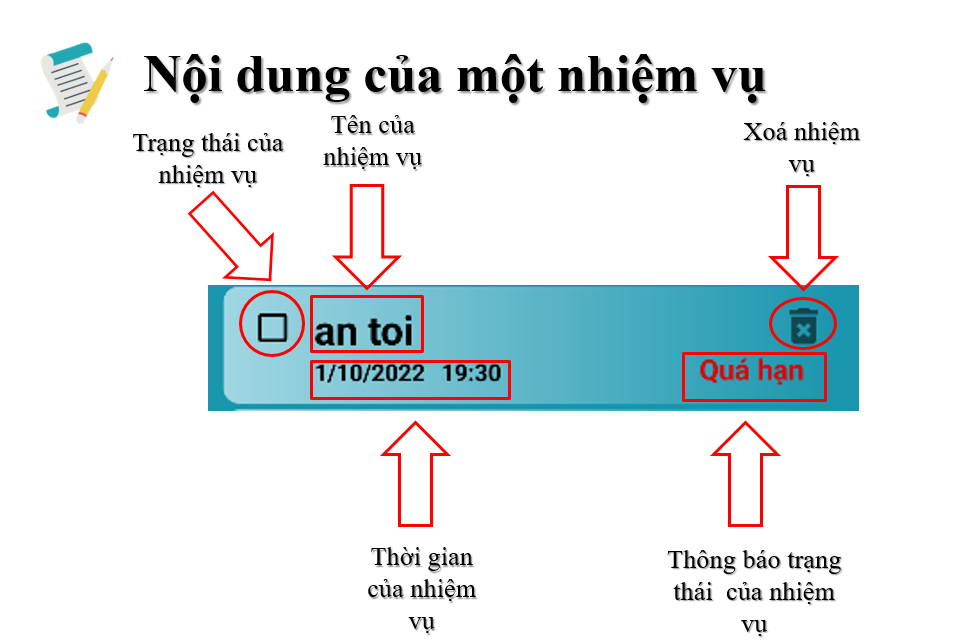
## 3.2 Thiết kế giao diện chương trình

### 3.2.1 Giao diện chính của chương trình



Hình 3. 2 Giao diện chính của chương trình

### 3.2.2 Nội dung của một nhiệm vụ



Hình 3. 3 Nội dung của nhiệm vụ

### 3.2.3 Thông báo của một nhiệm vụ

******

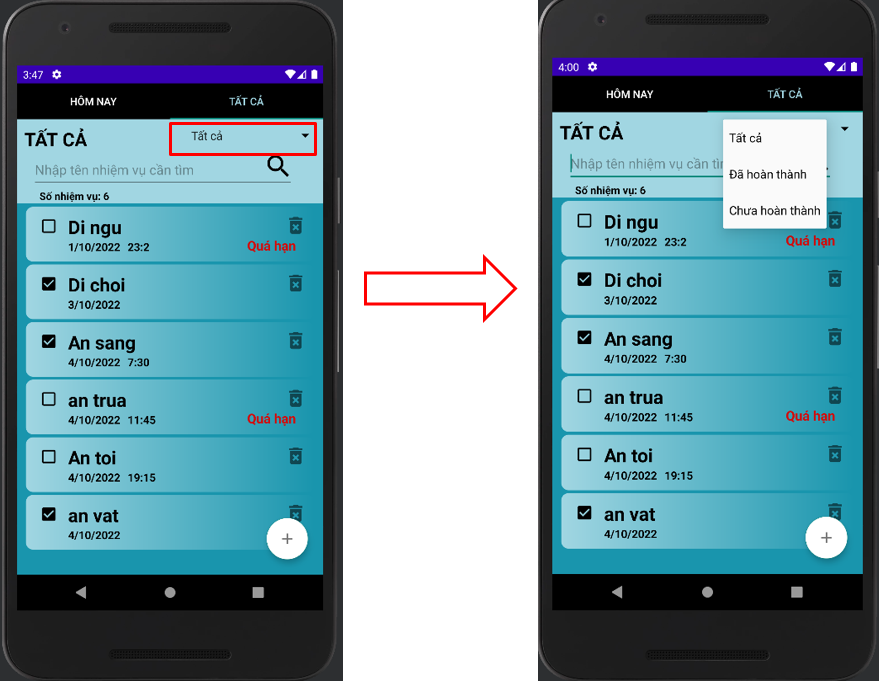
Hình 3. 4 Thông báo của nhiệm vụ

### 3.2.4 Tìm kiếm nhiệm vụ



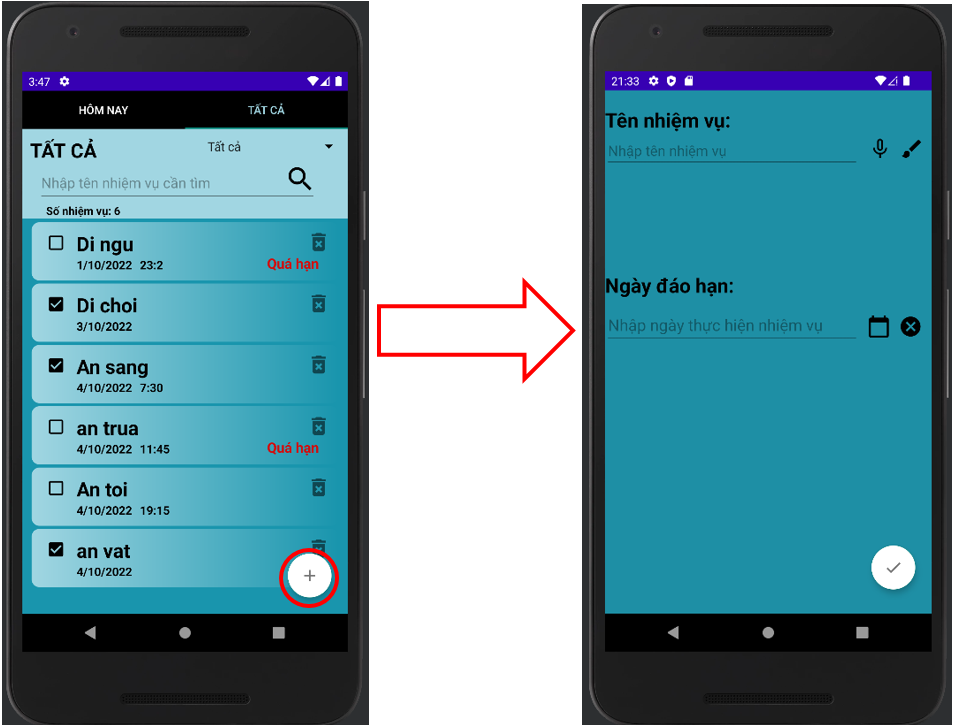
*Hình 3. 5 Thanh công cụ tìm kiếm*

### 3.2.5: Lọc các nhiệm vụ

**

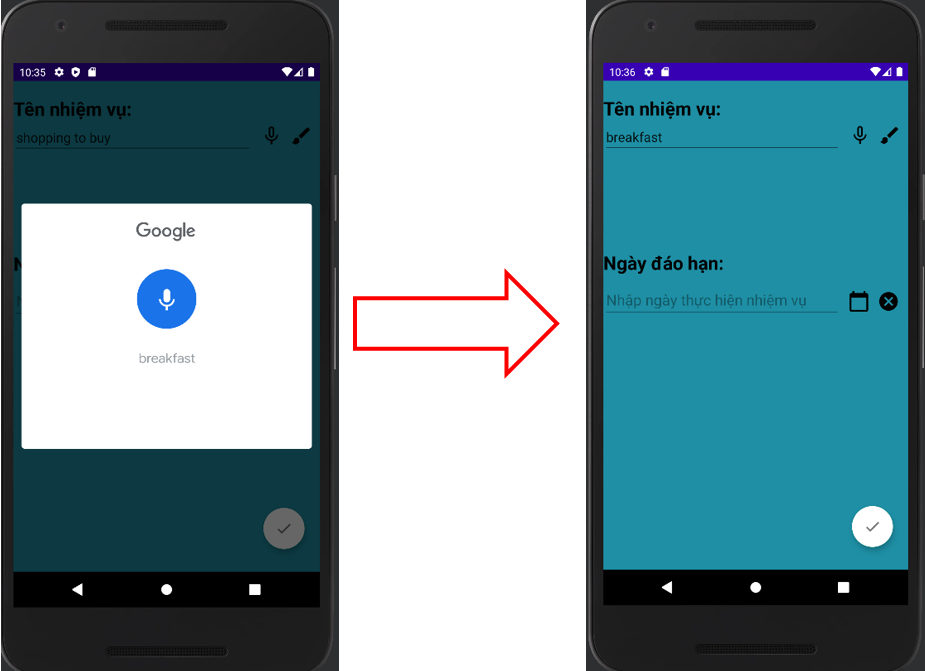
*Hình 3. 6 Lọc nhiệm vụ*

### 3.2.6: Thêm nhiệm vụ

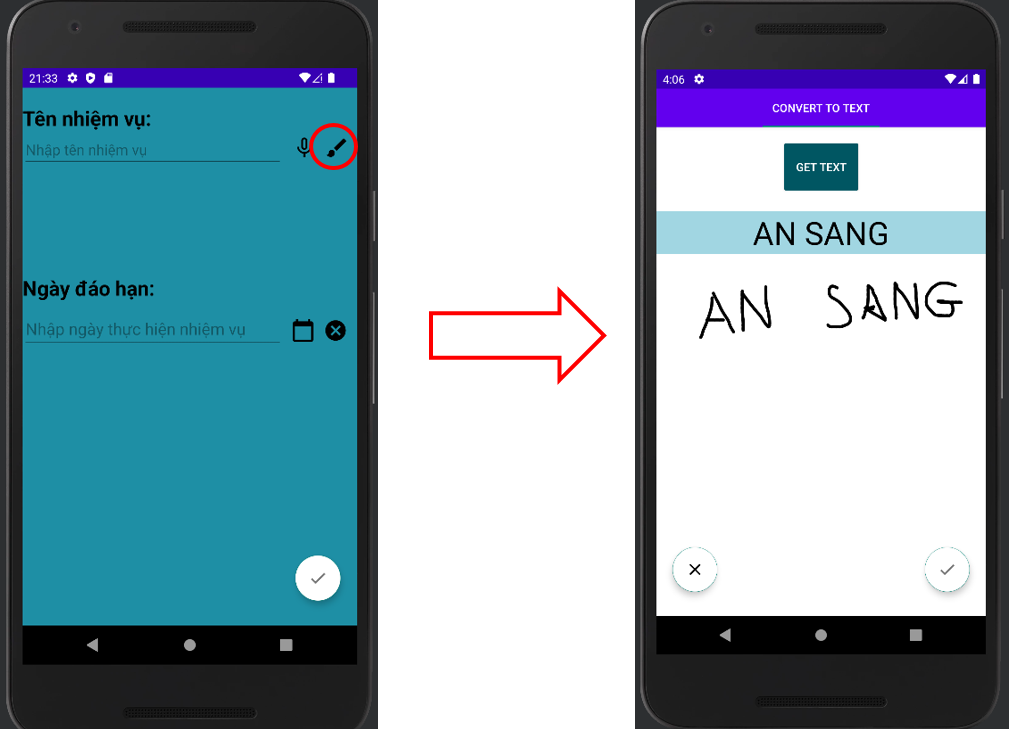
****

Hình 3. 7 Thêm nhiệm vụ

