



**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO KHÓA LUẬN TỐT NGHIỆP**

**ĐỀ TÀI**

XÂY DỰNG ỨNG DỤNG QUẢN LÝ NHẮC NHỞ TIÊM PHÒNG CHO TRẺ NHỎ

**Chuyên ngành đào tạo: KỸ THUẬT PHẦN MỀM**

**Giảng viên hướng dẫn: TS. HUỲNH HỮU NGHĨA**

**Lớp: DHKTPM12A**

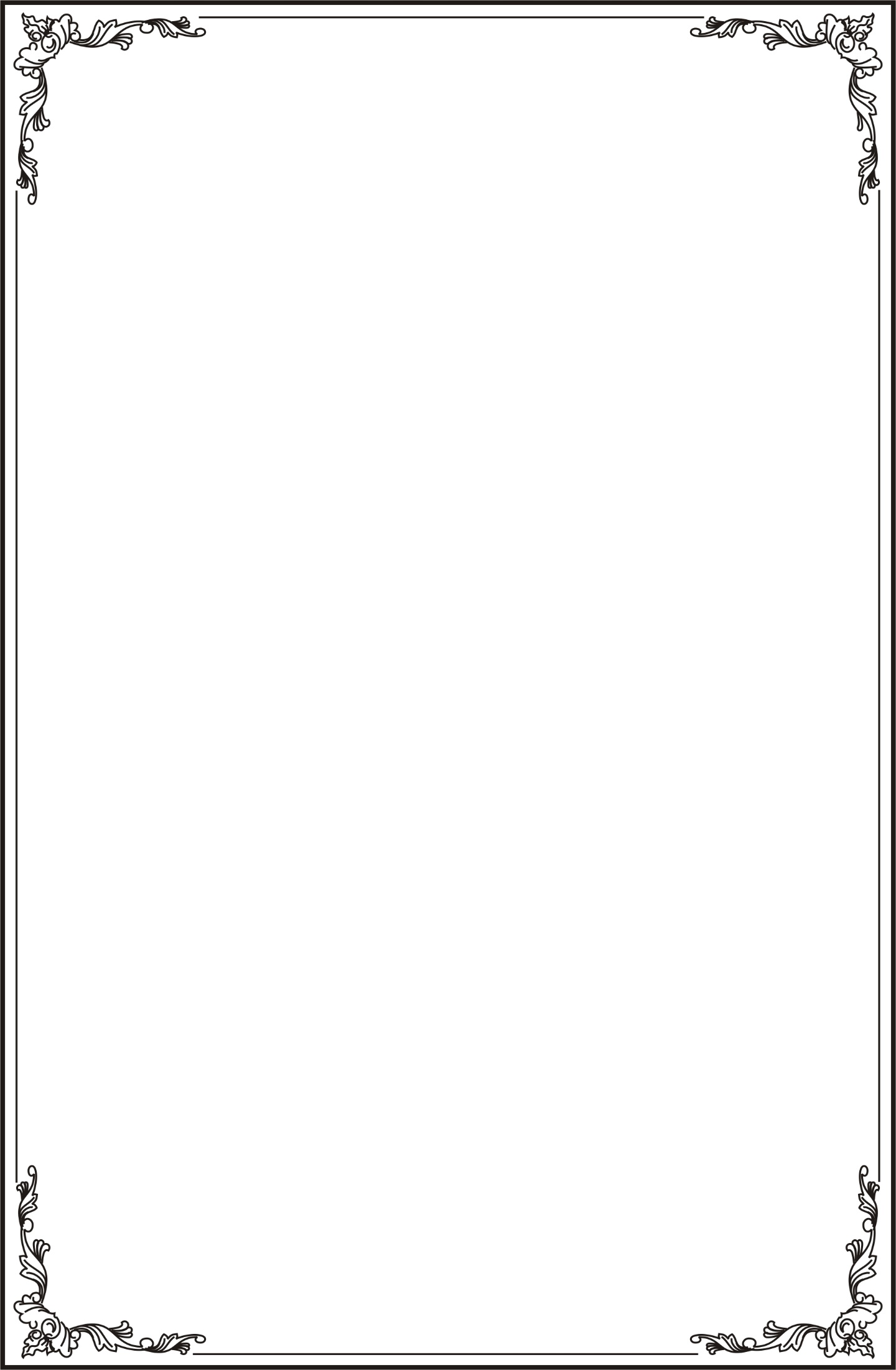
**Họ và tên sinh viên 1: NGUYỄN THANH TƯỜNG**

**Mã số sinh viên: 16028491**

**Họ và tên sinh viên 2: NGUYỄN NHẬT TRƯỜNG**

**Mã số sinh viên: 16026511**

TP.Hồ Chí Minh - Năm 2021

**INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**NGUYEN THANH TUONG**

**NGUYEN NHAT TRUONG**

**BUIDING THE APPLICATION MANAGEMENT REMINDERS FOR CHILDREN**

Major: Software engineer

**Supervisor: Dr. Huynh Huu Nghia**

HO CHI MINH CITY, 2021

**Title**: **BUIDING THE APPLICATION MANAGEMENT REMINDERS FOR CHILDERN**

**Abstract**:

1. **Reason for choosing topic:**

* The application will provide solutions to help manage vaccination time for babies more easily, supporting users to accurately inject according to the timeline of the vaccination regimen.
* The practical application is very good because of the increasing demand.

1. **Problems need to be slove:**

* Create vaccine datasets and injections (time binding).
* Manage on how the application's notification time.
* Learn to graph health statistics.
* Learn about regulations, issues in vaccination to create the most accurate chassis data.

1. **Methods of implementation:**

* Using Java language and SQLite database management system.
* Made on android platform 9.0
* Use the necessary libraries: Material (UI design), MPChart (Statistical chart design), Lottie Animations

1. **Result:**

* The application has implemented reminders and notification management to help users easily manage their time.
* Relative management function and CRUD to manage relatives.
* Employee management and CRUD functions to manage children's health.
* Preliminary processing of data sets. How to handle binding injections in the same vaccine.

1. **Conclude:**

* The group has performed the basic function of vaccination reminders on mobile devices and child management.
* Implement the basic management functions required for an application.
* Works well on phones from android 5 and up.
* The team understood how to make a real mobile application.

**Key word: vaccine (vacxin), relative (baby), health, injection, data processing**

**LỜI CẢM ƠN**

Thời gian 15 tuần làm khóa luận tốt nghiệp là cơ hội cho em tổng hợp và hệ thống hóa lại những kiến thức đã học. Từ đó em nhận thấy, việc cọ sát thực tế là vô cùng quan trọng giúp em xây dựng nền tảng lý thuyết được học ở trường vững chắc hơn.

Em xin gửi lời cảm ơn chân thành tới Trường Đại học Công Nghiệp Tp.HCM, Khoa Công Nghệ Thông Tin, các thầy, cô chuyên ngành Khoa Công nghệ thông tin đã chia sẽ những kiến thức quý báu để em có thêm tự tin làm hành trang hữu ích cho việc làm khóa luận tốt nghiệp.

Đặc biệt, em xin gởi đến thầy Huỳnh Hữu Nghĩa đã hướng dẫn tận tình, giúp đỡ em trong thời gian làm khóa luận tốt nghiệp với lời cảm ơn sâu sắc nhất.

Con xin chân thành cảm ơn gia đình đã luôn ở bên ủng hộ lúc con căng thẳng nhất.

Em xin cảm ơn bạn bè đã hỗ trợ em trong thời gian làm khóa luận tốt nghiệp, giúp em có thể ôn lại kiến thức và giải quyết cùng em những vấn đề khó khăn.

Em xin chân thành cảm ơn!

**Nguyễn Nhật Trường**

Tp. Hồ Chí Minh, ngày 25 tháng 5 năm 2021

**Sinh viên thực hiện**

**Nguyễn Thanh Tường**

**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN**……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….………………………………………………………………………………………. *Tp. Hồ Chí Minh, ngày… tháng … năm 2020*

**Giảng viên hướng dẫn**

**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN**……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….……………………………………………………………………………………….………………………………………………………………………………………. *Tp. Hồ Chí Minh, ngày… tháng … năm 2020*

**Giảng viên phản biện**

**Mục lục:**

[LỜI MỞ ĐẦU 14](#_Toc73394037)

[1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài 14](#_Toc73394038)

[2. Ý nghĩa khoa học và ý nghĩa thực tiễn của đề tài 15](#_Toc73394039)

[CHƯƠNG 1: GIỚI THIỆU 16](#_Toc73394040)

[1.1. Tổng quan 16](#_Toc73394041)

[1.2. Mục tiêu đề tài 17](#_Toc73394042)

[1.3. Phạm vi đề tài 17](#_Toc73394043)

[1.4. Mô tả yêu cầu chức năng 18](#_Toc73394044)

[1.4.1 Phân tích nghiệp vụ và thu nhận yêu cầu: 18](#_Toc73394045)

[1.4.1.1 Quản lý đối tượng tiêm chủng 19](#_Toc73394046)

[1.4.1.2 Quản lý thông báo tiêm chủng 20](#_Toc73394047)

[1.4.1.3 Đặt lại ứng dụng 21](#_Toc73394048)

[1.4.1.4 Xem phác đồ tổng quan 21](#_Toc73394049)

[1.4.1.5 Đánh giá ứng dụng 21](#_Toc73394050)

[1.4.1.6 Tra cứu kiến thức 21](#_Toc73394051)

[1.4.1.7 Xem quy định về tiêm chủng 21](#_Toc73394052)

[1.4.2 Yêu cầu phi chức năng 21](#_Toc73394053)

[1.4.3 Xác định loại ứng dụng 22](#_Toc73394054)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 23](#_Toc73394055)

[2.1 Lịch sử Android 23](#_Toc73394056)

[2.2 Kiến trúc của Android 24](#_Toc73394057)

[2.2.1. Tầng ứng dụng 25](#_Toc73394058)

[2.2.2. Application framework 25](#_Toc73394059)

[2.2.3. Library 26](#_Toc73394060)

[2.2.4. Android Runtime 26](#_Toc73394061)

[2.2.5. Linux kernel 27](#_Toc73394062)

[2.3 Android emulator 27](#_Toc73394063)

[2.4 Các thành phần trong một Android Project 28](#_Toc73394064)

[2.4.1. AndroidManifest.xml 28](#_Toc73394065)

[2.4.2. File R.java 29](#_Toc73394066)

[2.5 Chu kì ứng dụng Android 29](#_Toc73394067)

[2.5.1. Chu kỳ sống thành phần 30](#_Toc73394068)

[2.5.2. Các trạng thái của chu kỳ sống 30](#_Toc73394069)

[2.5.3. Chu kỳ sống của ứng dụng 31](#_Toc73394070)

[2.5.4. Các sự kiện trong chu kỳ sống của ứng dụng 31](#_Toc73394071)

[2.5.5. Các phương thức của chu kỳ sống 32](#_Toc73394072)

[2.6 Các thành phần giao diện trong Android 33](#_Toc73394073)

[2.6.1. View 33](#_Toc73394074)

[2.6.2. ViewGroup 34](#_Toc73394075)

[2.6.3. Button 37](#_Toc73394076)

[2.6.5. ImageView 38](#_Toc73394077)

[2.6.6. TextView 39](#_Toc73394078)

[2.6.9. ContextMenu 41](#_Toc73394079)

[2.6.10. Activity & Intend 41](#_Toc73394080)

[2.6.11. Expandable Listview 44](#_Toc73394081)

[2.7 Ngôn ngữ lập trình Java 45](#_Toc73394082)

[2.8 SQLite 46](#_Toc73394083)

[2.9 Service 47](#_Toc73394084)

[2.10 Broadcast Receivers 49](#_Toc73394085)

[2.11 Foreground services 49](#_Toc73394086)

[2.12 Alarm manager 50](#_Toc73394087)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 54](#_Toc73394088)

[3.1 Đặc tả yêu cầu 54](#_Toc73394089)

[3.2 Phân tích yêu cầu chức năng 54](#_Toc73394090)

[3.3 Các yêu cầu phi chức năng 55](#_Toc73394091)

[3.3.1 Mô hình Use case 56](#_Toc73394092)

[3.3.2 Danh sách tác nhân và mô tả 57](#_Toc73394093)

[3.3.3 Danh sách Use case và mô tả 57](#_Toc73394094)

[3.3.4 Đặc tả use case 61](#_Toc73394095)

[3.4.4.1 UC001 – Xem thông tin lưu ý 61](#_Toc73394096)

[3.4.4.2 UC002 – Reset ứng dụng 63](#_Toc73394097)

[3.4.4.3 UC003 – Cài đặt thông báo 65](#_Toc73394098)

[3.4.4.4 UC004 – Đánh giá ứng dụng 68](#_Toc73394099)

[3.4.4.5 UC005 – Chia sẻ ứng dụng 69](#_Toc73394100)

[3.4.4.6 UC006 – Tra cứu kiến thức tiêm chủng 71](#_Toc73394101)

[3.4.4.7 UC007 – Xem các câu hỏi thường gặp 73](#_Toc73394102)

[3.4.4.8 UC008 – Xem phác đồ tổng quan 75](#_Toc73394103)

[3.4.4.9 UC009 – Xem thông tin chi tiết vaccine 76](#_Toc73394104)

[3.4.4.10 UC010 – Quản lý thông tin trẻ 77](#_Toc73394105)

[3.4.4.11 UC011 – Thêm trẻ em 79](#_Toc73394106)

[3.4.4.12 UC012 – Xóa trẻ em 82](#_Toc73394107)

[3.4.4.13 UC013 – Cập nhật thông tin trẻ em 84](#_Toc73394108)

[3.4.4.14 UC014 – Xem lịch sử tiêm phòng 86](#_Toc73394109)

[3.4.4.15 UC015 – Cập nhật trạng thái phác đồ 87](#_Toc73394110)

[3.4.4.16 UC016 – Quản lý sức khỏe 90](#_Toc73394111)

[3.4.4.17 UC017 – Cập nhật sức khỏe 91](#_Toc73394112)

[3.4.4.18 UC018 – Xóa sức khỏe 94](#_Toc73394113)

[3.4.4.19 UC019 – Thêm sức khỏe 97](#_Toc73394114)

[3.4.4.20 UC020 – Xem thống kê sức khỏe 100](#_Toc73394115)

[3.4 Sơ đồ Class 102](#_Toc73394116)

[Danh sách các đối tượng 102](#_Toc73394117)

[Mô hình hóa các lớp đối tượng 102](#_Toc73394118)

[Mô hình dữ liệu quan hệ 104](#_Toc73394119)

[CHƯƠNG 4: HIỆN THỰC 105](#_Toc73394120)

[4.1. Giao diện Sổ Tiêm Chủng: 105](#_Toc73394121)

[4.2 Giao diện Danh sách vaccine 106](#_Toc73394122)

[4.3 Thông báo gửi đến 107](#_Toc73394123)

[4.4 Giao diện Cài Đặt 107](#_Toc73394124)

[4.5 Giao diện Thêm trẻ em 109](#_Toc73394125)

[4.6 Giao diện Phác đồ cá nhân của trẻ 110](#_Toc73394126)

[4.7 Giao diện Cài Đặt Thông Báo 111](#_Toc73394127)

[4.8 Giao diện Theo Dõi Sức Khỏe Tại Nhà 112](#_Toc73394128)

[CHƯƠNG 5: KẾT LUẬN 113](#_Toc73394129)

[5.1 Kết quả đạt được 113](#_Toc73394130)

[5.2 Hướng phát triển 113](#_Toc73394131)

[TÀI LIỆU THAM KHẢO 116](#_Toc73394132)

[PHỤ LỤC 118](#_Toc73394133)

[KẾ HOẠCH THỰC HIỆN 120](#_Toc73394134)

**Mục lục bảng:**

[Bảng 3. 1 Yêu cầu phi chức năng 42](#_Toc73379858)

[Bảng 3. 2 Danh sách các lớp đối tượng 88](#_Toc73379859)

[Bảng 3.3 1 Danh sách tác nhân và mô tả 44](#_Toc73379864)

[Bảng 3.3 2 Danh sách usecase và mô tả 48](#_Toc73379865)

[Bảng 3.4.4. 1 UC001 – Xem thông tin lưu ý 51](#_Toc73379870)

[Bảng 3.4.4. 2 UC002 – Reset ứng dụng 53](#_Toc73379871)

[Bảng 3.4.4. 3 UC003 – Cài đặt thông báo 55](#_Toc73379872)

[Bảng 3.4.4. 4 UC004 – Đánh giá ứng dụng 57](#_Toc73379873)

[Bảng 3.4.4. 5 UC005 – Chia sẻ ứng dụng 59](#_Toc73379874)

[Bảng 3.4.4. 6 UC006 – Tra cứu kiến thức tiêm chủng 61](#_Toc73379875)

[Bảng 3.4.4. 7 UC007 – Xem các câu hỏi thường gặp 62](#_Toc73379876)

[Bảng 3.4.4. 8 UC008 – Xem phác đồ tổng quan 64](#_Toc73379877)

[Bảng 3.4.4. 9 UC009 – Xem thông tin chi tiết vaccine 65](#_Toc73379878)

[Bảng 3.4.4. 10 UC010 – Quản lý thông tin trẻ 66](#_Toc73379879)

[Bảng 3.4.4. 11 UC011 – Thêm trẻ em 68](#_Toc73379880)

[Bảng 3.4.4. 12 UC012 – Xóa trẻ em 71](#_Toc73379881)

[Bảng 3.4.4. 13 UC013 – Cập nhật thông tin trẻ em 73](#_Toc73379882)

[Bảng 3.4.4. 14 UC014 – Xem lịch sử tiêm phòng 74](#_Toc73379883)

[Bảng 3.4.4. 15 UC015 – Cập nhật trạng thái phát đồ 76](#_Toc73379884)

[Bảng 3.4.4. 16 UC016 – Quản lý sức khỏe 78](#_Toc73379885)

[Bảng 3.4.4. 17 UC017 – Cập nhật sức khỏe 80](#_Toc73379886)

[Bảng 3.4.4. 18 UC018 – Xóa sức khỏe 83](#_Toc73379887)

[Bảng 3.4.4. 19 UC019 – Thêm sức khỏe 86](#_Toc73379888)

[Bảng 3.4.4. 20 UC20 – Xem thống kê sức khỏe 88](#_Toc73379889)

[Bảng 3. 1 Yêu cầu phi chức năng 44](#_Toc73380481)

[Bảng 3. 2 Danh sách các lớp đối tượng 90](#_Toc73380482)

[Bảng 3. 1 Yêu cầu phi chức năng 44](#_Toc73380499)

[Bảng 3. 2 Danh sách các lớp đối tượng 90](#_Toc73380500)

**Mục lục hình ảnh:**

[Hình 1.4.1 Sơ đồ phân rã chức năng của hệ thống thông báo nhắc nhở tiêm phòng 15](#_Toc73379910)

[Hình 1.4.1. 1 Phân rã chức năng quản lý đối tượng 17](#_Toc73379916)

[Hình 2. 1 Android timeline 21](#_Toc73379922)

[Hình 2. 2 Kiến trúc Android 22](#_Toc73379923)

[Hình 2. 3 Android emulator 24](#_Toc73379924)

[Hình 2. 4 Bố trí các widget sử dụng LinearLayout 31](#_Toc73379925)

[Hình 2. 5 Bố trí các widget trong FrameLayout 32](#_Toc73379926)

[Hình 2. 6 Bố trí widget trong RetaliveLayout 32](#_Toc73379927)

[Hình 2. 7 Bố trí widget trong TableLayout 33](#_Toc73379928)

[Hình 2. 8 Minh hoạ cho một ListView 35](#_Toc73379929)

[Hình 2. 9 Minh họa Radiobutton. 36](#_Toc73379930)

[Hình 2. 10 Minh hoạ context menu 37](#_Toc73379931)

[Hình 2. 11 Truyền dữ liệu giữa 2 Activity 38](#_Toc73379932)

[Hình 2. 12 Các thuộc tính của Intend 38](#_Toc73379933)

[Hình 2. 13 Các Action đã được định nghĩa sẵn trong Intend 39](#_Toc73379934)

[Hình 2. 14 Hình ảnh minh họa Expandable Listview 40](#_Toc73379935)

[Hình 2.5.2. 1 Chu kỳ sống của Activity 28](#_Toc73379936)

[Hình 2.5.4. 1 Các sự kiện 29](#_Toc73379943)

[Hình 2.5.4. 2 Cấu trúc một giao diện ứng dụng Android 31](#_Toc73379944)

[Hình 3. 1 Mô hình Use case 44](#_Toc73379951)

[Hình 3.4. 1 Activity Xem thông tin lưu 51](#_Toc73379958)

[Hình 3.4. 2 Actitity Reset ứng dụng 53](#_Toc73379959)

[Hình 3.4. 3 Activity Cài đặt thông báo 55](#_Toc73379960)

[Hình 3.4. 4 Activity Đánh giá ứng dụng 57](#_Toc73379961)

[Hình 3.4. 5 Activity Chia sẻ ứng dụng 58](#_Toc73379962)

[Hình 3.4. 6 Tra cứu kiến thức tiêm chủng 60](#_Toc73379963)

[Hình 3.4. 7 Activity Xem câu hỏi thường gặp 62](#_Toc73379964)

[Hình 3.4. 8 Xem phác đồ tổng quan 63](#_Toc73379965)

[Hình 3.4. 9 Activity Xem thông tin chi tiết vaccine 64](#_Toc73379966)

[Hình 3.4. 10 Quản lý thông tin trẻ 66](#_Toc73379967)

[Hình 3.4. 11 Hình 3- 19: Activity Thêm trẻ em 68](#_Toc73379968)

[Hình 3.4. 12 Hình 3- 20 Activity Xóa trẻ em 70](#_Toc73379969)

[Hình 3.4. 13 Activity Cập nhật thông tin trẻ em 72](#_Toc73379970)

[Hình 3.4. 14 Activity Xem lịch sử tiêm phòng 74](#_Toc73379971)

[Hình 3.4. 15 Activity Cập nhật trạng thái phát đồ 76](#_Toc73379972)

[Hình 3.4. 16 Activity xóa nhân viên 78](#_Toc73379973)

[Hình 3.4. 17 Cập nhật sức khỏe 80](#_Toc73379974)

[Hình 3.4. 18 Xóa sức khỏe 83](#_Toc73379975)

[Hình 3.4. 19 Thêm sức khỏe 86](#_Toc73379976)

[Hình 3.4. 20 Xem thống kê sức khỏe 88](#_Toc73379977)

[Hình 3.4. 21 Sơ đồ lớp đối tượng 90](#_Toc73379978)

[Hình 3.4. 22 Mô hình dữ liệu quan hệ 91](#_Toc73379979)

[Hình 4. 1 Sổ tiêm chủng 105](#_Toc73395161)

[Hình 4. 2 Danh sách vaccine 106](#_Toc73395162)

[Hình 4. 3 Thông báo gửi đến 107](#_Toc73395163)

[Hình 4. 4 Thông báo gửi đến 108](#_Toc73395164)

[Hình 4. 5 Màn hình cài đặt 114](#_Toc73395165)

[Hình 4. 6 Thêm trẻ em 115](#_Toc73395166)

[Hình 4. 7 Phác đồ điều trị cá nhân 115](#_Toc73395167)

[Hình 4. 8 Cài đặt thông báo 117](#_Toc73395168)

[Hình 4. 9 Theo dõi sức khỏe tại nhà 117](#_Toc73395169)

[Hình 5. 1 Hình ảnh mô tả Firebase Realtime Database 120](#_Toc73379988)

# LỜI MỞ ĐẦU

## Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài

Tiêm phòng vaccine (vacxin) cho trẻ là hình thức đứa vaccine vào cơ thể của trẻ em, giúp cơ thể trẻ em sinh ra kháng nguyên chống lại các loại bệnh tật nguy hiểm, việc này được xem là biện pháp phòng bệnh hiệu quả nhất để dùng để bảo vệ cho cơ thể người được tiêm không bị mắc các căn bệnh cũng như tránh xảy ra các vụ bùng phát bệnh dịch lớn, ảnh hưởng mạnh đến sức khỏe và tính mạng của con người nói riêng cũng như toàn nhân loại nói chung.

Chi phí tiêm phòng cũng thấp hơn rất nhiều so với chi phí chữa trị của các căn bệnh truyền nhiễm.

Vì vậy việc đi đến các cơ sơ y tế tiêm chủng đầy đủ , đúng lịch không chỉ làm giảm đi tỉ lệ mắc các bệnh và tử vong, bên cạnh đó trẻ em còn được phát triển toàn diện một cách khỏe mạnh cả về thể chất lẫn trí não

Với vòng xoay không ngừng của cuộc sống, nhịp sống của con người từ đó cũng bị cuốn theo một cách chóng mặt, đôi khi quý phụ huynh dễ quên đi lịch tiêm chủng của con em mình. Bên cạnh đó, các sổ ghi chép lịch tiêm phòng truyền thống cũng dể lãng quên ở đâu đó, dẫn đến mất sổ gây khó khăn cho đội ngũ y tế. Ngoài ra việc tiêm trể có thể dẫn đến nguy cơ mắc các bệnh nguy hiểm mặc dù đã được tiêm phòng.

Theo số liệu của tổ chức y tế thế giới WHO, cứ mỗi giây trôi qua lại có 4,2 trẻ được sinh ra đời, qua đó ta thấy được tổng số trẻ em mỗi ngày được sinh ra là rất lớn, nhu cầu được tiêm phòng từ đó mà được tăng lên một cách đáng kể, nhất là ở các nước có nền y học đang phát triển, việc kiểm soát lịch tiêm chủng là một điều hết sức khó khăn.

Chính vì những bất tiện đó, ứng dụng hỗ trợ quản lý tiêm phòng được ra đời, phù hợp hơn trong cuộc sống hiện đại ngày nay, giúp cho việc tiêm phòng được tiện lợi cũng như chính xác hơn, phòng trách được những rủi ro khổng lồ từ việc tiêm trễ cũng như không đủ liều lượng gây ra. Từ đó, nâng cao được sức khỏe cho thế hệ mầm non tương lai.

## Ý nghĩa khoa học và ý nghĩa thực tiễn của đề tài

Ứng dụng hỗ trợ quản lý tiêm phòng với nhiều ưu điểm như sau:

Quản lí tốt lịch tiêm chủng một cách chính xác tránh việc quên ngày, đến muộn hoặc đến sớm hơn lịch.

Nhắc nhở các trường hợp chưa tiêm đủ liều đủ mũi để được hoàn thành đúng thời hạn.

Thông báo đến các phụ huynh về tình trạng tiêm chủng của con em mình, giúp cho phụ huynh nắm được tình hình rõ hơn về việc tiêm vaccine của trẻ.

Giúp nâng cao tương tác giữa phụ huynh và đội ngũ y tế trong công cuộc tiêm phòng cho trẻ, giúp cho mọi thắc mắc cầ thiết của phụ huynh được giải đáp.

Vì sự thiếu hụt của các loại vaccine, ứng dụng còn giúp thông báo số lượng vaccine còn hay hết để tránh mất thời gian vào trường hợp đến cơ sở y tế nhưng lại không có vaccine

Từ những lợi ích của ứng dụng mang lại thông các ưu điểm được nêu trên, đây quả thực là một ứng dụng mang tính áp dụng thực tiễn rất cao, mở tra một bước ngoặc cho việc quản lí tiêm phòng của Việt Nam nói riêng cũng như toàn thế giới nói chung, một bước tiến mới trong việc kết hợp giữa công nghệ thông tin vào nền y học nước nhà, qua đó cải thiện được sức khỏe cho con người ở hiện tại và cả tương lai, góp phần trong công cuộc công nghiệp hóa- hiện đại hóa đất nước.

# CHƯƠNG 1: GIỚI THIỆU

## Tổng quan

Ngày nay các ứng dụng công nghệ thông tin được xem là một trong các yếu tố mang tính quyết định, tác động đến các hoạt động của các  tổ chức, công ty cũng như đời sống người dân, nó đóng vai trò hết sức quan trọng, ảnh hưởng to lớn đến đời sống con người, là bàn đạp để giúp cải thiện cuộc sống, giúp xã hội ngày càng được nâng cao, phát triển và hoàn thiện hơn .

Hiện nay nhu cầu của người dùng ngày càng tăng lên, vì vậy ngành công nghệ thông tin cũng ngày càng phát triển mạnh mẽ, chiếm một vị trí quan trọng không thể thiếu trong cuộc sống.

Với sự giúp đỡ của công nghệ thông tin và đặc biệt là sự phát triển của các thiết bị di động cũng như là các ứng dụng trên thiết bị di động, chúng ta đã thuận tiện và nhanh chóng hơn rất nhiều trong công việc, nó giúp ta giảm chi phí hơn, tiết kiệm thời gian hơn so với trước khi công nghệ chưa phát triển. Nó đã thúc đẩy sự phát triển và góp phần làm biến đổi bộ mặt của thế giới, cải thiện lại cuộc sống của con người.

Hiện nay trong lĩnh vực y tế cũng đã áp dụng công nghệ thông tin vào trong ngành với những thiết bị phần mềm, ứng dụng liên quan đến y tế , trong đó có việc quản lý tiêm phòng. Nó giúp cho người dùng cũng như là y bác sĩ có thể thuận lợi trong việc khám, chữa trị và tư vấn.

Bộ y tế cũng đã khuyến cáo với người dân là nên đưa trẻ tiêm chủng đầy đủ, đó là cách đơn giản và hiệu quả nhất trong việc trẻ em không mắc các bênh như sởi, rubella và ho gà. Khuyến cáo các bậc cha mẹ nên kiểm tra sổ của con em mình để đảm bảo rằng các mũi tiêm của trẻ được cập nhật một cách đầy đủ và tuân thủ chặt lẽ việc tiêm chủng quốc gia mà không có bất kì trì hoãn nào

Trong việc quản lý tiêm phòng khi chưa có áp dụng công nghệ thông tin thì bác sĩ và người tiêm phòng phải quản lý bằng cách truyền thống thủ công đó là ghi chép vào trong cuốn sổ tiêm chủng. Việc quản lý bằng cách truyền thống đó có rất nhiều hạn chế như: mất sổ trong lúc dọn dẹp, theo thời gian cuốn sổ ghi chép cũng sẽ hư hỏng hoặc rách do thời tiết cũng như là do con người dẫn đến sẽ không theo đúng tiến độ của lịch tiêm quốc gia.

Vì vậy, nhóm em xây dựng đề tài “ỨNG DỤNG QUẢN LÝ TIÊM PHÒNG CHO TRẺ EM ”. Thông qua ứng dụng, người dùng có thể dễ dàng quản lý thông tin cũng như là thời gian tiêm phòng cho các bé.

## Mục tiêu đề tài

Tạo được bộ dữ liệu về thông tin vaccine và các mũi tiêm của vaccine.

Xử lý ràng buộc về dữ liệu của các mũi tiêm. Ví dụ như: khoảng cách thời gian giữa 2 mũi tiêm, thời gian thích hợp để tiêm vaccine nào, kiểm tra trạng thái mũi tiêm trước, cập nhật lại lịch các mũi tiêm sau khi hoàn tất mũi tiêm trước.

Tạo bộ dữ liệu về thông tin quy định của tiêm chủng, những điều cần lưu ý trong vấn đề tiêm chủng để hỗ trợ thêm kiến thức người dùng.

Quản lý dữ liệu của trẻ sơ sinh.

Tạo phác đồ cá nhân phù hợp dựa vào ngày sinh của trẻ sơ sinh.

Nhắc nhở chính xác ngày tiêm phòng của trẻ: tiêm mũi nào của vaccine gì,....

🡪Mục tiêm chính của đề tài là giải quyết được vấn đề ràng buộc dữ liệu giữa thông báo, tiêm phòng thông qua cách kiểm tra ngày

## Phạm vi đề tài

Dữ liệu mà ứng dụng sử dụng được giới hạn trong độ tuổi từ 0-3.

## Mô tả yêu cầu chức năng

### Phân tích nghiệp vụ và thu nhận yêu cầu:

Sau khi xác định thực trạng và đặt ra các mục tiêu, bắt đầu tiến hành khảo sát thực tế và phân tích nghiệp vụ. Các yêu cầu, thông tin dữ liệu thực tế thu thập được thông qua khảo sát tại VNVC (là tên viết tắt của công ty cổ phần vaccine Việt Nam - VietNam Vaccine JSC) Lê Đại Hành (Lầu 1, Trung tâm thương mại Flemington, Cao ốc Bảo Gia, 184 Lê Đại Hành, P.15, Q.11, TP.HCM). Từ việc trao đổi thông tin với nhân viên tư vấn kết hợp cùng nguồn thông tin từ các văn bản, trang thông tin tiêm chủng quốc gia và hàng loại bài báo trên internet, đã hình thành sơ bộ được nguồn dữ liệu và các chức năng cần thiết của ứng dụng.

Từ những kết quả thu được, sơ đồ phân rã chức năng của ứng dụng được thể hiện như sau:

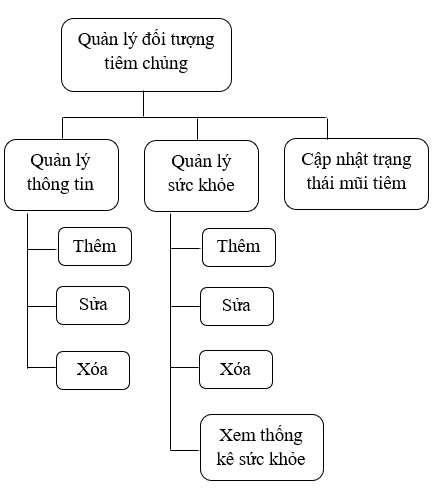


Hình 1.4. Sơ đồ phân rã chức năng của hệ thống thông báo nhắc nhở tiêm phòng

Qua sơ đồ, thấy rằng bài toán thông báo nhắc nhở tiên chủng có thể chia nhỏ thành 5 bài toán chính gồm: Quản lý đối tượng tiêm chủng, đánh giá ứng dụng, xem phác đồ tổng quan, quản lý thông báo tiêm chủng và xem quy định tiêm chủng.

#### Quản lý đối tượng tiêm chủng

Đối tượng tiêm chủng ở đây chính là người thực hiện tiêm chủng – trẻ sơ sinh. Riêng đối với quản lý đối tượng tiêm chủng sẽ được phân nhỏ thành: quản lý thông tin hồ sơ, quản lý sức khỏe, xem lịch sử và cập nhật trạng mũi tiêm.



Hình 1.4.1.  Phân rã chức năng quản lý đối tượng

Quản lý thông tin hồ sơ của trẻ gồm: họ tên, tên ở nhà, ngày sinh, giới tính. Với các chức năng thêm, sửa, xóa. Khi phát sinh (thêm) một trẻ em, phác đồ tiêm chủng cá nhân sẽ tự động phát sinh dựa vào ngày sinh của trẻ.

Quản lý sức khỏe thực chất là việc lưu lại tình trạng sức khỏe của đối tượng qua từng giai đoạn khác nhau, ví dụ như theo dõi về cân nặng, chiều cao. Các thông tin này là cơ sở để xác định, dự đoán tình trạng sức khỏe của trẻ và thể hiện thông qua biểu đồ thống kê (thống kê qua các tuần).

Xem lịch sử và cập nhật trạng thái mũi tiêm giúp người dùng quản lý trạng thái các mũi tiêm. Việc này hỗ trợ người dùng theo dõi phác đồ tiêm phòng của từng trẻ, biết được chính xác các mũi đã được tiêm, sắp tiêm và đã bỏ lỡ của trẻ nhỏ. Khi cập nhật mũi tiêm, hệ thống tự động kiểm tra ràng buộc dữ liệu giữa mũi tiêm hiện tại và mũi tiêm trước đó. Nếu khoảng cách giữa 2 mũi tiêm không đủ số tháng quy định hay trạng thái mũi tiêm trước chưa hoàn thành hệ thống hiển thị thông báo lỗi cho người sử dụng. Điều này giúp người dùng dễ dàng rà soát lại mũi tiêm trước đó đã toàn tất chưa, và khoảng cách thời gian giữa 2 mũi tiêm đã đủ số ngày quy định không. Sau khi cập nhật thành công mũi tiêm, chuyển thông tin mũi tiêm vừa cập nhật thành công sang danh sách tương ứng với trạng thái của mũi, cập nhật lại phác đồ tiêm của trẻ.

Xem lịch sử giúp đỡ người dùng dễ dàng thao tác, quản lý và xem các mũi tiêm của trẻ. Biết được mũi tiêm mà trẻ đã thực hiện. Những mũi tiêm mà trẻ đã bỏ lỡ trong thời gian qua. Và sắp tới trẻ sẽ chuẩn bị tiêm những mũi gì.

#### Quản lý thông báo tiêm chủng

Quản lý thông báo tiêm chủng được chia nhỏ ra thành 2 bài toán: cài đặt thông báo và gửi (nhắc nhở) thông báo đến điện thoại.

Cài đặt thông báo giúp người dùng tự mình quản lý thời gian ứng dụng thông báo trong ngày. Thông báo lúc mấy giờ? Thông báo đúng ngày hay nhắc nhở trước vài ngày. Bật/tắt trạng thái của thông báo từ ứng dụng.

Gửi/nhắc nhở thông báo đến điện thoại nhằm mục đích cho biết được thông tin về mũi tiêm được thực hiện vào lúc nào. Thông báo hỗ trợ người dùng tiêm chủng cho trẻ theo đúng lịch mà phác đồ đề ra. Hệ thống tự động xử lý bằng cách kiếm tra ràng buộc thời gian giữa ngày hiện hành, ngày tiêm, khoảng thời gian thông báo. Nếu

#### Đặt lại ứng dụng

Chức năng giúp người dùng khởi động ứng dụng về lúc ban đầu nhằm khắc phục và xử lý lỗi. Lúc này toàn bộ dữ liệu tồn tại trước đó sẽ bị xóa khỏi cơ sở dữ liệu.

#### Xem phác đồ tổng quan

Hỗ trợ người dùng biết được phác đồ tiêm chủng chung dành cho mọi trẻ em từ khi mới sinh đến 3 tuổi. Phác đồ gồm các mũi tiêm của nhiều vaccine. Khi người dùng chọn một mũi tiêm bất kỳ trong phát đồ, hệ thống hiển thị thông tin chi tiết vaccine tương ứng: tên vaccine, phòng bệnh, tác dụng phụ sau khi tiêm, thông tin các mũi tiêm và lịch tiêm và bộ y tế đã đưa ra,... Giúp người dùng có thêm kiến thức cần thiết về vaccine.

#### Đánh giá ứng dụng

Chức năng này để người dùng nêu lên quan điểm của bản thân sau khi sử dụng ứng dụng, đóng góp ý kiến giúp ứng dụng ngày càng được hoàn thiện hơn

#### Tra cứu kiến thức

Tra cứu kiến thức hỗ trợ người dùng có các thông tin, dữ liệu cần thiết trong tiêm chủng. Bản thân có thể giải quyết những vấn đề gặp phải trong thực tiễn.

#### Xem quy định về tiêm chủng

Giúp người dùng tránh được những rủi ro gặp phải trong tiêm chủng mở rộng.

### Yêu cầu phi chức năng

Ràng buộc dữ liệu chặt chẽ. Bộ dữ liệu khởi tạo gồm các mũi tiêm trong cùng một vaccine, phải thỏa yêu cầu 2 mũi tiêm liên tiếp phải cách nhau trong thời gian quy định của bộ y tế. Ví dụ vaccine viêm gan B có 4 mũi, trong đó mũi 1 và mũi 2 phải cách nhau từ 28~31 ngày (1 tháng – dùng tháng làm đơn vị tính thời gian chuẩn trong ứng dụng) –là khoảng thời gian hợp lý giữa 2 mũi trong tiêm phòng.

Có thể tiêm trễ hơn nhưng KHÔNG được tiêm sớm hạn.

Dữ liệu phát sinh (phác đồ cá nhân): phát sinh ngày tiêm được tính gần chính xác dựa trên ngày sinh của bé.

Ràng buộc sự kiện, các phím trên điện thoại di động.

Giao diện dễ nhìn, có các icon minh họa trực quan, sinh động

Thời gian trả về kết quả từ 2~5s

### Xác định loại ứng dụng

Ứng dụng chạy trên nền tảng android.

Ngôn ngữ lập trình ứng dụng là java.

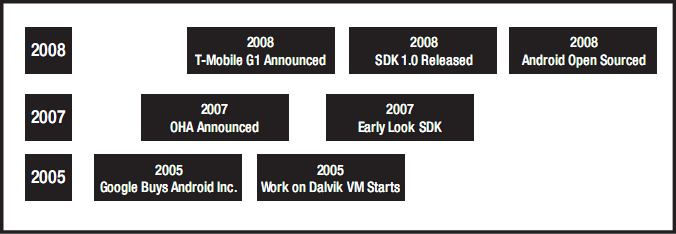
Ứng dụng dùng hệ quản trị cơ sở dữ liệu SQLite

Loại ứng dụng là : Mobile application

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## Lịch sử Android

Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên lõi Linux do công ty Android Inc. (California, Mỹ) thiết kế. Công ty này sau đó được Google mua lại vào năm 2005 và bắt đầu xây dựng Android Platform. Các thành viên chủ chốt tại ở Android Inc. gồm có: Andy Rubin, Rich Miner, Nick Sears, and Chris White.



Hình 2.  Android timeline

Và sau tiếp, vào cuối năm 2007, thuộc về Liên minh Thiết bị Cầm tay Mã Nguồn mở (Open Handset Alliance) gồm các thành viên nổi bật trong ngành viễn thông và thiết bị cầm tay như:

Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, and Vodafone Group,…

Mục tiêu của Liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android.

Phiên bản SDK lần đầu tiên phát hành vào tháng 11 năm 2007, hãng T-Mobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảng Android. Một vài ngày sau đó, Google lại tiếp tục công bố sự ra mắt phiên bản Android SDK release Candidate 1.0. Trong tháng 10 năm 2008, Google được cấp giấy phép mã nguồn mở cho Android Platform.

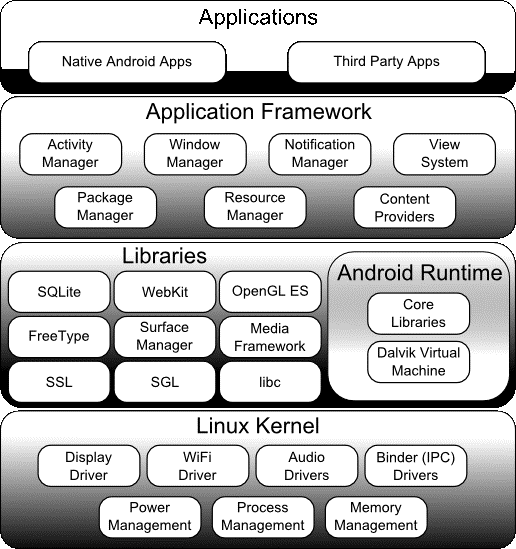
Khi Android được phát hành thì một trong số các mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ những ứng dụng khác. Việc tái sử dụng không chỉ được áp dụng cho các dịch vụ mà nó còn được áp dụng cho cả các thành phần dữ liệu và giao diện người dùng.

Vào cuối năm 2008, Google cho phát hành một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được các ứng dụng Android mà không bị ràng buộc vào các nhà cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên vản vá lỗi 1.1 của hệ điều hành này. Ở cả hai phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sử dụng bàn phím vật lý. Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng Tư năm 2009, cùng với một số tính năng khác. Chẳng hạn như nâng cao khả năng ghi âm truyền thông, vật dụng, và các live folder.[1]

## Kiến trúc của Android

Mô hình sau thể hiện một cách tổng quát các thành phần của hệ điều hành

Android. Mỗi một phần sẽ được đặc tả một cách chi tiết dưới đây.



Hình 2. Kiến trúc Android

### 2.2.1. Tầng ứng dụng

Android được tích hợp sẵn một số ứng dụng cần thiết cơ bản như: contacts, browser, camera, Phone. Tất cả các ứng dụng chạy trên hệ điều hành Android đều được viết bằng Java.[1]

### 2.2.2. Application framework

Tầng này Google xây dựng cho các nhà phát triển xây dựng ứng dụng của họ trên Android, bằng cách gọi các API có sẵn mà Google đã viết để sử dụng các tính năng của phần cứng mà không cần hiểu cấu trúc bên dưới.[10]

### 2.2.3. Library

Android bao gồm một tập hợp các thư viên C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Một số các thư viện cơ bản được liệt kê dưới đây:

* System C library: a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices.
* Media Libraries - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
* Surface Manager – Quản lý việc truy xuất vào hệ thống hiển thị
* LibWebCore - a modern web browser engine which powers both the Android browser and an embeddable web view.
* SGL - the underlying 2D graphics engine.
* 3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer.
* FreeType - bitmap and vector font rendering.
* SQLite - a powerful and lightweight relational database engine available to all applications.[2]

### 2.2.4. Android Runtime

Mỗi ứng dụng Android chạy trên một tiến trình riêng của máy ảo Dalvik(Vitural machine). Dalvik được viết để chạy nhiều máy ảo cùng lúc một các hiệu quả trên thiết bị di động.[1]

### 2.2.5. Linux kernel

Đây là nhân của hệ điều hành Android, mọi xử lý hệ thống đều phải qua tầng này. Linux Kernel cung cấp các trình điều khiển thiết bị phần cứng như: camera, USB, bluetooth. Kernel hoạt động như một lớp trừu tượng giữa phần cứng và phần mềm còn lại của hệ thống.[1]

## Android emulator

Android SDK và Plugin Eclipse được gọi là một Android Deverloper Tool (ADT). Các Android coder sẽ cần phải sử dụng công cụ IDE (Integrated Development Enveronment) này để phát triển, debugging và testing cho ứng dụng. Tuy nhiên, các coder cũng có thể không cần phải sử dụng IDE mà thay vào đó là sử dụng command line để biên dịch và tất nhiên là vẫn có Emulator như thường.