### 3.2.7: Đa dạng đầu vào

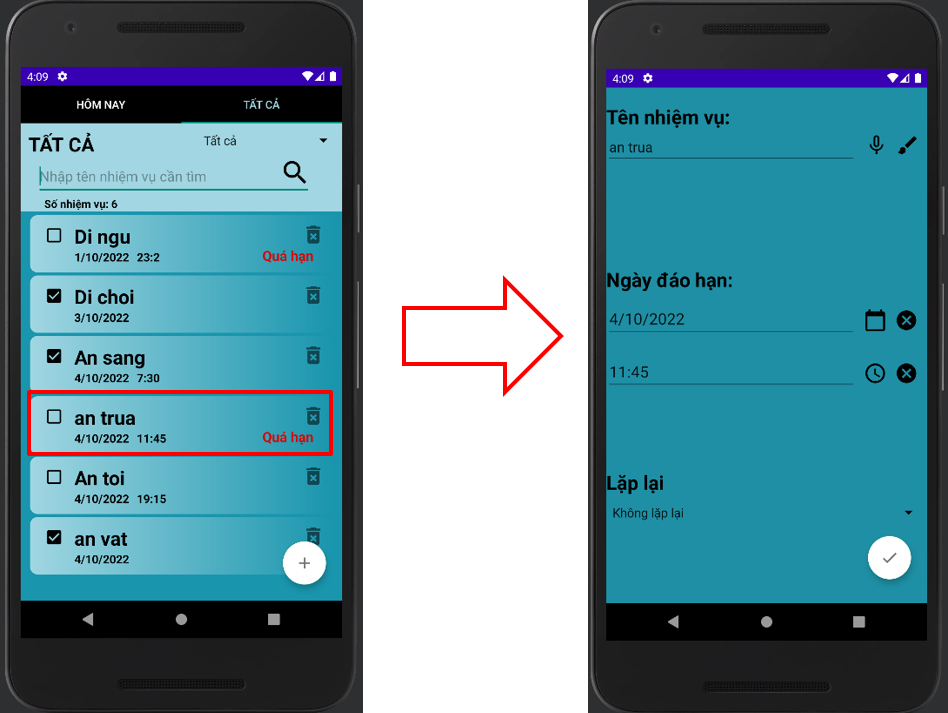
******

Hình 3. 8 Speech To Text



Hình 3. 9 Chữ viết tay

### 3.2.8: Sửa nhiệm vụ

******

*Hình 3. 10 Sửa nhiệm vụ*

# KẾT LUẬN

Sau hơn 3 tháng tìm hiểu nghiên cứu và thực hiện đề tài, đồ án tốt nghiệp đã được hoàn thành đúng thời hạn và đạt được những kết quả như sau :

* Nghiên cứu và tìm hiểu rõ hơn về nền tảng Android
* Hiểu sâu hơn về quản trị CSDL với SQliteHelper
* Nắm rõ được ngôn ngữ lập trình java

**Phần mềm đã hoàn thiện được các tính năng sau:**

* Hiển thị giao diện nhiệm vụ
* Giúp quản lý được thông tin các sự việc nhắc nhở
* Quản lý báo thức khi đến thời gian nhắc nhở