Android Emulator được trang bị đầy đủ hầu hết các tính năng của một thiết bị thật. Tuy nhiên, một số đã bị giới hạn như là kết nối qua cổng USB, camera và video, nghe phone, nguồn điện giả lập và bluetooth.

Android Emulator thực hiện các công việc thông qua một bộ xử lý mã nguồn mở, công nghệ này được gọi là QEMU [(http://bellard.org/qemu/](http://bellard.org/qemu/)) được phát triển bởi Fabrice Bellard. [2]



Hình 2. Android emulator

## Các thành phần trong một Android Project

### 2.4.1. AndroidManifest.xml

Trong bất kì một project Android nào khi tạo ra đều có một file AndroidManifest.xml, file này được dùng để định nghĩa các screen sử dụng, các permission cũng như các theme cho ứng dụng. Đồng thời nó cũng chứa thông tin về phiên bản SDK cũng như main activity sẽ chạy đầu tiên.

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

* Application :

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

+ ndroid:icon = “drawable resource” : Ở đây đặt đường dẫn đến file icon của ứng dụng khi cài đặt. VD: android:icon = [“@drawabl](about:blank)e/icon”.

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

+android:theme=“drawabletheme”: thuộc tính này để đặt theme cho ứng dụng. Các theme là các cách để hiển thị giao diện ứng dụng.

Ngoài ra còn nhiều thuộc tính khác…[8]

* Permission:

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

* **S**DK version

Thẻ xác định phiên bản SDK được khai báo như sau:

<uses-sdk android:minSdkVersion="7" />.

Ở đây chỉ ra phiên bản SDK nhỏ nhất mà ứng dụng hiện đang sử dụng.[9]

### 2.4.2. File R.java

File R.java là một file tự động sinh ra ngay khi tạo ứng dụng, file này được sử dụng để quản lý các thuộc tính được khai báo trong file XML của ứng dụng và các tài nguyên hình ảnh.

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

Có thể nói file R.java hoàn toàn không cần phải đụng chạm gì đến trong cả quá trình xây dựng ứng dụng.

## Chu kì ứng dụng Android

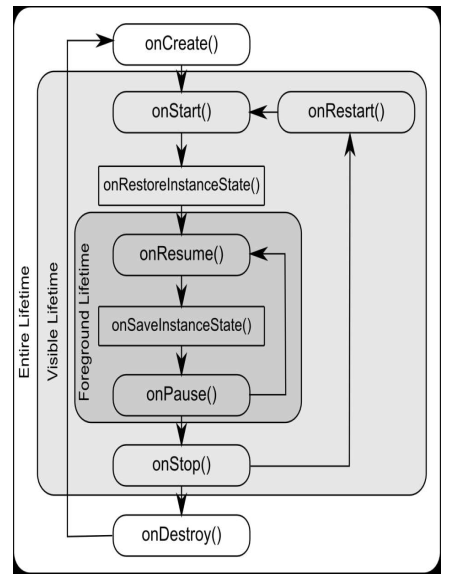
Một tiến trình Linux gói gọn một ứng dụng Android đã được tạo ra cho ứng dụng khi codes cần được run và sẽ còn chạy cho đến khi:

* Nó không phụ thuộc.
* Hệ thống cần lấy lại bộ nhớ mà nó chiếm giữ cho các ứng dụng khác
* Một sự khác thường và đặc tính cơ bản của Android là thời gian sống của tiến trình ứng dụng không được điều khiển trực tiếp bới chính nó. Thay vào đó, nó được xác định bởi hệ thống qua một kết hợp của:
* Những phần của ứng dụng mà hệ thống biết đang chạy
* Những phần quan trọng như thế nào đối với người dùng
* Bao nhiêu vùng nhớ chiếm lĩnh trong hệ thống.[2]

### 2.5.1. Chu kỳ sống thành phần

Các thành phần ứng dụng có một chu kỳ sống, tức là mỗi thành phần từ lúc bắt đầu khởi tạo và đến thời điểm kết thúc. Giữa đó, đôi lúc chúng có thể là active hoặc inactive, hoặc là trong trường hợp activies nó có thể visible hoặc invisible

### 2.5.2. Các trạng thái của chu kỳ sống



Hình 2.5.2. Chu kỳ sống của Activity

Một Activity chủ yếu có 3 chu kỳ chính sau:

* Active hoặc running: Khi Activity là được chạy trên màn hình. Activity này tập trung vào những thao tác của người dùng trên ứng dụng.
* Paused: Activity là được tạm dừng (paused) khi mất focus nhưng người dùng vẫn trông thấy. Có nghĩa là một Activity mới ở trên nó nhưng không bao phủ đầy màn hình. Một Activity tạm dừng là còn sống nhưng có thể bị kết thúc bởi hệ thống trong trường hợp thiếu vùng nhớ.
* Stopped: Nếu nó hoàn toàn bao phủ bởi Activity khác. Nó vẫn còn trạng thái và thông tin thành viên trong nó. Người dùng không thấy nó và thường bị loại bỏ trong trường hợp hệ thống cần vùng nhớ cho tác vụ khác.[2]

### 2.5.3. Chu kỳ sống của ứng dụng

Trong một ứng dụng Android có chứa nhiều thành phần và mỗi thành phần đều có một chu trình sống riêng. Và ứng dụng chỉ được gọi là kết thúc khi tất cả các thành phần trong ứng dụng kết thúc. Activity là một thành phần cho phép người dùng giao tiếp với ứng dụng. Tuy nhiên, khi tất cả các Activity kết thúc và người dùng không còn giao tiếp được với ứng dụng nữa nhưng không có nghĩa là ứng dụng đã kết thúc. Bởi vì ngoài Activity là thành phần có khả năng tương tác người dùng thì còn có các thành phần không có khả năng tương tác với người dùng như là Service, Broadcastreceiver. Có nghĩa là những thành phần không tương tác người dùng có thể chạy background dưới sự giám sát của hệ điều hành cho đến khi người dùng tự tắt chúng.[1]

### 2.5.4. Các sự kiện trong chu kỳ sống của ứng dụng

Nếu một Activity được tạm dừng hoặc dừng hẳn, hệ thống có thể bỏ thông tin khác của nó từ vùng nhớ bởi việc finish() (gọi hàm finish() của nó), hoặc đơn giản giết tiến trình của nó. Khi nó được hiển thị lần nữa với người dùng, nó phải được hoàn toàn restart và phục hồi lại trạng thái trước. Khi một Activity chuyển qua chuyển lại giữa các trạng thái, nó phải báo việc chuyển của nó bằng việc gọi hàm transition.



Hình 2.5.4. Các sự kiện

Tất cả các phương thức là những móc nối mà bạn có thể override để làm tương thích công việc trong ứng dụng khi thay đổi trạng thái. Tất cả các Activity bắt buộc phải có onCreate() để khởi tạo ứng dụng. Nhiều Activity sẽ cũng hiện thực onPause() để xác nhận việc thay đổi dữ liệu và mặt khác chuẩn bị dừng hoạt động với người dùng.[1]

### 2.5.5. Các phương thức của chu kỳ sống

Phương thức:onCreate()

* Được gọi khi activity lần đầu tiên được tạo
* Ở đây bạn làm tất cả các cài đặt tĩnh -- tạo các view, kết nối dữ liệu đế list
* Phương thức này gửi qua một đối tượng Bundle chứa đựng từ trạng thái trược của Activity
* Luôn theo sau bởi onStart()
* Phương thức:onRestart()
* Được gọi sau khi activity đã được dừng, chỉ một khoảng đang khởi động lần nữa (stared again)
* Luôn theo sau bởi onStart()
* Phương thức:onStart()
* Được gọi trước khi một activity visible với người dùng.
* Theo sau bởi onResume() nếu activity đến trạng thái foreground hoặc onStop() nế nó trở nên ẩn.
* Phương thức: onResume()
* Được gọi trước khi activity bắt đầu tương tác với người dung
* Tại thời điểm này activity ở trên dỉnh của stack activity.
* Luôn theo sau bởi onPause()
* Phương thức: onPause()
* Được gọi khi hệ thống đang resuming activity khác.
* Phương thức này là điển hình việc giữ lại không đổi dữ liệu.
* Theo sau bởi onResume nếu activity trở về từ ở trước, hoặc bởi onStop nếu nó trở nên visible với người dùng.
* Trạng thái của activity có thể bị giết bởi hệ thống.
* Phương thức: onStop()
* Được gọi khi activity không thuộc tầm nhìn của người dùng.
* Nó có thể diễn ra bởi vì nó đang bị hủy, hoặc bởi vì activity khác vữa được resumed và bao phủ nó
* Phương thức: onDestroy()
* Được gọi trước khi activity bị hủy.
* Đó là lần gọi cuối cùng mà activity này được nhận.
* Nó được gọi khác bởi vì activity đang hoàn thành, hoặc bởi vì hệ thống tạm thởi bị hủy diệt để tiết kiệm vùng nhớ.[2]

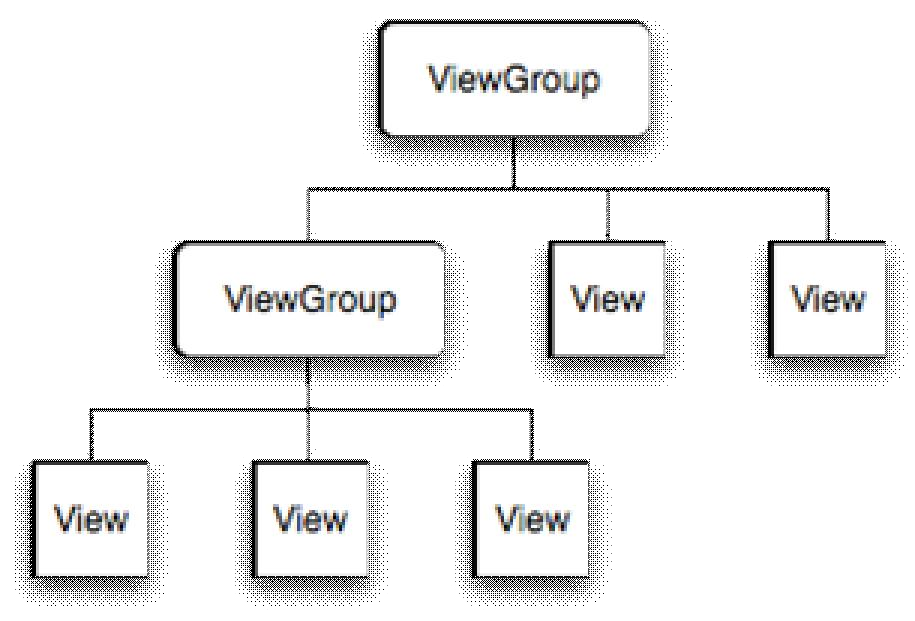
## Các thành phần giao diện trong Android

### 2.6.1. View

Trong một ứng dụng Android, giao diện người dùng được xây dựng từ các đối tượng View và ViewGroup. Có nhiều kiểu View và ViewGroup. Mỗi một kiểu là một hậu duệ của class View và tất cả các kiểu đó được gọi là các Widget.

Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, background, kích thước, lề. Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng View.

Trong Android Platform, các screen luôn được bố trí theo một kiểu cấu trúc phân cấp như hình dưới. Một screen là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một screen thì trong hàm onCreate của mỗi Activity cần phải được gọi một hàm là setContentView(R.layout.main); hàm này sẽ load giao diện từ file XML lên để phân tích thành mã bytecode.[1]



Hình 2.5.4. Cấu trúc một giao diện ứng dụng Android

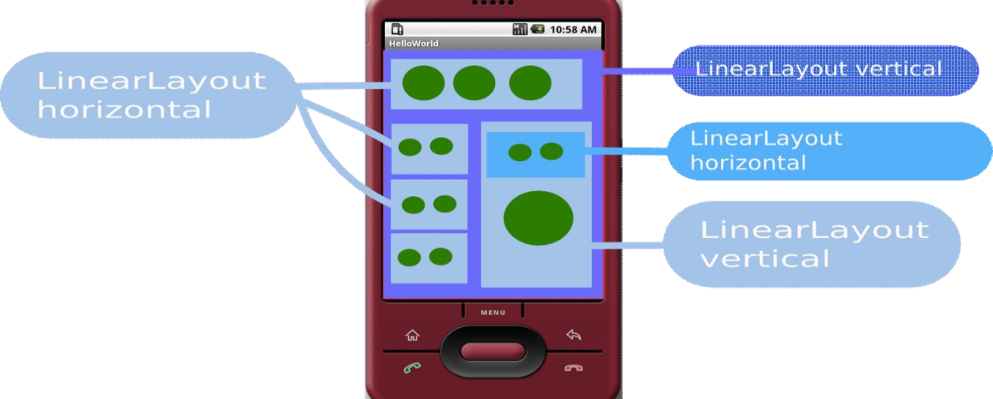
### 2.6.2. ViewGroup

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

* ***LinearLayout***

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

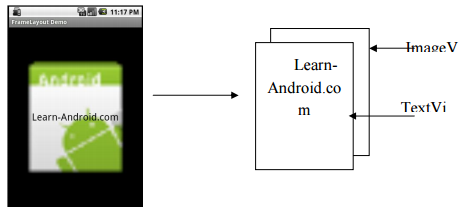
LinearLayout làm cho các thành phần trong nó không bị phụ thuộc vào kích thước của màn hình. Các thành phần trong LinearLayout được dàn theo những tỷ lệ cân xứng dựa vào các ràng buộc giữa các thành phần.[2]



Hình 2. Bố trí các widget sử dụng LinearLayout

* ***FrameLayout***

FrameLayout được dùng để bố trí các đối tượng theo kiểu giống như là các Layer trong Photoshop. Những đối tượng nào thuộc Layer bên dưới thì sẽ bị che khuất bởi các đối tượng thuộc Layer nằm trên. FrameLayer thường được sử dụng khi muốn tạo ra các đối tượng có khung hình bên ngoài chẳng hạn như contact image button.



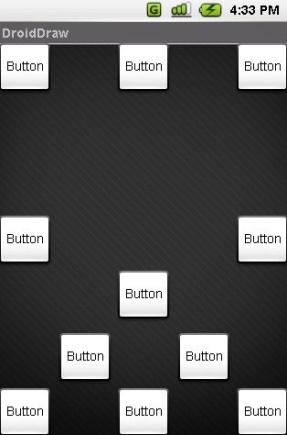
Hình 2. Bố trí các widget trong FrameLayout

* AbsoluteLayout

Layout này được sử dụng để bố trí các widget vào một vị trí bất kì trong layout dựa vào 2 thuộc tính toạ độ x, y. Tuy nhiên, kiểu layout này rất ít khi được dùng bởi vì toạ độ của các đối tượng luôn cố định và sẽ không tự điều chỉnh được tỷ lệ khoảng cách giữa các đối tượng. Khi chuyển ứng dụng sang một màn hình có kích thước với màn hình thiết kế ban đầu thì vị trí của các đối tượng sẽ không còn được chính xác như ban đầu.[1]

* RetaliveLayout

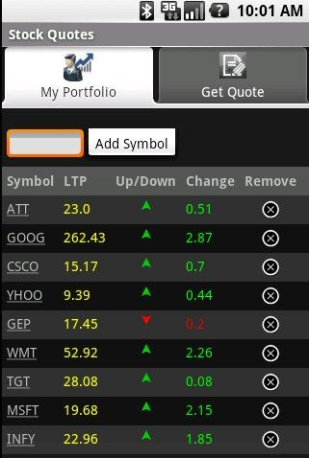
Layout này cho phép bố trí các widget theo một trục đối xứng ngang hoặc dọc. Để đặt được đúng vị trí thì các widget cần được xác định một mối ràng buộc nào đó với các widget khác. Các ràng buộc này là các ràng buộc trái, phải, trên, dưới so với một widget hoặc so với layout parent. Dựa vào những mối ràng buộc đó mà RetaliveLayout cũng không phụ thuộc vào kích thước của screen thiết bị. Ngoài ra, nó còn có ưu điểm là giúp tiết kiệm layout sử dụng nhằm mục đích giảm lượng tài nguyên sử dụng khi load đồng thời đẩy nhanh quá trình xử lý.[1]



Hình 2. Bố trí widget trong RetaliveLayout

* TableLayout

Layout này được sử dụng khi cần thiết kế một table chứa dữ liệu hoặc cần bố trí các widget theo các row và column. Chẳng hạn như, giao diện của một chiếc máy tính đơn giản hoặc một danh sách dữ liệu.



Hình 2. Bố trí widget trong TableLayout

### 2.6.3. Button

Sở dĩ widget button được giới thiệu đầu tiên trong số các widget khác là vì đây là đối tượng có thể nói là được dùng nhiều nhất trong hầu hết các ứng dụng Android.Để thiết kế giao diện với một button ta có 2 cách như sau:

Thiết kế bằng XML

<Button android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:id=[*"@+id*](about:blank)*/cmdButton1"*

android:text=*"Touch me!"*

android:onClick=*"touchMe"*/>

Thuộc tính android:onClick=*"touchMe"* được dùng để nắm bắt sự kiện click vào button. Khi sự kiện click button xảy ra thì phương thức “touchMe” được khai báo trong thẻ thuộc tính sẽ được gọi. Nếu trường hợp phương thức “touchMe” chưa được khai báo trong file mã nguồn tương ứng thì sẽ phát sinh một exception. Ngược lại, phương thức “touchMe” sẽ nhận được một đối tham biến là đối tượng View nơi đã phát sinh ra sự kiện. Đối tượng View này có thể ép kiểu trực tiếp sang kiểu Button vì thực chất nó là một button.

VD: trong file mã nguồn khai báo một hàm như sau:

public void touchMe(View v){

Button me = (Button) v; Me.setText(“Touched”);

}

Thiết kế bằng code

Thực ra mà nói thì nếu không phải đòi hỏi phải custom lại một widget thì không cần phải sử dụng tới code. Trong một số trường hợp bắt buộc chúng ta phải custom các widget để cho phù hợp với hoàn cảnh. Chẳng hạn như trong game, các menu hay các nút điều khiển.

Để khai báo một Button trong code ta làm như sau:

Button cmdButton = new Button(this);

cmdButton.setText(“Touch Me!”);

cmdButon.setOnClickListener(…);[12]

2.6.4. ImageButton

Cũng tương tự như Button, ImageButton chỉ có thêm một thuộc tính android:src = [“@drawabl](about:blank)e/icon” để thêm hình ảnh vào và không có thẻ text <ImageButton

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:id=["@+id](about:blank)/cmdButton1"

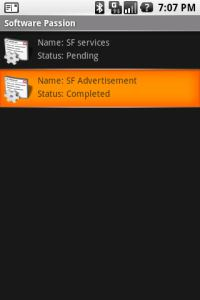
android:src=["@drawable](about:blank)/icon" android:onClick=*"touchMe"*/>

### 2.6.5. ImageView

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

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

* **Data Source:** Data Source có thể là một ArrayList, HashMap hoặc bất kỳ một cấu trúc dữ liệu kiểu danh sách nào.
* **Adapter:** Adapter là một class trung gian giúp ánh xạ dữ liệu trong Data Source vào đúng vị trí hiển thị trong ListView. Chẳng hạn, trong Data Source có một trường name và trong ListView cũng có một TextView để thể hiện trường name này. Tuy nhiên, ListView sẽ không thể hiển thị dữ liệu trong Data Source lên được nếu như Adapter không gán dữ liệu vào cho đối tượng hiển thị.
* **ListView:** ListView là đối tượng để hiển thị các thông tin trong Data Source ra một cách trực quan và người dùng có thể thao tác trực tiếp trên đó.[4]



Hình 2. Minh hoạ cho một ListView

### 2.6.6. TextView

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

VD: TextView textView = (TextView)findViewById(R.id.textView);

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

Nội dung TextView cũng có thể được định dạng bằng thẻ html ngay trong XML.[1]

**2.6.7. EditText**

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

<EditText android:id=["@+id](about:blank)/EditText01" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textStyle="bold"

android:textSize="20dip" android:textColor="#000000" android:text="Hello Android!" android:singleLine="true" android:inputType="textCapWords"/>

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

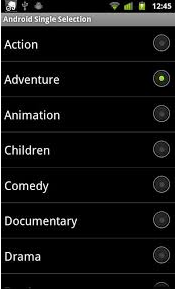
android:inputType = “…” sử dụng để xác định phương thức nhập cho EditText. Chẳng hạn như khi bạn muốn một ô để nhập password hay một ô để nhập Email thì thuộc tính này sẽ làm điều đó.

android:singleLine = “true” EditText của bạn sẽ trở thành một TextField, ngược lại sẽ là TextBox.[2]

**2.6.8. Radiobutton**

Radiobutton là một dạng nút 2 trạng thái được chọn (checked) hoặc không được chọn (unchecked). Khác với checkbox ở chỗ khi có 1 radiobutton checked thì không unchecked được nữa.

Radiobutton thường được dung với RadioGroup. Một radiobutton trong radiogroup đã được check thì tất cả các RadioButton khác cùng RadioGroup sẽ bị unchecked.[2]



Hình 2. Minh họa Radiobutton.

### 2.6.9. ContextMenu

ContextMenu được sử dụng để hiển thị các tuỳ chọn khi người dùng nhấn dài vào một cell nào đó trong ListView. Để tạo một ContextMenu ta cũng có 2 cách giống như tạo MenuOptions ở trên chỉ khác tên phương thức.

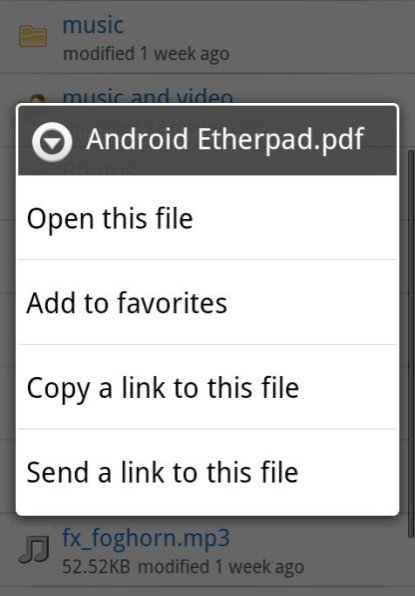
Khi nhấn dài vào một cell trong ListView thì phương thức:

**public void** onCreateContextMenu (ContextMenu menu, View v, ContextMenuInfo menuInfo) sẽ được gọi và truyền vào 3 tham số là:

ContextMenu: đối tượng để add các context menu item

View: Đối tượng nơi mà xảy ra sự kiện.