**Đề xuất hướng phát triển tiếp cho phần mềm hoàn thiện hơn**:

Để ứng dụng phổ biến hơn với người dùng thì cần phải phát trên nhiều nền tảng hơn như (Android, IOS, Web). Trong phạm vi về đề tài này, do khả năng hạn chế về thời gian cũng như kinh nghiệm nên phần mềm còn một số phần chưa hoàn chỉnh, đây sẽ là hướng phát triển để em tiếp tục phát triển trong tương lai gần :

* Hoàn thiện hơn các chức năng của hệ thống
* Phát triển quản lý dữ liệu nhắc nhở online
* Phát triển trên các hệ điều hành khác: IOS, windows phone,…

# DANH MỤC TÀI LIỆU THAM KHẢO

* Tài liệu Tiếng Việt

[1] Tài liệu lập trình Android tiếng việt

* Tài liệu Tiếng Anh

[2] Mike O'Docherty, *Object - Oriented Analysis & Design*, Wiley India Pvt, Limited, 2005.

[3] *Andy Ju An Wang Kai Quian*, Component - Oriented Programming, Wikey, 2005.

* Website tham khảo

[4] https://viblo.asia/p/su-dung-fragment-trong-android-4P856a2alY3

[5] https://www.tinhte.vn/categories/android.150/

[6] Khảo sát các hệ thống hiện tại: <https://play.google.com/store>

[7] <https://www.youtube.com/user/cellphoneuk>

[8] http://vi.wikipedia.org/wiki/Android, truy cập ngày 10/11/2014

[9] <http://vietandroid.com>

[10] http://www.developer.android.com

# PHỤ LỤC

## Phụ lục code:

<https://github.com/NguyenHung1911/Project/blob/main/ToDoApp.7z>