ContextMenuInfo: Cho biết vị trí xảy ra sự kiện trong ListView. Để biết được vị trí xảy ra sự kiện ta làm như sau



Hình 2. Minh hoạ context menu

### 2.6.10. Activity & Intend

* ***Activity***

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

Chu kỳ sống của một Activity

Tạo menu và dialog

Khởi động một Activity

Để khởi động một Activity ta sử dụng Intend sẽ tìm hiểu kỹ hơn ở phần b. Tuy nhiên, trong phần này tôi sẽ hướng dẫn cách chuyển giữa các Intend theo 2 loại:

Khai báo không tường minh:

Cung cấp chính xác thông tin của activity cần gọi bằng cách truyền vào tên class của Activity đó

Task:

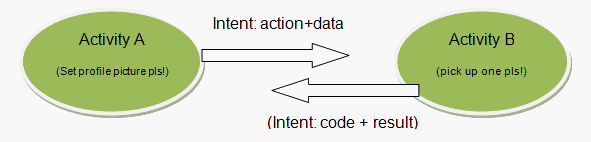
Android là một hệ điều hành đa tiến trình. Khi lập trình trên nền tảng Android thì tiến trình là một vấn đề cần phải được chú ý nhiều nhất. Mặc dù Android hỗ trợ đa tiến trình nhưng trên một thiết bị di động với cấu hình thấp mà chúng ta quá lạm dụng tiến trình thì sẽ rất tốn bộ xử lý điều này cũng đồng nghĩa với việc bạn đang biến ứng dụng của bạn trở thành một thứ phần mềm tiêu thụ điện năng.[14]

* ***Intent***

Khái niệm Intend:

Là một cấu trúc dữ liệu mô tả cách thức, đối tượng thực hiện của một Activity

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

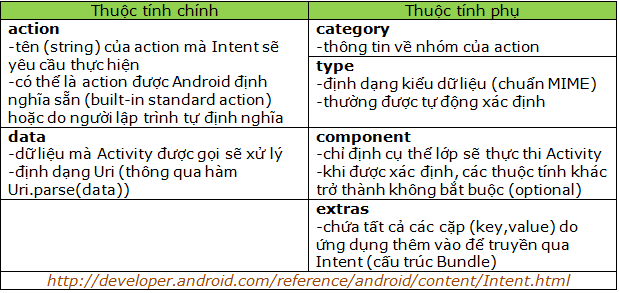


Hình 2. Truyền dữ liệu giữa 2 Activity

Dữ liệu của Intend:

Intent về cơ bản là một cấu trúc dữ liệu, được mô tả trong lớp android.content.Intent

Các thuộc tính của một đối tượng Intend:



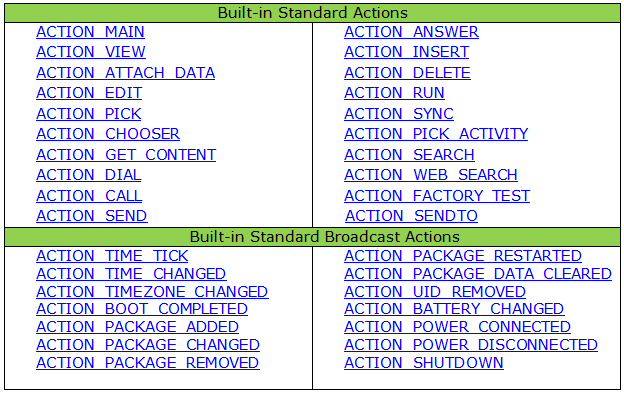
Hình 2. Các thuộc tính của Intend

Các Action được định nghĩa sẵn:

Dưới đây là những hằng String đã được định nghĩa sẵn trong lớp Intent. Đi kèm với nó là các Activity hay Application được xây dựng sẵn sẽ được triệu gọi mỗi khi Intent tương ứng được gửi (tất nhiên khi được cung cấp đúng data). VD: Gọi tới một số điện thoại:

Intent dialIntent =new Intent(Intent.ACTION\_DIAL, ri.parse("tel:123456"));

startActivity(dialIntent);[5]



Hình 2. Các Action đã được định nghĩa sẵn trong Intend

### 2.6.11. Expandable Listview

**Android Expanable** ListView là view hiển thị các mục trong danh sách hai cấp cuộn theo chiều dọc. Nó khác với Listview là nó bao gồm các nhóm con(group item) có thể dễ dàng mở rộng bằng cách chạm vào dể xem các mục con tương ứng của chúng (child item)

**Expandable ListView Adapter** trong android được sử dụng để load dữ liệu vào cho các items của expandable listview. Sau đây là một số method quan trọng của class expandable listview

getGroupView: Bản chất giống getView trong ListView. Nhưng getGroupView sẽ trả về View hiển thị Group Header.

getChildView: Bản chất cũng giống như getView(). Nhưng ở đây getChildView trả về View để hiển thị view trong Header View.

getGroupCount: Trả về số phần tử của group.

getChildrenCount: Trả về số phần tử con ứng với groupPosition.

getGroup: Trả về object của header group. Có nghĩa là trả về phần tử tại groupPosition trong danh sách header group.

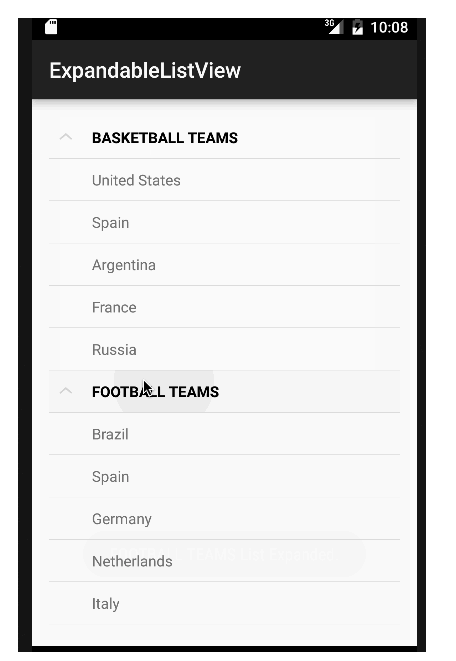
Các sự kiện đáng chú ý của ExpandableListView được đưa ra dưới đây:

OnChildClick: Được gọi khi click vào một child item.

OnGroupClick: Được gọi khi click vào một group item.

OnGroupCollapse: Được gọi khi một group được rút gọn.

OnGroupExpand: Được gọi khi một group được mở rộng



Hình 2. Hình ảnh minh họa Expandable Listview

## Ngôn ngữ lập trình Java

**Java** (phiên âm Tiếng Việt: "Gia-va") là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) (OOP) và dựa trên các lớp (class)[[9]](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)#cite_note-FOOTNOTEGoslingJoySteeleBracha20141-9). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì [biên dịch](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_bi%C3%AAn_d%E1%BB%8Bch) [mã nguồn](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n) thành [mã máy](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%A1y) hoặc [thông dịch](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_th%C3%B4ng_d%E1%BB%8Bch) mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành [bytecode](https://vi.wikipedia.org/wiki/Bytecode), bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và [C++](https://vi.wikipedia.org/wiki/C%2B%2B), nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Perl](https://vi.wikipedia.org/wiki/Perl), [PHP](https://vi.wikipedia.org/wiki/PHP) gấp nhiều lần. Java chạy tương đương so với [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng), một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy[[10]](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)" \l "cite_note-10)[[11]](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)#cite_note-11)

[Cú pháp](https://vi.wikipedia.org/wiki/C%C3%BA_ph%C3%A1p_h%E1%BB%8Dc) Java được vay mượn nhiều từ [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) & [C++](https://vi.wikipedia.org/wiki/C%2B%2B) nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn. Nhưng về lập trình hướng đối tượng thì Java phức tạp hơn.

Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác". Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.[2]

## SQLite

**SQLite** là hệ thống [cơ sở dữ liệu quan hệ](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u_quan_h%E1%BB%87" \o "Cơ sở dữ liệu quan hệ) nhỏ gọn, hoàn chỉnh, có thể cài đặt bên trong các [trình ứng dụng](https://vi.wikipedia.org/w/index.php?title=Tr%C3%ACnh_%E1%BB%A9ng_d%E1%BB%A5ng&action=edit&redlink=1" \o "Trình ứng dụng (trang chưa được viết)) khác. SQLite được Richard Hipp viết dưới dạng [thư viện](https://vi.wikipedia.org/wiki/Th%C6%B0_vi%E1%BB%87n" \o "Thư viện) bằng ngôn ngữ lập trình [C](https://vi.wikipedia.org/wiki/C).

SQLite có các ưu điểm sau:

Tin cậy: các hoạt động transaction (chuyển giao) nội trong [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u" \o "Cơ sở dữ liệu) được thực hiện [trọn vẹn](https://vi.wikipedia.org/w/index.php?title=T%C3%ADnh_to%C3%A0n_v%E1%BA%B9n_(giao_d%E1%BB%8Bch)&action=edit&redlink=1" \o "Tính toàn vẹn (giao dịch) (trang chưa được viết)), không gây lỗi khi xảy ra sự cố [phần cứng](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_c%E1%BB%A9ng" \o "Phần cứng)

Tuân theo chuẩn [SQL92](https://vi.wikipedia.org/w/index.php?title=SQL92&action=edit&redlink=1) (chỉ có một vài đặc điểm không hỗ trợ)

Không cần cài đặt cấu hình

Kích thước chương trình gọn nhẹ, với cấu hình đầy đủ chỉ không đầy 300 [kB](https://vi.wikipedia.org/wiki/Kilobyte)

Thực hiện các thao tác đơn giản nhanh hơn các hệ thống cơ sở dữ liệu [khách/chủ](https://vi.wikipedia.org/w/index.php?title=Kh%C3%A1ch/ch%E1%BB%A7&action=edit&redlink=1" \o "Khách/chủ (trang chưa được viết)) khác[[1]](https://vi.wikipedia.org/wiki/SQLite" \l "cite_note-1)

Không cần [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m" \o "Phần mềm) phụ trợ

[Phần mềm tự do](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_t%E1%BB%B1_do) với [mã nguồn](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n" \o "Mã nguồn) mở, được chú thích rõ rang

## Service

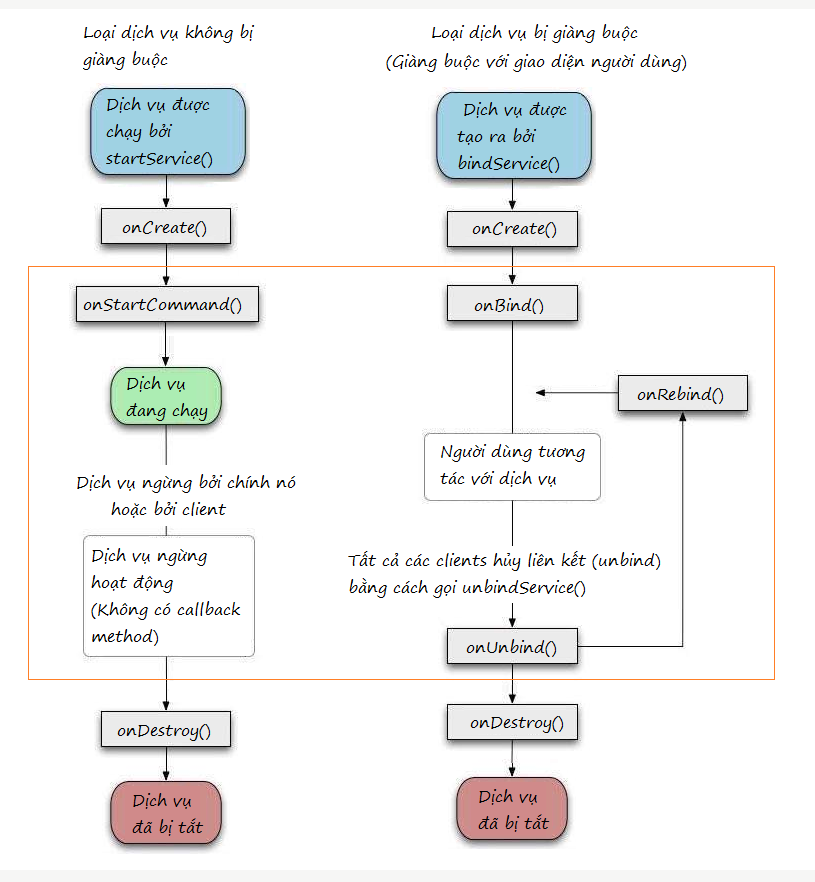
Một dịch vụ (Service[19]) là một thành phần chạy ngầm trên hệ điều hành để thực hiện các hoạt động dài hạn mà không cần phải tương tác với người sử dụng và nó hoạt động ngay cả khi ứng dụng bị phá hủy. Service không cung cấp giao diện cho người dùng, ví dụ như có thể phát nhạc trong khi người dùng đang sử dụng một ứng dụng khác.

Có hai loại service là Started và Bound(Bind).

Một dịch vụ được gọi là started (được khởi động) khi một thành phần ứng dụng, chẳng hạn như **Activity** khởi động nó bằng cách gọi **startService()**. Một khi được gọi, dịch vụ này có thể chạy ở chế độ nền vô thời hạn, thậm chí cả khi thành phần đã khởi động nó bị phá hủy.  
Dịch vụ này còn được gọi là dịch vụ không bị giàng buộc (Un Bounded Service).

Một bound service cung cấp một giao diện client-server để cho phép các thành phần có thể tương tác được với nhau. Thời gian sống của một bound service phụ thuộc vào thành phần liên kết (bound) với nó, tức là thành phần liên kết còn tồn tại thì bound service  còn tồn tại và ngược lại.Nhiều thành phần có thể liên kết với một bound service tại một thời điểm, nhưng khi tất cả các thành phần này đều ngưng liên kết (unbind) thì service loại này cũng bị hủy theo.

Dưới đây là vòng đời service của android:



Hình 2. Vòng đời của service

onStartCommand(): Hàm này được gọi khi service bắt đầu bằng cách gọi hàm startService(). Khi hàm này thực hiện, service được khởi động và có thể chạy nền background vô thời hạn. Để có thể hủy service thì chỉ có 2 cách:

* Bạn tự gọi hàm stopSelf() hoặc stopService().
* May rủi chờ đến khi hệ thống Android cần tài nguyên cho ứng dụng khác thì sẽ kill service này .

onBind(): Hàm này chỉ được thực hiện nếu service là bound service. Bạn phải cung cấp một giao diện(interface) mà client sử dụng để giao tiếp với Service bằng cách trả về một IBinder. Hàm này được gọi bất kể khi nào có một thành phần nào đó muốn bind vào service

onCreate(): Đây là hàm được khi service được khởi tạo và chỉ được gọi duy nhất một lần

onDestroy(): Tương tự như Activity thì hàm này được khi service bị hủy(có thể do bạn gọi hàm stopService()hay do hệ thống tự hủy) để giải phóng tài nguyên.

## Broadcast Receivers

Là một thành phần dùng để trả lời các thông điệp phát ra từ hệ thống. Trong quá trình thiết bị di động vận hành có rất nhiều thông điệp khác nhau từ hệ thống thông báo như: màn hình tắt, pin yếu, chụp màn hình,… Một broadcast receiver[17] không tạo ra một giao diện người dùng nhưng nó tạo ra một thông báo (notification) trên thanh trạng thái (status bar) để báo cho người dùng một sự kiện nào đó. Nó phù hợp với việc thông báo các mũi tiêm từ ứng dụng đến người dùng

## Foreground services

Như đã nói ở trên service sẽ bị dead khi app bị kill hoặc sau 1 khoảng thời gian cố định. Vậy câu hỏi đặt ra ở đây làm thế nào để ứng dụng khi bị kill mà service vẫn chạy ngầm một cách riêng biệt. Để giải quyết được vấn đề này, foreground services[18] đã được ra đời.

Foreground services phải được thể hiện bởi một notification với độ ưu tiên PRIORITYLOW trở lên giúp đảm bảo người dùng vẫn biết được ứng dụng của mình đang làm gì trong service. Trong khi foreground service đang thực thi thì ta không thể hủy bỏ notification

1. Alarm manager

Alarm manager trong android một cầu nối giữa ứng dụng và alarm service của hệt thống android. Nó có thể gửi 1 bản tin broadcast tới ứng dụng ở thời điểm đã được lên lịch trước đó. Sau đó ứng dụng có thể thực hiện bất kỳ tác vụ nào cần thiết ở thời điểm đó. VD điển hình đó chính là báo thức

Cùng với sự trợ giúp của PendingIntent và Intent, một bundle đóng gói các thông tin cần thiết sẽ được gửi cùng broadcast hệ thống tới ứng dụng của bạn khi thời gian đặt lịch đến

Từ phiên bản Android M thì google đã có sự thay đổi với Alarm Manager để hạn chế việc tiêu thụ pin quá mức. Alarm sẽ không còn được ưu tiên bật chính xác vào thời điểm đã đặt, mà tùy thuộc vào tình hình tài nguyên máy thời điểm đó mà thời gian bật alarm sẽ có sai lệch chút ít

Có 2 bước để khởi tạo AlarmManager trong Android, đó là:

Tạo một BroadcastReceiver để nhận các bản tin broadcast từ hệ thống, tất nhiên bạn cần phải đăng ký trong xml

Đăng ký alarm với Alarm hệ thống ALARM\_SERVICE thông qua PendingIntent kèm thời gian đặt trước.

Xử lý sự kiện Alarm

AlarmManager cung cấp 2 cách để lắng nghe một alarm broadcast. Đó là:

* Local listener (AlarmManager.OnAlarmListener)
* Broadcast Receiver được quy định tại Intent bọc bên trong một PendingIntent

Việc triển khai AlarmManager.OnAlarmListener() tương tự như cách sử dụng PendingIntent. Đơn giản là yêu cầu một callback xử lý sự kiện alarm trong đó.

Có một hạn chế với cách sử dụng AlarmManager.OnAlarmListener() đó là nó không thể hoạt động nếu Activity/Fragment tương ứng bị destroy. Ngoài ra, dùng PendingIntent còn có ưu điểm là ứng dụng có thể nhận broadcast alarm ngay cả khi ứng dụng bị kill bởi người dùng trước đó.

Những cách thiết lập thời gian schedule cho alarm manager Android

Như mình đã nói ở trên, do Google muốn hạn chế việc tiêu thụ pin quá đà nên việc thiết lập thời gian schedule bởi Alarm sẽ không chính xác.

Ví dụ: bạn thiết lập đúng 8h00 AM thì ứng dụng tự động bật một bài hát. Nhưng thực tế thì có thể 8h02 AM ứng dụng mới nhận broadcast alarm và mới bật được nhạc. Vẫn có những sai số về thời gian.

Bình thường bạn hay dùng tới 2 api:

1.set() – lên lịch thời gian nhưng cho phép Android linh động thời điểm kích hoạt

2.setExact()- yêu cầu Android kích hoạt chính xác thời điểm

Tuy nhiên, từ Android M, Google có thêm khái niệm Doze và App Standby. Hai chế độ này sẽ hủy các alarm được lên lịch trước đó khi thiết bị không được cắm sạc.

Khi người dùng tắt màn hình hoặc không cắm sạc, thiết bị sẽ chuyển sang chế độ Doze. Trong chế độ Doze, Android sẽ hạn chế thiết bị truy cập vào mạng và dịch vụ sử dụng nhiều CPU. Do đó, các alarm cũng sẽ bị hủy hoặc bị hoãn cho tới khi thiết bị sẵn sàng.

Android sẽ định kỳ thoát và vào lại chế độ Doze, gọi là khoảng thời gian bảo trì. Trong thời gian bảo trì này, tất cả các hạn chế trước đó sẽ được giải phóng. Tức là các yêu cầu truy cập mạng sẽ được phép, các alarm được kích hoạt (đó chính là lý do tại sao mình nói ở trên là thời gian kích hoạt alarm sẽ bị sai lệch).

Chế độ App Standby cũng tương tự chế độ Doze, chỉ khác là màn hình không bắt buộc phải tắt

Trong dự án, sử dụng alarm manager để thông báo bằng cách cài đặt thời gian thông báo cũng như là âm thanh thông báo đến cho người dùng

1. **Notification**

Notification là 1 user interface element, được hiển thị bên ngoài user interface của ứng dụng để thông báo cho người dùng những thông tin cần thiết. Người dùng có thể xem được những thông báo này ở màn hình khóa hoặc từ Notification Drawer

Để tạo một notification trong Android, bạn nên sử dụng NotificationCompat.Builder. Một notification được tạo ra phải có ít nhất ba thông tin dưới đây :

* small icon, sử dụng method setSmallIcon()
* title, sử dụng method setContentTitle()
* Nội dung của notificaiton, sử dụng method setContentText()

Để Notification hiển thị cho người dùng, bạn phải sử dụng NotificationManager như sau :

* Tạo một đối tượng của NotificationManager
* Sử dụng method notify() của đối tượng NotificationManager để đẩy thông báo cho người dùng bằng cách truyền vào hai tham số : Id của notification và đối tượng Notification.

Tạo action cho Notification trong Android Khi người dùng chạm vào Notification, bạn nên có navigation đến một Activity nào đó. Như vậy, bạn sẽ tạo một action cho Notification. Việc tạo action cho Notification, bạn nên sử dụng PendingIntent bằng cách sau:

* Tạo Intent với Activity sẽ được hiển thị khi người dùng chạm vào Notification
* Tạo PendingIntent để mang thông tin của Intent khi gửi cho Notification

Một Notification đầy đủ bao gồm 6 thành phần:

* Small icon: Nó là cần thiết và được gắn qua setSmallIcon().
* App name: Nó được cung cấp bởi hệ thống.
* Timestam: Nó được cung cấp bởi hệ thống, nhưng bạn có thể override với setWhen() hoặc không hiển thị với setShowWhen(false).
* Large icon: Nó không bắt buộc phải có (nó thường được sử dụng các hình ảnh liên lạc, không sử dụng nó cho icon ứng dụng của bạn) và gán nó qua setLargeIcon().
* Title: Nó là không bắt buộc và gán nó qua setContentTitle().
* Text: Nó không bắt buộc bạn có thể gán nó qua setContentText(). Để gán quyền ưu tiên cho một notification bạn gán qua setPriority(). Giá trị priority xác định mức độ được hiển thị trong các tình huống khác nhau trong Android 7.1 hoặc thấp hơn. (Với Android 8.0 hoặc cao hơn, bạn phải gán giá trị chanel importance - sẽ được trình bày ở phần dưới đây.)

Trong dự án sử dụng notification để giúp người dùng thấy được thông báo của ứng dụng trên màn hình khóa hoặc ở Notification Drawer

# CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Đặc tả yêu cầu

Từ những mô tả yêu cầu chức năng được trình bày ở mục 1.4 (chương 1), phân tích làm rõ quy trình nghiệp vụ của hệ thống quản lý đặt món gồm các chức năng:

Chức năng quản lý thông tin trẻ: cho phép người người dùng tự quản lý thông tin cá nhân về trẻ em của họ.

Chức năng quản lý sức khỏe trẻ: cho phép người người dùng tự quản lý sức khỏe trẻ em của họ. Từ đó tạo nên biểu đồ thống kê về sức khỏe

Chức năng xem lịch sử, cập nhật thông tin mũi tiêm của từng trẻ: cho phép người người dùng tự quản lý thông tin mũi tiêm phòng của từng trẻ em như. Cập nhật lại trạng thái mũi tiêm (đã tiêm hoặc chưa tiêm), xem lịch sử tiêm phòng (đã tiêm/chưa tiêm/đã qua).

Chức năng nhắc nhở người dùng tới ngày tiêm phòng. Người dùng tự quản lý thời gian nhắc nhở tiêm phòng của họ thông qua cài đặt thông báo.

## Phân tích yêu cầu chức năng

Dựa vào khảo sát và đặc tả chức năng, ta có thể xác định được các tác nhân chính của hệ thống như sau:

Người dùng: Quản lý thông tin trẻ em, quản lý/theo dõi sức khỏe, quản lý lịch tiêm phòng của từng trẻ nhỏ, quản lý về thời gian thông báo. Để thực hiện các chức năng này người cần phải khởi động và có dữ liệu trong ứng dụng.

Hệ thống: thực hiện các chức năng thông báo tiêm phòng, tạo phác đồ cá nhân và tạo thống kê sức khỏe cho từng trẻ.

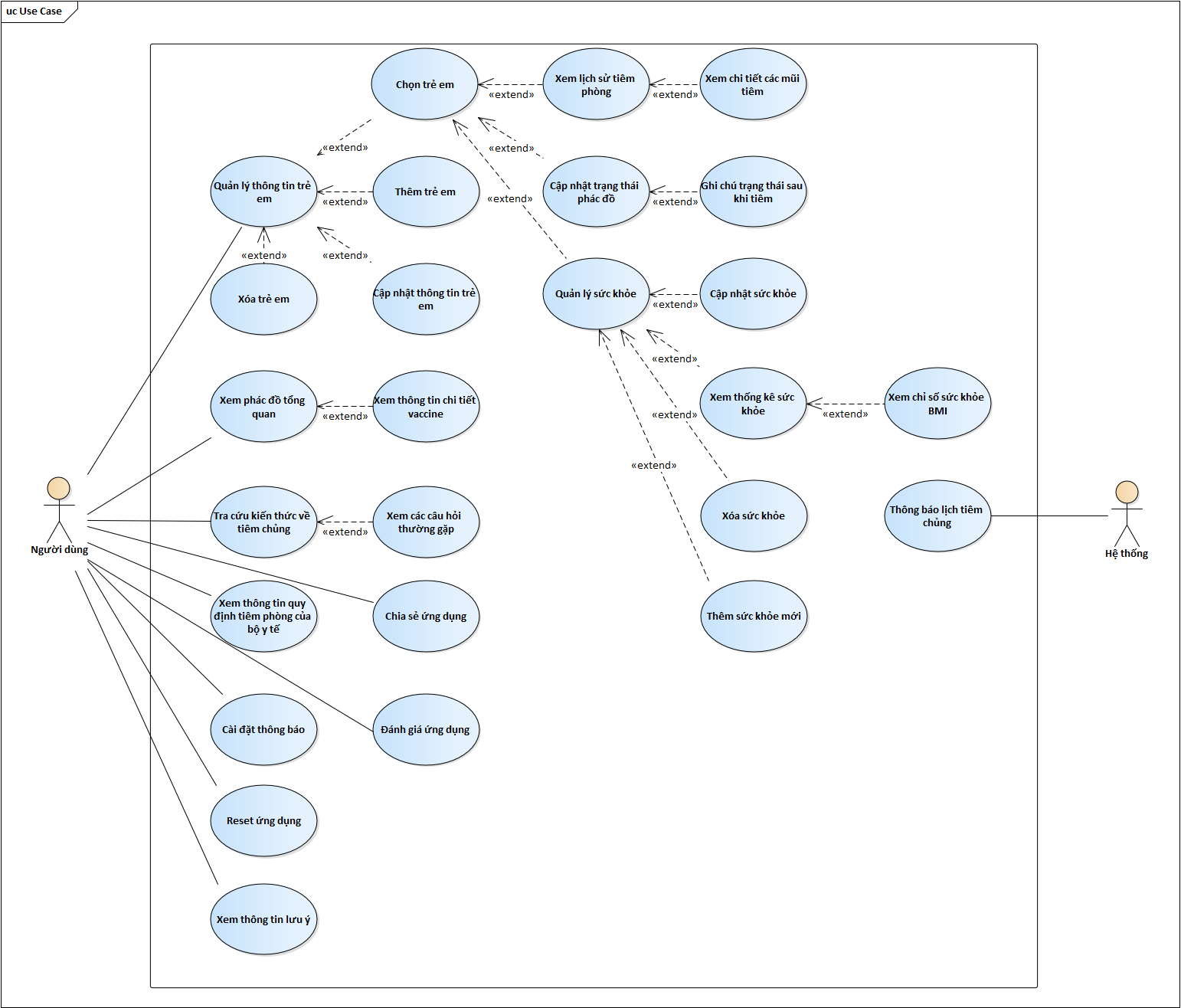
## Các yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| **Mục** | **Tên yêu cầu** | **Mô tả yêu cầu** |
| 1 | Giao diện | Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với mọi người dùng. |
| 2 | Tốc độ xử lý | Hệ thống phải xử lý nhanh chóng và chính xác. |
| 3 | Bảo mật | Tính bảo mật và độ an toàn cao. |
| 4 | Thời gian hoạt động | Có thể hoạt động tốt 24/24. |

Bảng 3. Yêu cầu phi chức năng

Biểu đồ use case

### Mô hình Use case



Hình 3. Mô hình Use case

### Danh sách tác nhân và mô tả

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả tác nhân** |
| Người dùng | Là những người sử dụng được hầu hết các chức năng của ứng dụng như: quản lý thông tin trẻ em, cài đặt thông báo, theo dõi sức khỏe của trẻ…. |
| Hệ thống | Hệ thống có chức năng thông báo nhắc nhở tiêm phòng sau khi xử lý các vấn đề về mặt thời gian. Giúp người dùng nhận biết được ngày tiêm phòng |

Bảng 3.3 Danh sách tác nhân và mô tả

### Danh sách Use case và mô tả

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Tên Use case** | **Mô tả ngắn gọn Use case** | **Chức năng** |
| UC001 | Xem thông tin lưu ý | Người dùng xem thông tin lưu ý để biết thêm các thông tin chi tiết về ứng dụng. | Xem thông tin lưu ý |
| UC002 | Reset ứng dụng | Người dùng khởi tạo lại mặc định trạng thái ban đầu của ứng dụng khi vừa tải về (tại thời điểm lần đầu sử dụng). | Quản lý thông tin ứng dụng |
| UC003 | Cài đặt thông báo | Người dùng tự mình quản lý thời gian thông báo mỗi khi đến ngày tiêm. Có thể mở/tắt thông báo tiêm chủng. Với các thông tin của cài đặt thông báo là: switch(bật/tắt thống báo), báo đúng ngày và trước 3 ngày, thời điểm thông báo. | Quản lý thông báo |
| UC004 | Đánh giá ứng dụng | Giúp người dùng có thể đánh giá được ứng dụng có tốt hay không, hỗ trợ người dùng nên lên đánh giá của bản than, |  |
| UC005 | Chia sẻ ứng dụng | Chia sẽ cài đặt ứng dụng với máy khác |  |
| UC006 | Tra cứu về kiến thức tiêm chủng | Người dùng biết thêm một số thông tin hữu ích về vấn đề tiêm chủng cho trẻ thông qua các công hỏi thường gặp. | Xem danh sách các câu hỏi thường gặp trong vaccine |
| UC007 | Xem các câu hỏi thường gặp | Người dùng có thể xem chi tiết về thông tin câu trả lời của từng câu hỏi thường gặp trong vấn đề tiêm chủng cho trẻ em. | Xem thông tin chi tiết của câu hỏi thường gặp |
| UC008 | Xem phác đồ tổng quan | Hệ thống hiển thị phác đồ tổng quan giúp người dùng dễ dàng biết lịch tiêm chủng chung. | Hiển thị phác đồ tổng quát lịch tiêm phòng cho trẻ |
| UC009 | Xem thông tin chi tiết vaccine | Người dùng chọn bất kỳ item trong phát đồ tổng quan sẽ hiển thị thông tin cơ bản | Xem thông tin chi tiết của từng loại vaccine |
| UC010 | Quản lý thông tin trẻ em | Người dùng xem danh sách trẻ em hiện có và có thể chọn trẻ em để xem thông tin và cập nhật các mũi tiêm của trẻ đấy | Hiển thị danh sách trẻ em đã có dữ liệu tồn tại trong ứng dụng |
| UC011 | Thêm trẻ em | Người dùng tạo mới trẻ em với các thông tin cơ bản: họ tên, tên ở nhà, ngày sinh, cân nặng và chiều cao,... Sau đó hệ thống tự động phát sinh phác đồ dựa tên ngày sinh của trẻ | Tạo mới một trẻ em |
| UC012 | Xóa trẻ em | Người dùng xóa thông tin trẻ em (có dữ liệu tồn tại trong danh sách trẻ em) khỏi cơ sở dữ liệu | Xóa thông tin trẻ em đã tồn tại trong danh sách |
| UC013 | Cập nhật thông tin trẻ em | Người dùng cập nhật thông tintrẻ em: họ tên, tên ở nhà, ngày sinh, giới tính | Cập nhật thông tin trẻ em đã chọn |
| UC014 | Xem lịch sử tiêm phòng | Người dùng có thể xem được các mũi đã tiêm, sắp tiêm hoặc đã bỏ lỡ | Xem danh sách các mũi đã tiêm, đã bỏ lỡ, sắp tiêm |
| UC015 | Cập nhật trạng thái phát đồ | Người dùng cập nhật chi tiết mũi tiêm của từng trẻ em khác nhau | Cập nhật thông tin chi tiết mũi tiêm của trẻ |
| UC016 | Quản lý sức khỏe | Người dùng xem danh sách sức khỏe của trẻ em và thao tác với nó | Hiển thị danh sách sức khỏe từ cơ sở dữ liệu tương ứng với trẻ đã chọn |
| UC017 | Cập nhật sức khỏe | Người dùng xem danh sách sức khỏe của trẻ em và thao tác với nó | Cập nhật thông tin sức khỏe của trẻ đã chọn. |
| UC018 | Xóa sức khỏe | Người dùng xóa thông tin về sức khỏe của trẻ em (đã chọn) khỏi cơ sở dữ liệu | Xóa sức khỏe của trẻ đã chọn khỏi cơ sở dữ liệu |
| UC019 | Thêm sức khỏe mới | Người dùng tạo mới trẻ em với các thông tin cơ bản: họ tên, tên ở nhà, ngày sinh, cân nặng và chiều cao,... Sau đó hệ thống tự động phát sinh phác đồ dựa tên ngày sinh của trẻ | Tạo mới một sức khỏe của trẻ đã chọn |
| UC020 | Xem thống kê sức khỏe | Người dùng xem thống kê về sức khỏe của trẻ dựa trên dữ liệu về sức khỏe của chính trẻ ấy | Xem biểu đồ thống kê |

Bảng 3.3 Danh sách usecase và mô tả

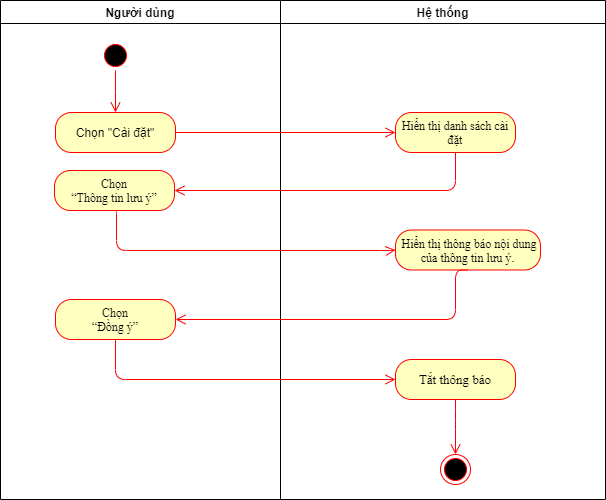
### Đặc tả use case

#### UC001 – Xem thông tin lưu ý

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC001 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng xem thông tin lưu ý trước khi sử dụng úng dụng. | |
| Điều kiện trước | Khởi động app | |
| Điều kiện sau | Hiển thị thành công nội dung “Thông tin lưu ý” | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “cài đặt” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị danh sách cài đặt |
| 3. Chọn “Thông tin lưu ý” | |  |
|  | | 4. Hiển thị nội dung của thông tin lưu ý. |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 5. Chọn Đồng ý | |  |
|  | | 6. Tắt hiển thị nội dung |

Bảng 3.4.4. UC001 – Xem thông tin lưu ý

Biểu đồ activity



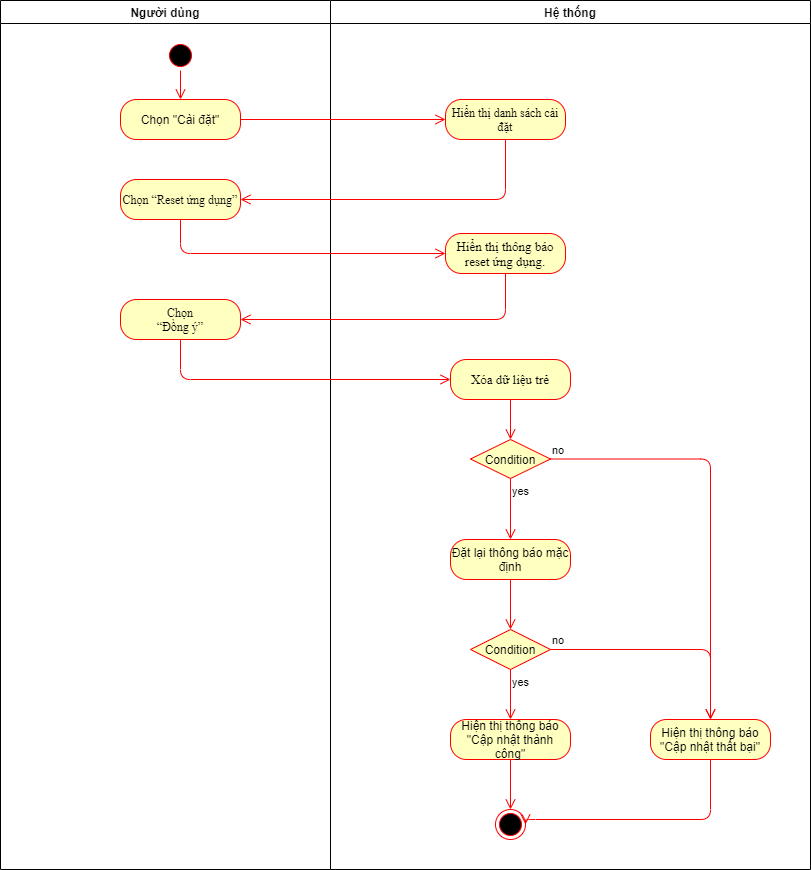
Hình 3.4. Activity Xem thông tin lưu

#### UC002 – Reset ứng dụng

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC002 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng khởi động lại úng dụng khi gặp sự cố. | |
| Điều kiện trước | Khởi động app | |
| Điều kiện sau | Hiển thị thông báo. Xóa hết dữ liệu liên quan đến trẻ em trong cơ sở dữ liệu, cập nhật lại trạng thái thông báo mặc định. | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “cài đặt” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị danh sách cài đặt |
| 3. Chọn “Reset ứng dụng” | |  |
|  | | 4. Hiển thị thông báo reset ứng dụng. |
| 5. Chọn “Đồng ý” | |  |
|  | | 6. Xóa hết dữ liệu của trẻ em |
|  | | 7. Cập nhật lại thông báo |
| 8. Hiển thị thông báo “Cập nhật thành công” | |  |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 5.1. Chọn “Hủy bỏ” | |  |
|  | | 6.1 Tắt thông báo. |

Bảng 3.4.4. UC002 – Reset ứng dụng

Biểu đồ activity



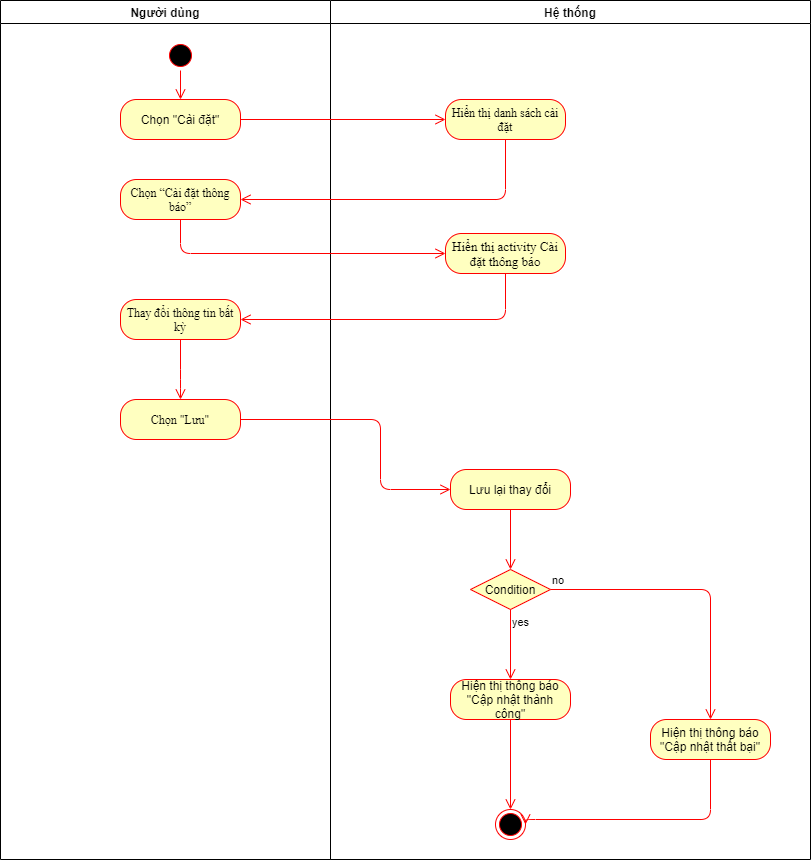
Hình 3.4. Actitity Reset ứng dụng

#### UC003 – Cài đặt thông báo

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC003 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dung tự cài đặt giờ, phút, ngày được thông báo. | |
| Điều kiện trước | Đăng nhập vào hệ thống | |
| Điều kiện sau | Hiển thị thông báo va lưu lại những thay đổi xuống cơ sở dữ liệu. | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “cài đặt” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị danh sách cài đặt |
| 3. Chọn “Cài đặ thông báo” | |  |
|  | | 4. Hiển thị activity “Cài đặt thông báo” |
| 5. Thay đổi thông tin bất kỳ | |  |
|  | | 6. Hiển thị thông tin thay đổi |
| 7. Chọn “Lưu” | |  |
|  | | 8. Hiển thị thông báo “Cập nhật thành công” và lưu lại thay đổi xuống db |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 7.1 Chọn hủy | | 8.1 Đóng activity “Cài đặt thông báo” và không lưu lại thay đổi |

Bảng 3.4.4. UC003 – Cài đặt thông báo

Biểu đồ activity



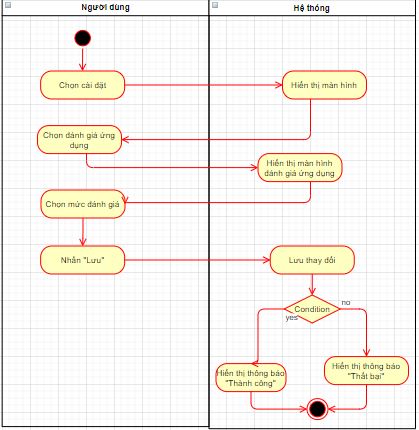
Hình 3.4. Activity Cài đặt thông báo

#### UC004 – Đánh giá ứng dụng

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC004 | |
| Tác nhân | Người dùng | |
| Mô tả | Giúp người dùng có thể đánh giá được ứng dụng có tốt hay không | |
| Điều kiện trước | Khởi động ứng dụng | |
| Điều kiện sau | Ứng dụng đã được đánh giá | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “cài đặt” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị màn hình “Cài đặt” |
| 3. Chọn đánh giá ứng dụng | |  |
|  | | 4. Hiển thị màn hình “Đánh giá ứng dụng” |
| 5. Chọn mức đánh giá | |  |
| 6. Nhấn “Xác nhận” | |  |
|  | | 7. Quay trở lại màn hình “Cài đặt” |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 6.1 Nhấn “Hủy” | | 7.1 Quay trở lại màn hình cài đặt |

Bảng 3.4.4. UC004 – Đánh giá ứng dụng

Biểu đồ activity



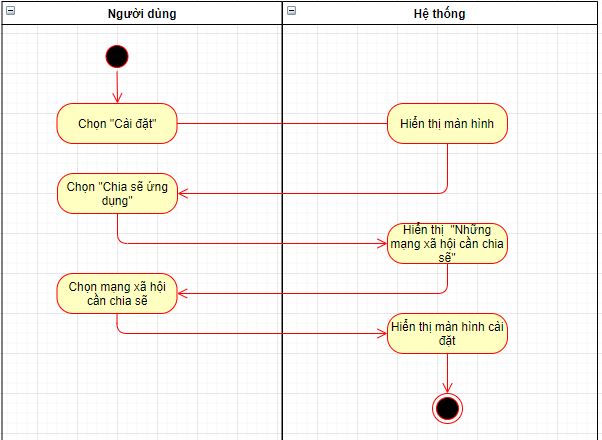
Hình 3.4. Activity Đánh giá ứng dụng

#### UC005 – Chia sẻ ứng dụng

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC005 | |
| Tác nhân | Người dùng | |
| Mô tả | Giúp người dùng có thể chia sẽ ứng dụng lên mạng xã hội | |
| Điều kiện trước | Khởi động ứng dụng | |
| Điều kiện sau | Chia sẽ ứng dụng thành công | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “Cài đặt” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị màn hình “Cài đặt” |
| 3. Chọn “Chia sẽ ứng dụng” | |  |
|  | | 4. Hiển thị “Những mạng xã hội cần chia sẽ” |
| 5. Chọn mạng xã hội cần chia sẽ | |  |
|  | | 6. Quay lại màn hình Cài đặt |

Bảng 3.4.4. UC005 – Chia sẻ ứng dụng

Biểu đồ activity



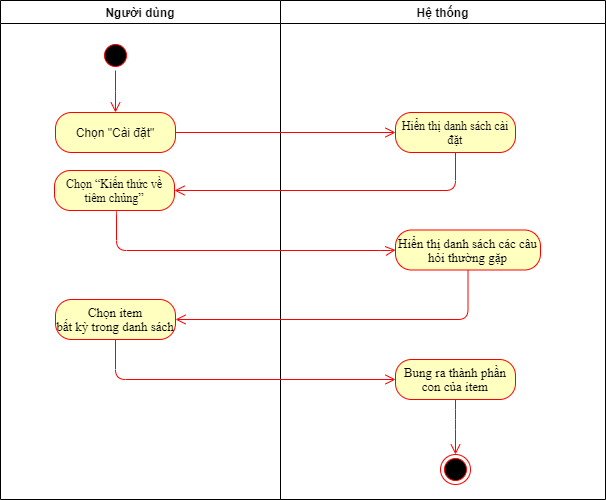
Hình 3.4. Activity Chia sẻ ứng dụng

#### UC006 – Tra cứu kiến thức tiêm chủng

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC006 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng biết thêm một số thông tin hữu ích về vấn đề tiêm chủng cho trẻ thông qua các công hỏi thường gặp | |
| Điều kiện trước | Khởi động hệ thống. | |
| Điều kiện sau | Hiển thị danh sách các câu hỏi thường gặp. | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “cài đặt” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị danh sách cài đặt |
| 3. Chọn “Kiến thức về tiêm chủng” | |  |
|  | | 4. Hiển thị danh sách các câu hỏi thường gặp. |
| 5. Chọn item bất kỳ trong danh sách | |  |
|  | | 6. Bung ra thành phần con của item đấy |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 5.1 Chọn lại vào i tem đó | | 6.1 Ẩn thành phần con của item |

Bảng 3.4.4. UC006 – Tra cứu kiến thức tiêm chủng

Biểu đồ activity



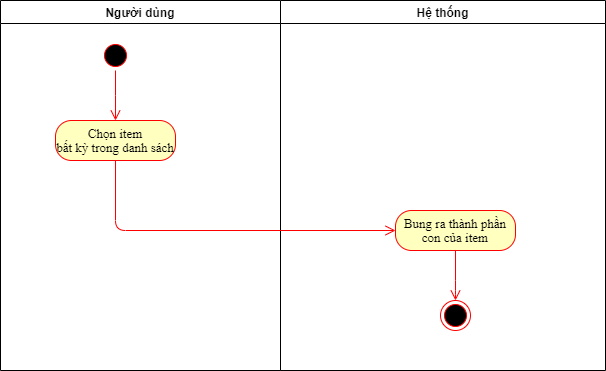
Hình 3.4. 6 Tra cứu kiến thức tiêm chủng

#### UC007 – Xem các câu hỏi thường gặp

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC007 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng biết thêm một số thông tin hữu ích về vấn đề tiêm chủng cho trẻ thông qua các công hỏi thường gặp | |
| Điều kiện trước | Thực hiện UC006 | |
| Điều kiện sau | Hiển thị thành phần con (có dữ liệu) của item đã chọn | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn item bất kỳ trong danh sách | |  |
|  | | 2. Bung ra thành phần con của item đấy |

Bảng 3.4.4. UC007 – Xem các câu hỏi thường gặp

Biểu đồ activity



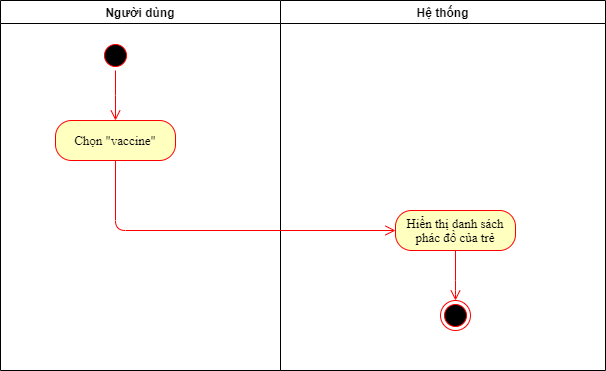
Hình 3.4. Activity Xem câu hỏi thường gặp

#### UC008 – Xem phác đồ tổng quan

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC008 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng khởi động hệ thống và chọn vaccine để xem thông tin phác đồ chung cho trẻ em | |
| Điều kiện trước | Khởi động hệ thống | |
| Điều kiện sau | Hiển thị phát đồ chung | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “vaccine” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 4. Hiển thị danh sách phác đồ của trẻ. |

Bảng 3.4.4. UC008 – Xem phác đồ tổng quan

Biểu đồ activity



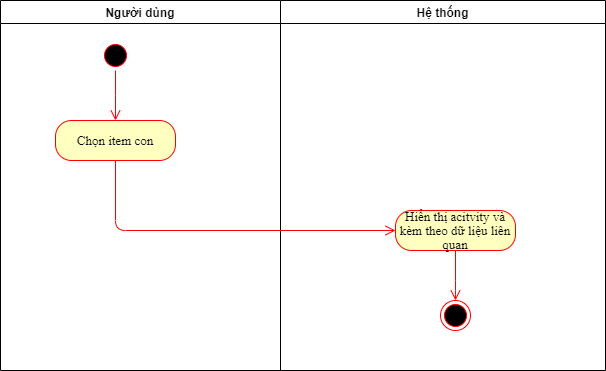
Hình 3.4. Xem phác đồ tổng quan

#### UC009 – Xem thông tin chi tiết vaccine

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC009 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng xem được các thông chi tiết của vaccine | |
| Điều kiện trước | Thực hiện UC008 | |
| Điều kiện sau | Hiển thị thông tin chi tiết của vaccine | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn item con bất kỳ trong group item | |  |
|  | | 2. Hiển thị activity “Thông tin vaccine” và kèm theo dữ liệ liên quan |

Bảng 3.4.4. UC009 – Xem thông tin chi tiết vaccine

Biểu đồ activity



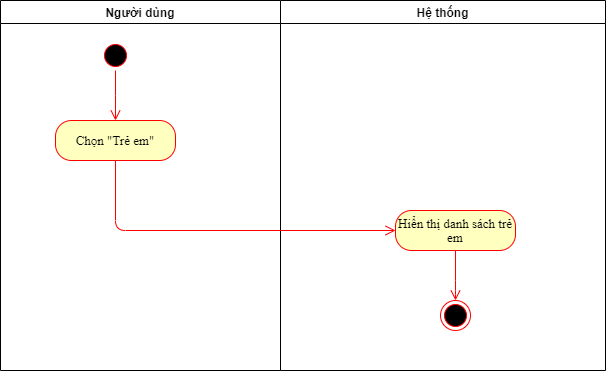
Hình 3.4. Activity Xem thông tin chi tiết vaccine

#### UC010 – Quản lý thông tin trẻ

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC010 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng xem được danh sách trẻ em hiện có trong ứng dụng. Cỏ thể thao tác thêm sửa hoặc xóa trẻ | |
| Điều kiện trước | Khởi động ứng dụng | |
| Điều kiện sau | Hiển thị danh sách trẻ | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “Trẻ em” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị danh sách trẻ em |

Bảng 3.4.4. UC010 – Quản lý thông tin trẻ

Biểu đồ activity



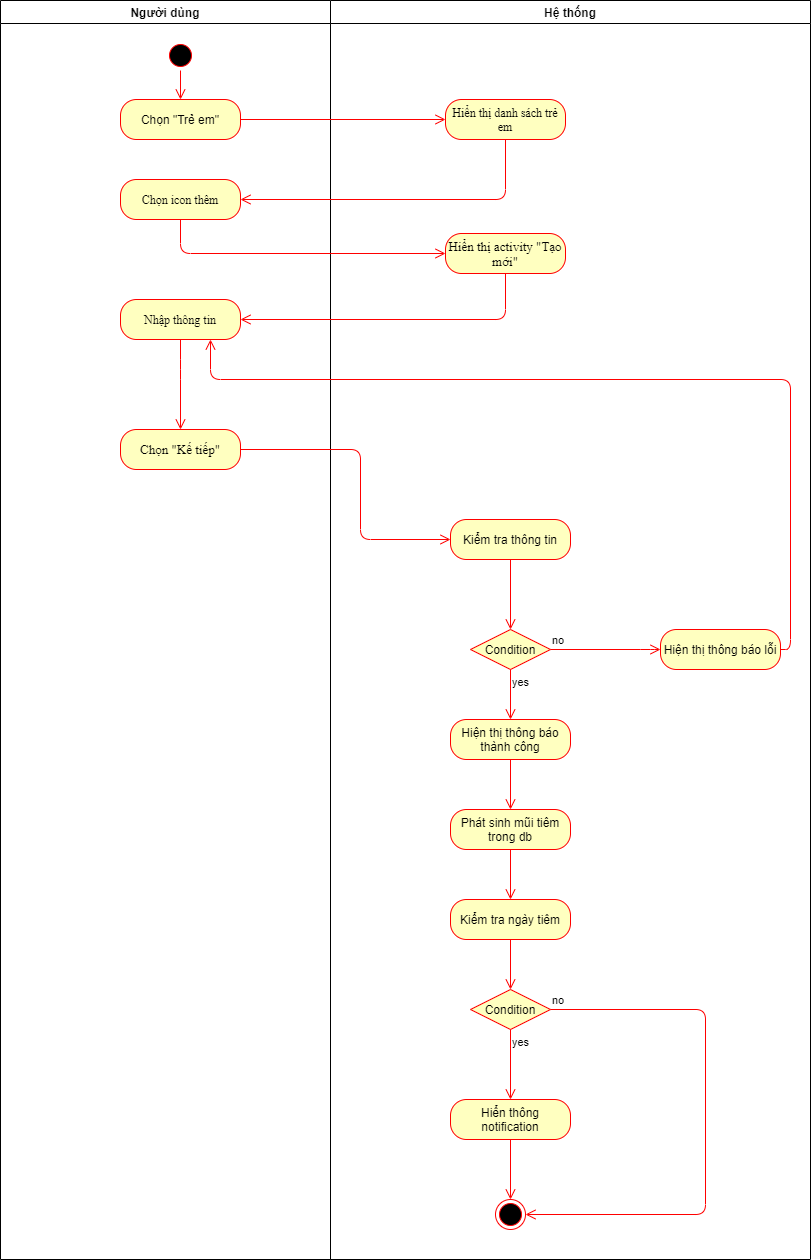
Hình 3.4. Quản lý thông tin trẻ

#### UC011 – Thêm trẻ em

|  |  |  |  |
| --- | --- | --- | --- |
| Mã usecase | UC011 | | |
| Tác nhân | Người dùng | | |
| Mô tả | Người dùng thêm mới thông tin trẻ | | |
| Điều kiện trước | Khởi động app | | |
| Điều kiện sau | Lưu dữ liệu trẻ vừa tạo xuống cơ sở dữ liệu và hiển thị trẻ vừa thêm trong danh sách trẻ em | | |
| Luồng sự kiện chính | | | |
| Tác nhân | | | Hệ thống |
| 1. Chọn “Trẻ em” trên thanh điều hướng dưới cùng (bottom navigation) | | |  |
|  | | | 2. Hiển thị danh sách trẻ em |
| 3. Chọn icon thêm mới trẻ trên thanh task bar | | |  |
|  | | | 4. Hiển thị activity “Tạo mới” |
| 5. Nhập thông tin trẻ | | |  |
| 6. Chọn “Kế tiếp” | | |  |
|  | | | 7. Hệ thống kiểm tra thông tin |
|  | | | 8. Lưu thông tin vừa tạo xuống db |
|  | | | 9. Phát sinh mũi tiêm trong cơ sở dữ liệu úng với trẻ em |
|  | | | 10. Kiểm tra ngày tiêm của các mũi tiêm vữa phát sinh cho trẻ |
|  | | | 11. Hiển thị thông báo notification |
| Luồng sự kiện phụ | | | |
| Tác nhân | | | Hệ thống |
| 5.1 Chọn icon quay lại trên thanh task bar | | |  |
|  | | | 6.1 Quay lại activity “home” |
| Luồng sự kiện ngoại lệ | | | |
|  | | 8.1 Hiển thị thông báo nếu sai.  8.2 Quay lại bước 5 | |
|  | | 11.1 Không hiển thị nếu thời gian không khớp | |

Bảng 3.4.4. UC011 – Thêm trẻ em

Biểu đồ activity



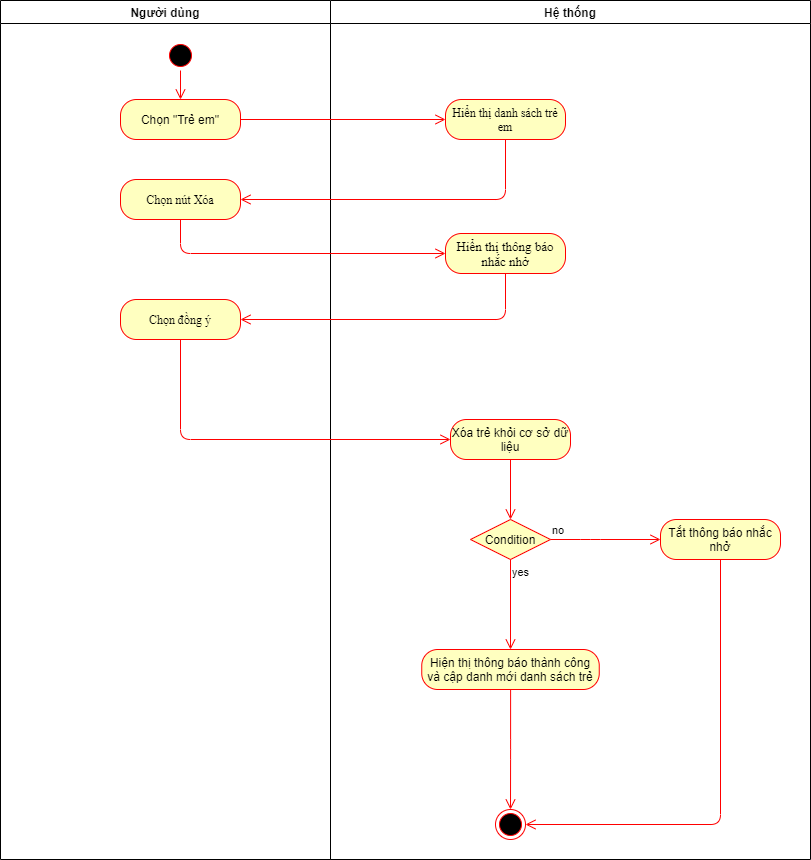
Hình 3.4.  Hình 3- : Activity Thêm trẻ em

#### UC012 – Xóa trẻ em

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC012 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng xóa thông tin trẻ em | |
| Điều kiện trước | Khởi động app và có trẻ em trong danh sách trẻ | |
| Điều kiện sau | Xóa dữ liệu trẻ em khỏi cơ sở dữ liệu và hiển thị thông báo và cập nhật danh sách | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “Trẻ em” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị danh sách trẻ em |
| 3. Chọn nút xóa mà muốn xóa trẻ trong danh sách | |  |
|  | | 4. Hiển thị thông báo nhắc nhở |
| 5. Chọn “Đồng ý” | |  |
|  | | 6. Xóa trẻ ra khỏi cơ sở dữ liệu |
|  | | 7. Hiển thị thông báo và cập nhật mới danh sách trẻ em |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 5.1 Chọn “Hủy” | | 6.1 Tắt thông báo nhắc nhở |
| Luồng sư kiện ngoại lệ | | |
|  | | 6.1 Xóa trẻ thất bại.  Tắt thông báo nhắc nhở và không cập nhật mới danh sách trẻ |

Bảng 3.4.4. UC012 – Xóa trẻ em

Biểu đồ activity



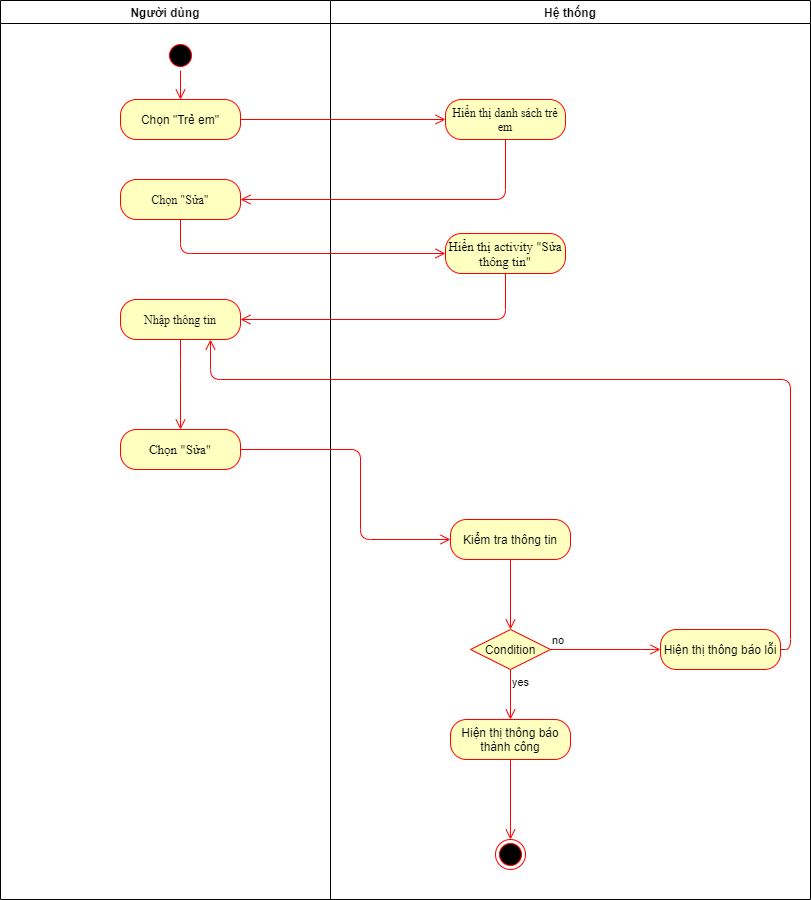
Hình 3.4.  Hình 3- Activity Xóa trẻ em

#### UC013 – Cập nhật thông tin trẻ em

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC013 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng cập nhật lại các thông tin cơ bản của trẻ nhỏ | |
| Điều kiện trước | Có dữ liệu của trẻ em cần cập nhật | |
| Điều kiện sau | Cập nhật dữ liệu thành công và hiển thị thông báo | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn “Trẻ em” trên thanh điều hướng dưới cùng (bottom navigation) | |  |
|  | | 2. Hiển thị danh sách trẻ em |
| 3. Chọn icon thêm mới trẻ trên thanh task bar | |  |
|  | | 4. Hiển thị activity “Tạo mới” |
| 5. Nhập thông tin trẻ | |  |
| 6. Chọn “Kế tiếp” | |  |
|  | | 7. Hệ thống kiểm tra thông tin |
|  | | 8. Lưu thông tin vừa tạo xuống db |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 5.1 Chọn icon quay lại trên thanh task bar | |  |
|  | | 6.1 Quay lại activity “home” |
| Luồng sự kiện ngoại lệ | |  |
|  | | 8.1 Hiển thị thông báo nếu sai.  8.2 Quay lại bước 5 |

Bảng 3.4.4. UC013 – Cập nhật thông tin trẻ em

Biểu đồ activity



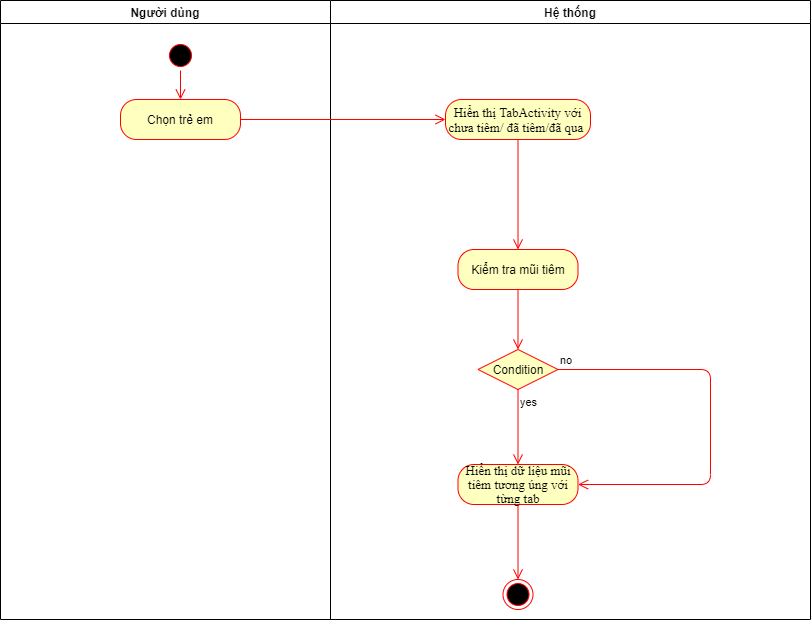
Hình 3.4. Activity Cập nhật thông tin trẻ em

#### UC014 – Xem lịch sử tiêm phòng

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC014 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng có thể xem được các mũi đã tiêm, sắp tiêm hoặc đã bỏ lỡ. Với mũi tiêm đã lỡ tính từ thời điểm hiện tại về trước, đã tiêm là các mũi đã hoàn thành và chưa tiêm là các mũi còn lại | |
| Điều kiện trước | Có trẻ em trong danh sách trẻ | |
| Điều kiện sau | Hiển thị Tabactivity với thông tin chưa tiêm/ đã tiêm/đã qua. Hiển thị dữ liệu các mũi tiêm tương úng với từng tab | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn trẻ em | |  |
|  | | 2. Hiển thị TabActivity với chưa tiêm/ đã tiêm/đã qua |
|  | | 3. Kiểm tra mũi tiêm (ngày tiêm và trạng thái) |
|  | | 3. Hiển thị dữ liệu mũi tiêm tương úng với từng tab |

Bảng 3.4.4. UC014 – Xem lịch sử tiêm phòng

Biểu đồ activity



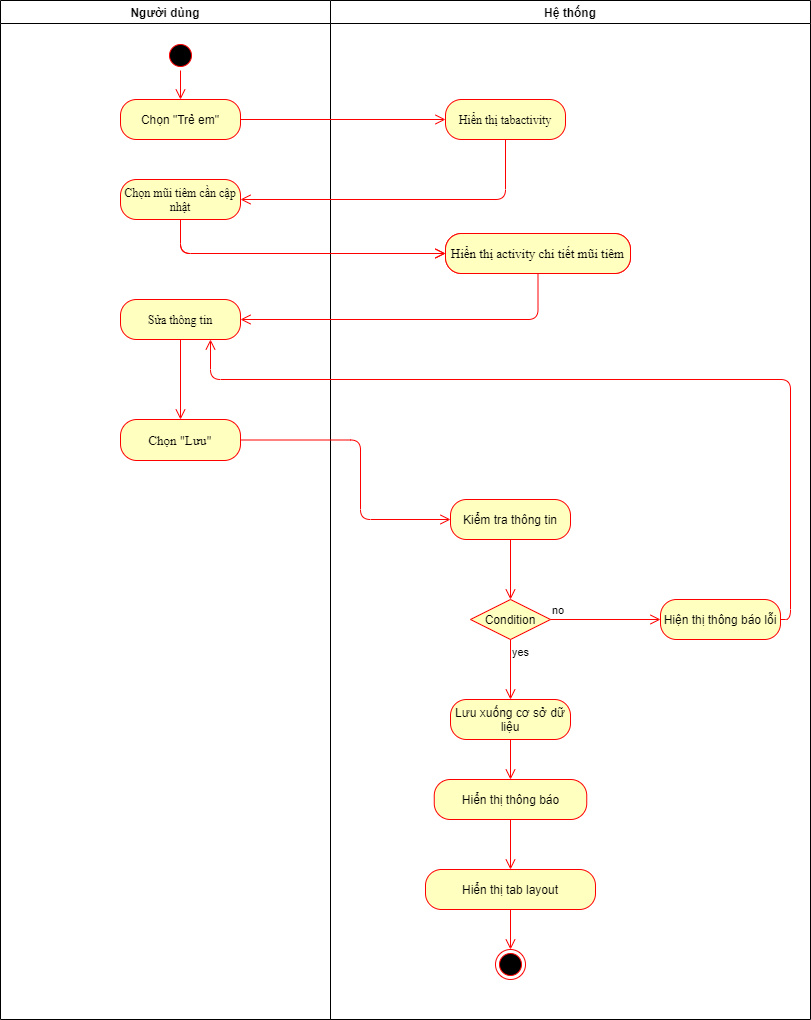
Hình 3.4. Activity Xem lịch sử tiêm phòng

#### UC015 – Cập nhật trạng thái phác đồ

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC015 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng cập nhật lại trạng thái mũi tiêm | |
| Điều kiện trước | Dữ liệu của trẻ muốn cập nhật phải có trong cơ sở dữ liệu | |
| Điều kiện sau | Cập nhật thành công trạng thái mũi tiêm | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn trẻ trong danh sách trẻ em | |  |
|  | | 2. Hiển thị TabLayout |
| 3. Chọn mũi tiêm cần cập nhật trong 3 tablayout | |  |
|  | | 4. Hiển thị activity thông tin chi tiết mũi tiêm |
| 5. Chỉnh sửa thông tin mũi tiêm | |  |
| 6. Chọn lưu | |  |
|  | | 7. Hệ thống kiểm tra |
|  | | 8. Thông báo thành công và lưu dữ liệu xuống cơ sở dữ liệu |
|  | | 9. Hiển thị TabLayout |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 6.1 Chọn Hủy | | 7.1. Hiển thị TabLayout |

Bảng 3.4.4. UC015 – Cập nhật trạng thái phát đồ

Biểu đồ activity



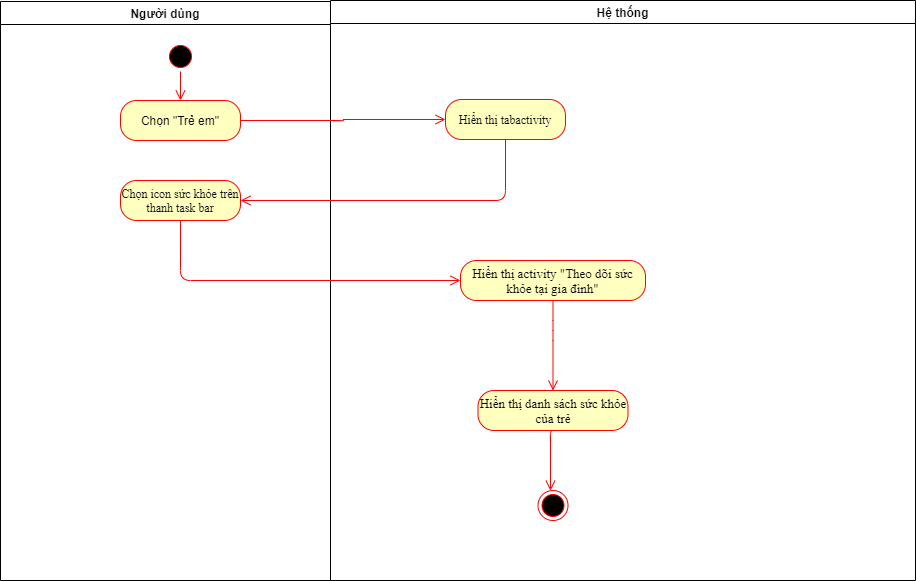
Hình 3.4. Activity Cập nhật trạng thái phát đồ

#### UC016 – Quản lý sức khỏe

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC016 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng quản lý các thông tin về sức khỏe của trẻ em | |
| Điều kiện trước | Cần có dữ liệu của trẻ trong cơ sở dữ liệu | |
| Điều kiện sau | Hiển thị danh sách sức khỏe của trẻ | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn trẻ cần sửa sức khỏe trong danh sách trẻ em | |  |
|  | | 2. Hiển thị TabLayout |
| 3. Chọn icon sức khỏe trên thanh taskbar | |  |
|  | | 4. Hiển thị activity “Theo dõi sức khỏe tại gia đình” |
|  | | 5. Hiển thị danh sách sức khỏe của trẻ |

Bảng 3.4.4. UC016 – Quản lý sức khỏe

Biểu đồ activity



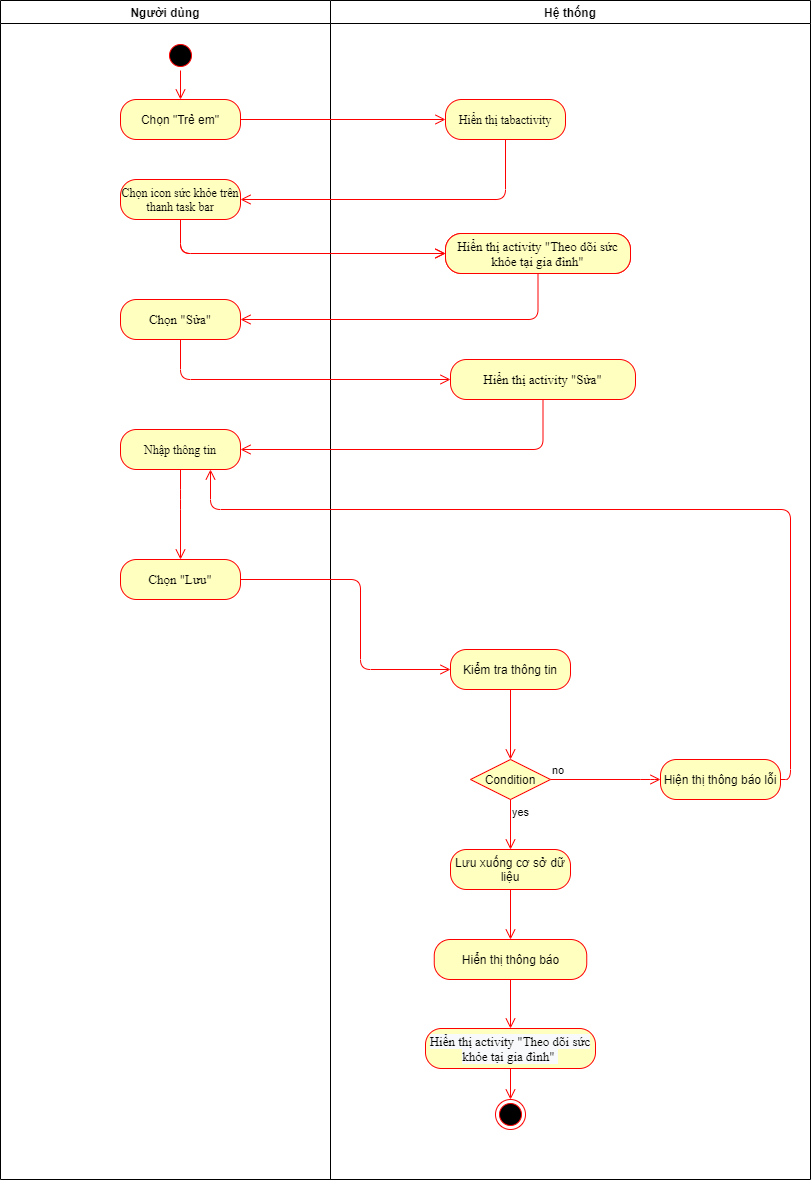
Hình 3.4. Activity xóa nhân viên

#### UC017 – Cập nhật sức khỏe

|  |  |  |  |
| --- | --- | --- | --- |
| Mã usecase | UC017 | | |
| Tác nhân | Người dùng | | |
| Mô tả | Người dùng cập nhật lại thông tin sức khỏe của trẻ em | | |
| Điều kiện trước | Cần có dữ liệu của trẻ trong cơ sở dữ liệu | | |
| Điều kiện sau | Cập nhật thành công sức khỏe của trẻ em trong cơ sở dữ liệu | | |
| Luồng sự kiện chính | | | |
| Tác nhân | | | Hệ thống |
| 1. Chọn trẻ cần thêm sức khỏe trong danh sách trẻ em | | |  |
|  | | | 2. Hiển thị TabLayout |
| 3. Chọn icon sức khỏe trên thanh taskbar | | |  |
|  | | | 4. Hiển thị activity “Theo dõi sức khỏe tại gia đình” |
| 5. Chọn Sửa | | |  |
|  | | | 6. Hiển thị activity Sửa |
| 7. Nhập thông tin sửa | | |  |
| 8. Chọn “Lưu” | | |  |
|  | | | 9. Hệ thống kiểm tra dữ liệu |
|  | | | 10. Hiển thị thông báo và cập nhật sức khỏe xuống cơ sở dữ liệu |
|  | | | 11. Hiển thị activity “Theo dõi sức khỏe tại gia đình” |
|  | | | 12. Cập nhật mới danh sách sức khỏe trẻ em |
| Luồng sự kiện phụ | | | |
| Tác nhân | | | Hệ thống |
| 8.1 Chọn icon quay lại trên thanh task bar | | | 9.1 Hiển thị activity “Theo dõi sức khỏe tại gia đình” |
| Luồng sư kiện ngoại lệ | | | |
|  | | 9.1 Nếu dữ liệu sai quay về bước 7 và hiển thị thông báo lỗi | |

Bảng 3.4.4. UC017 – Cập nhật sức khỏe

Biểu đồ activity



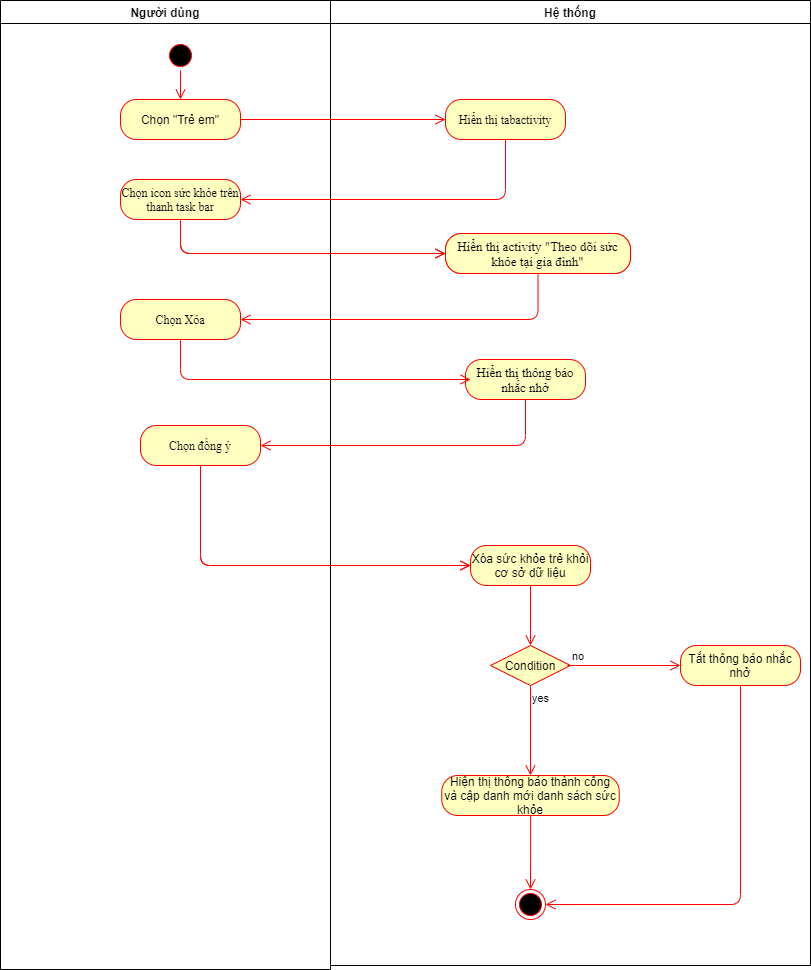
Hình 3.4. Cập nhật sức khỏe

#### UC018 – Xóa sức khỏe

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC018 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng xóa thông tin sức khỏe khỏi cơ sở dữ liệu | |
| Điều kiện trước | Có thông tin sức khỏe cần xóa | |
| Điều kiện sau | Xóa sức khỏe khỏi cơ sở dữ liệu và cập nhật lại dnah sách trên màn hình hiển thị | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn trẻ cần thêm sức khỏe trong danh sách trẻ em | |  |
|  | | 2. Hiển thị TabLayout |
| 3. Chọn icon sức khỏe trên thanh taskbar | |  |
|  | | 4. Hiển thị activity “Theo dõi sức khỏe tại gia đình” |
| 5. Chọn Xóa | |  |
|  | | 6. Hiển thị thông báo nhắc nhở |
| 7. Chọn “Đồng ý” | |  |
|  | | 8. Xóa sức khỏe trẻ em vừa chọn ra khỏi cơ sở dữ liệu |
|  | | 9. Hiển thị thông báo và cập nhật mới danh sách sức khỏe trẻ em |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 7.1 Chọn “Hủy” | | 8.1 Tắt thông báo nhắc nhở |
| Luồng sư kiện ngoại lệ | | |
|  | | 8.1 Xóa trẻ thất bại.  Tắt thông báo nhắc nhở và không cập nhật mới danh sách trẻ |

Bảng 3.4.4. UC018 – Xóa sức khỏe

Biểu đồ activity



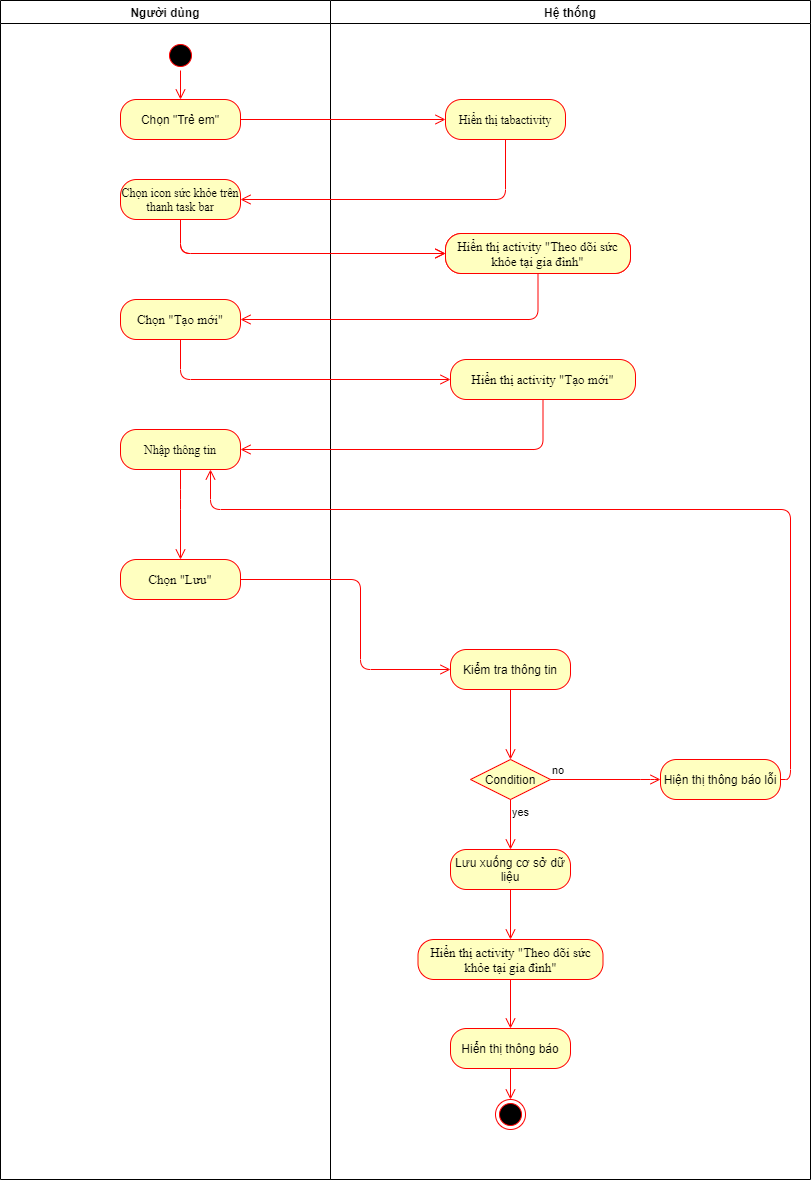
Hình 3.4. Xóa sức khỏe

#### UC019 – Thêm sức khỏe

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC018 | |
| Tác nhân | Người dùng | |
| Mô tả | Người dùng thêm mới sức khỏe của trẻ em | |
| Điều kiện trước | Phải có trẻ em trước đó | |
| Điều kiện sau | Thêm sức khỏe thành công và hiển thị trong danh sách | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn trẻ cần thêm sức khỏe trong danh sách trẻ em | |  |
|  | | 2. Hiển thị TabLayout |
| 3. Chọn icon sức khỏe trên thanh taskbar | |  |
|  | | 4. Hiển thị activity “Theo dõi sức khỏe tại gia đình” |
| 5. Chọn nút “Tạo mới” | |  |
|  | | 6. Hiển thị activity “Tạo mới sức khỏe” |
| 7. Nhập thông tin | |  |
| 8. Chọn Lưu | |  |
|  | | 9. Hệ thống kiểm tra |
|  | | 10. Lưu thông tin xuống cơ sở dữ liệu |
|  | | 11. Quay lại activity “Theo dõi sức khỏe tại gia đình” |
|  | | 12. Hiển thị sức khỏe mới trong danh sách và thông báo thành công |
| Luồng sự kiện phụ | | |
| Tác nhân | | Hệ thống |
| 5. Chọn nút quay lại trên thanh taskbar | |  |
|  | | 6. Quay lại activity “Theo dõi sức khỏe tại gia đình” |
| Luồng sự kiện ngoại lệ | | |
|  | | 9.1 Hiển thị thông báo nếu sai thông tin  9.2 quay lại bước 5 |

Bảng 3.4.4. UC019 – Thêm sức khỏe

Biểu đồ activity



Hình 3.4. Thêm sức khỏe

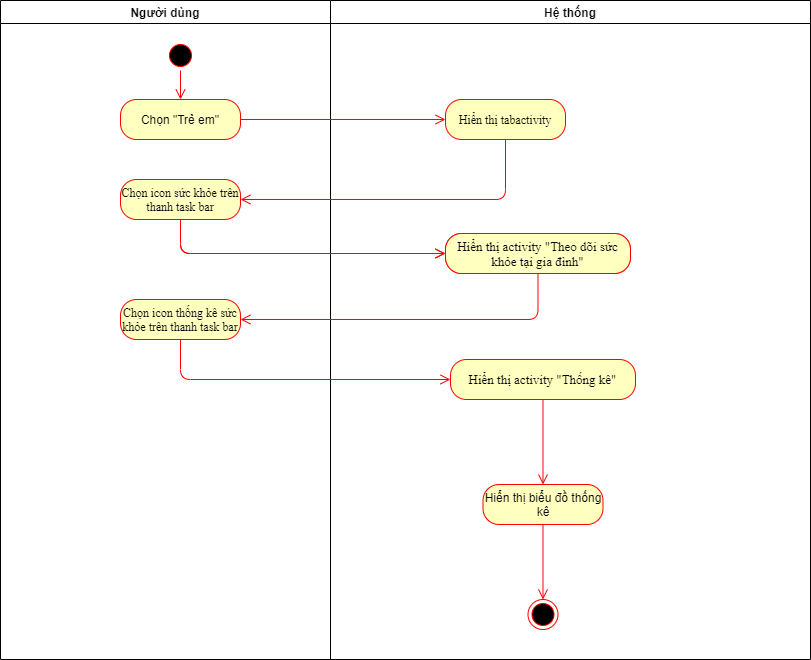
#### UC020 – Xem thống kê sức khỏe

Mô tả usecase UC020

|  |  |  |
| --- | --- | --- |
| Mã usecase | UC020 | |
| Tác nhân | Người dùng | |
| Mô tả | Xem được thống kê sức khỏe của bé qua các tuần | |
| Điều kiện trước | Có dữ liệu sức khỏe của bé | |
| Điều kiện sau | Hiển thị biểu đồ thống kê | |
| Luồng sự kiện chính | | |
| Tác nhân | | Hệ thống |
| 1. Chọn trẻ cần thêm sức khỏe trong danh sách trẻ em | |  |
|  | | 2. Hiển thị TabLayout |
| 3. Chọn icon sức khỏe trên thanh taskbar | |  |
|  | | 4. Hiển thị activity “Theo dõi sức khỏe tại gia đình” |
| 5. Chọn icon thống kê sức khỏe trên thanh taskbar | |  |
|  | | 6. Hiển thị activity “thống kê” |
|  | | 7. Hiển thị biểu đồ thống kê |

Bảng 3.4.4. UC20 – Xem thống kê sức khỏe

Biểu đồ activity



Hình 3.4. Xem thống kê sức khỏe

## Sơ đồ Class

### Danh sách các đối tượng

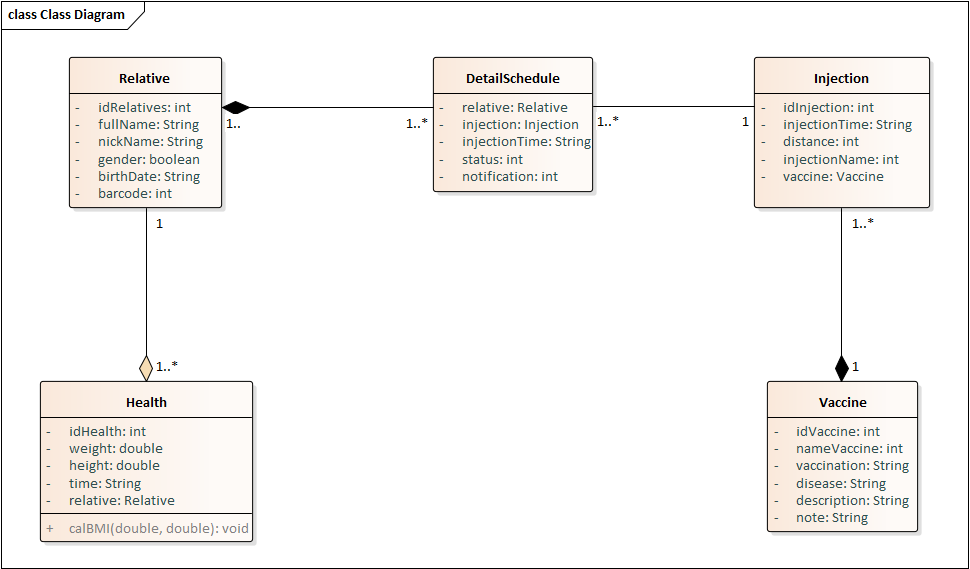
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp/quan hệ** | **Ý nghĩa/Ghi chú** |
| 1 | Relative | Mô tả các thuộc tính và các phương thức liên quan tới nghiệp vụ quản lý trẻ em |
| 2 | Health | Mô tả các thuộc tính và các phương thức liên quan tới nghiệp vụ quản lý sức khỏe của trẻ em |
| 3 | DetailSchedule | Mô tả các thuộc tính và các phương thức liên quan tới nghiệp vụ quản lý chi tiết các mũi tiêm của trẻ em. |
| 4 | Injection | Mô tả các thuộc tính và các phương thức liên quan tới nghiệp vụ quản lý mũi tiêm phòng. |
| 5 | Vaccine | Mô tả các thuộc tính và các phương thức liên quan tới nghiệp vụ quản lý vaccine |
| 6 | NotiicationTask | Mô tả các thuộc tính và phương thức liên quan đến việc báo thức của ứng dụng |

Bảng 3. Danh sách các lớp đối tượng

### Mô hình hóa các lớp đối tượng

Mỗi một vaccine có từ một đến nhiều mũi tiêm (injection). Với mỗi mũi tiêm trong cùng một vaccine, khoảng cách giữa mũi này với mũi trước đó từ 29~31 ngày (1 tháng). Mỗi trẻ em (Relative) có nhiều mũi tiêm (Injection). Một mũi tiêm được thực hiện bởi nhiều trẻ. Điều này phát sinh ra chi tiết lịch tiêm chủng (DetailSchedule). Lịch tiêm chủng chứa mã trẻ và mã mũi tiêm, trạng thái thực hiện, thời gian thực hiện. Không chỉ vậy còn phát sinh ra trạng thái thông báo (notification), nó giúp ta xác định được mũi tiêm đã được thông báo rồi hay chưa. Sức khỏe được phát sinh bởi trẻ em, trẻ em sẽ có nhiều khoảng thời gian phát triển, khác nhau kéo theo đó là nhiều sức khỏe khác nhau không cùng một thời điểm.

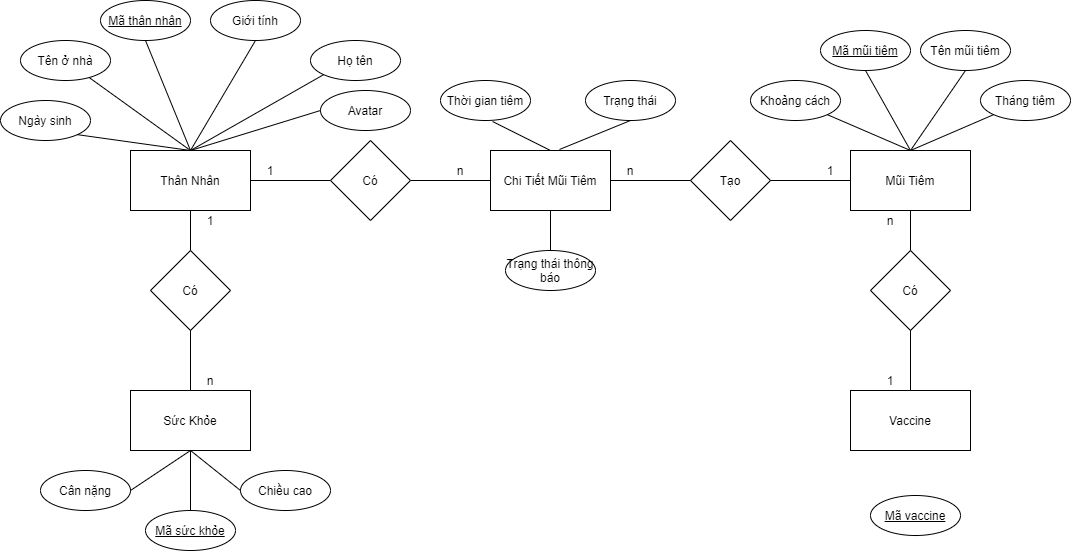
Vì thế, sơ đồ lớp đối tượng được hình thành như sau:



Hình 3.4. Sơ đồ lớp đối tượng

### Mô hình dữ liệu quan hệ

Từ phân tích nghiệp vụ và sơ đồ lớp đối tượng, mô hình dữ liệu quan hệ được mô tả như hình sau:



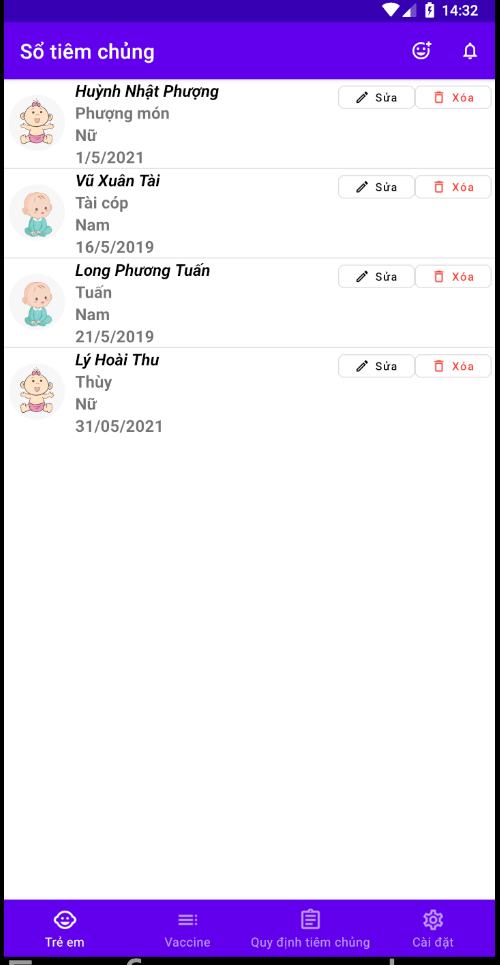
Hình 3.4. Mô hình dữ liệu quan hệ

Thân nhân sẽ có từ nhiều chi tiết mũi tiêm, chi tiết mũi tiêm được tạo từ các mũi tiêm. Mỗi vaccine sẽ có nhiều mũi tiêm. Mỗi thân nhân có nhiều sức khỏe

# CHƯƠNG 4: HIỆN THỰC

## 4.1. Giao diện Sổ Tiêm Chủng:

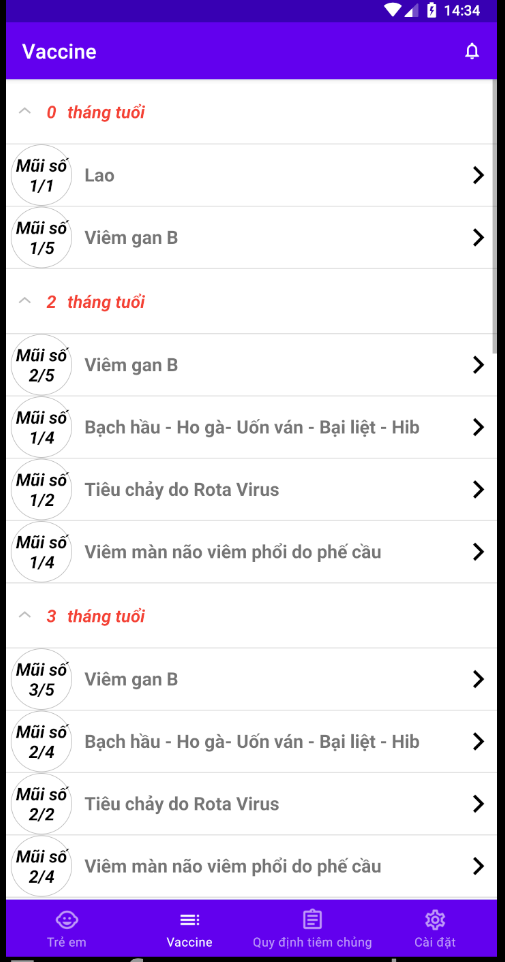
Màn hình này thể hiện danh sách các trẻ em hiện có trong gia đình. Sau khi thêm mới trẻ em danh sách này tự động cập nhật và hiển thị lại. Trong từng item có 2 nút xóa và sửa. Khi người dùng muốn xóa thì chọn nút xóa. Với người dùng chọn nút sửa sẽ hiển thị sang màn hình sửa thông tin trẻ em (chi tiết thông qua ứng dụng). Trên thanh công cụ (task bar) gồm có icon thêm và icon thông báo. Khi chọn icon add child hiển thị sang màn hình tạo mới với mục đích thêm mới thông tin trẻ em. Bên dưới là thanh diều hướng của ứng dụng.



Hình 4. Sổ tiêm chủng

## 4.2 Giao diện Danh sách vaccine

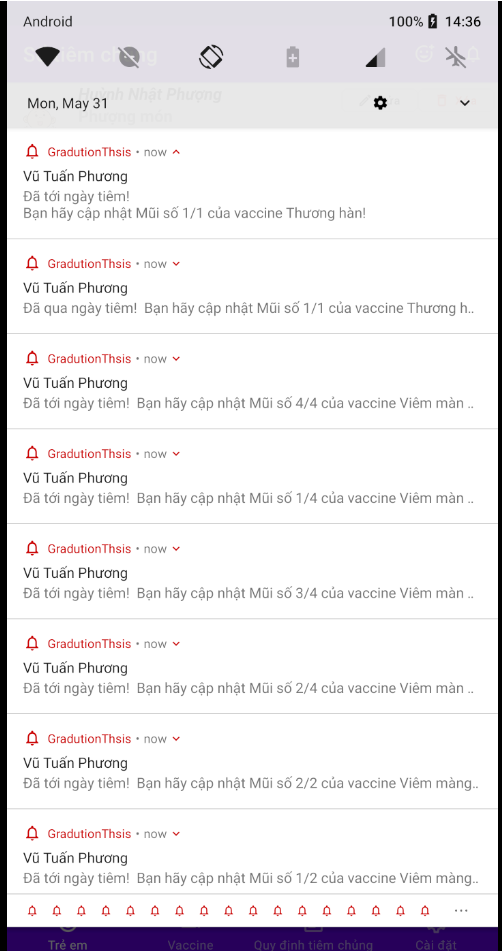
Giao diện này hiển thị phác đồ điều trị chung áp dụng cho tất cả các trẻ em. Khi chọn vào item con sẽ chuyển sang màn hình chi tiết về vacxin. Màn hình chi tiết vacxin giúp người dùng có thêm kiến thức về vacxin (chi tiết thông qua ứng dụng).



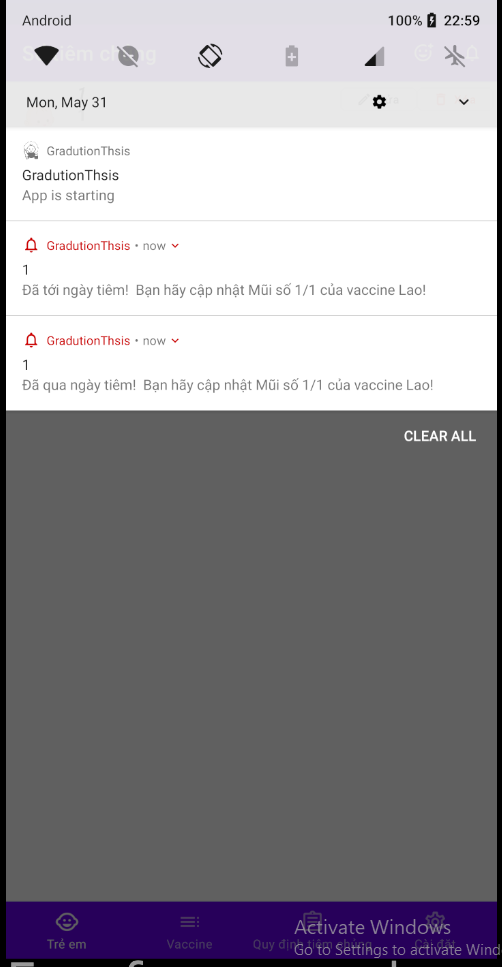
Hình 4. Danh sách vaccine

## 4.3 Thông báo gửi đến

Hệ thống sẽ kiểm tra và gửi thông báo từ ứng đến điện thoại thông qua alarm service và hiển thị lên màn hình Notification. Các thông được gửi đến với nội dung là: title là tên của trẻ em, message là nội dung tin nhắn gửi đến. Ở đây sử dụng một service chạy ngầm giúp ứng dụng theo dõi chính xác giờ giấc để nhắc tiêm chủng đúng lúc. Thông báo này là thông báo mũi tiêm của trẻ



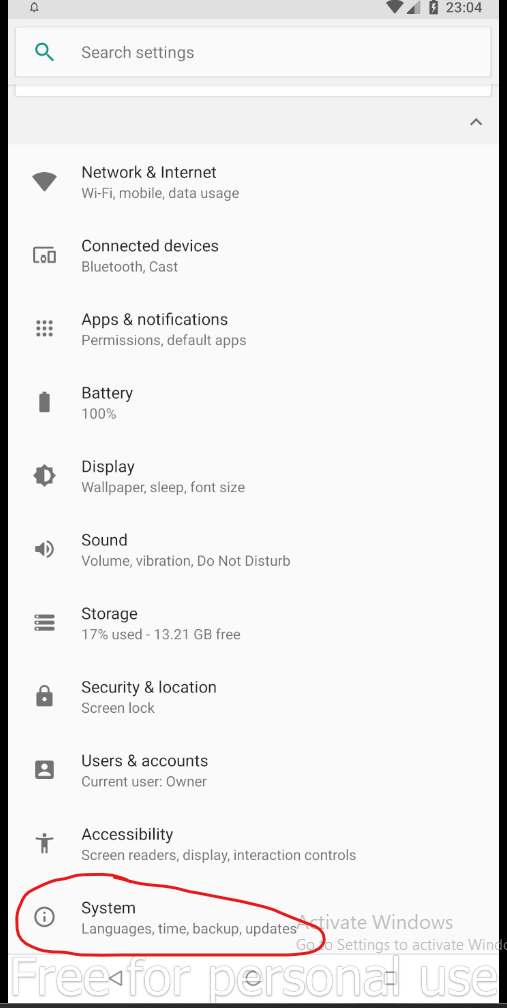
Hình 4. Thông báo gửi đến



Hình 4. 4 Thông báo gửi đến

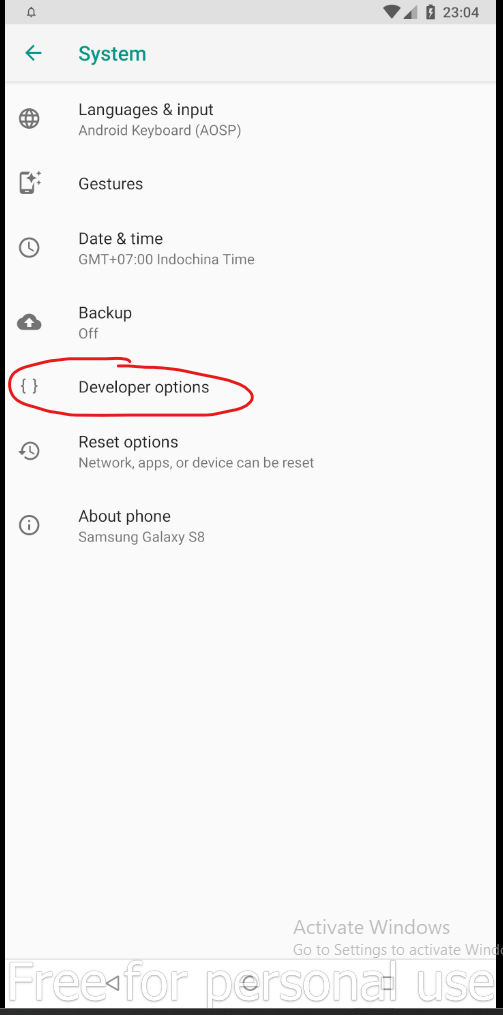
Ơ đây có 2 loại thông báo, thông báo gắn cố định trên cùng dịnh dạng form đen chính là foreground services của chúng ta. Điều này chứng tỏ app chúng ta vẫn đang được chạy nhằm hỗ thông báo chính xác thời điểm tiêm đến người dùng. Chúng ta sẽ kiểm tra foreground service như sau:

Chọn system trong cài đặt của thiết bị



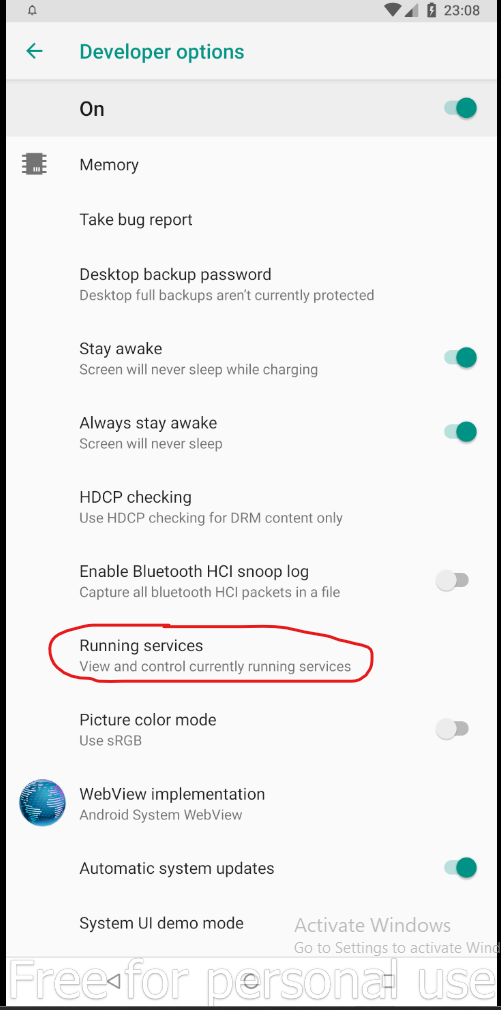
Hình 4.5. 1 Màn hình cài đặt của thiết bị

Chọn Developer options:



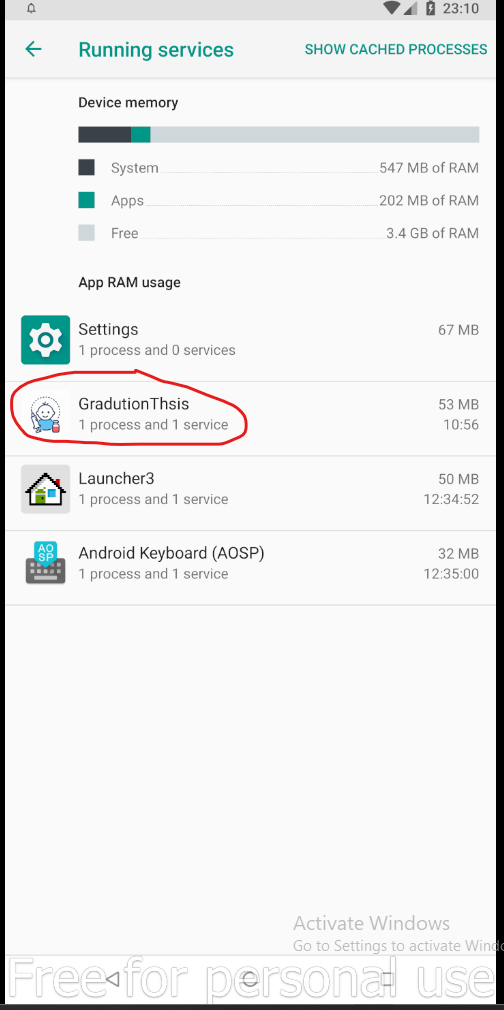
Hình 4.5. 2 Màn hình System

Sau đó chọn running service



Hình 4.5. 3 Màn hình Developer options

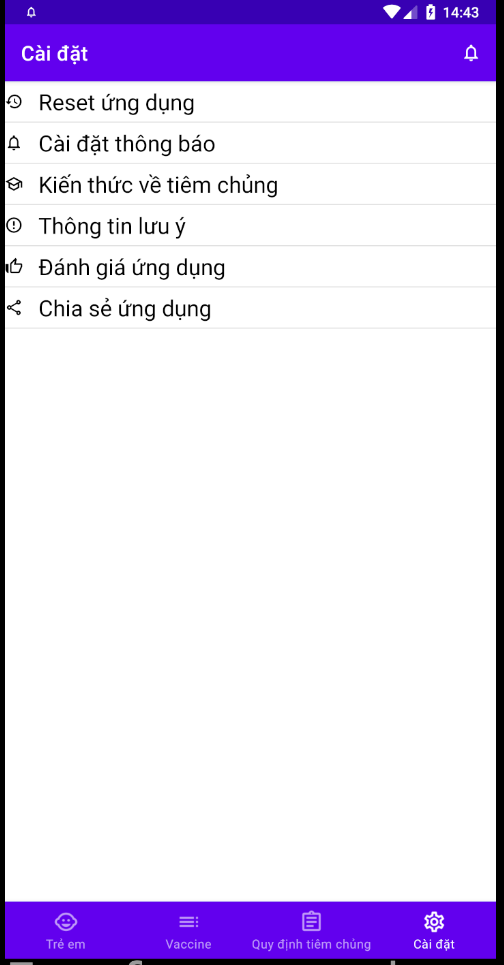
Hệ thống hiển thị các service của các app đang chạy trên thiết bị, và ta kiểm tra được ứng dụng có thực sự đang được chạy ngầm



Hình 4.5. 4 Màn hình Running service

## 4.4 Giao diện Cài Đặt

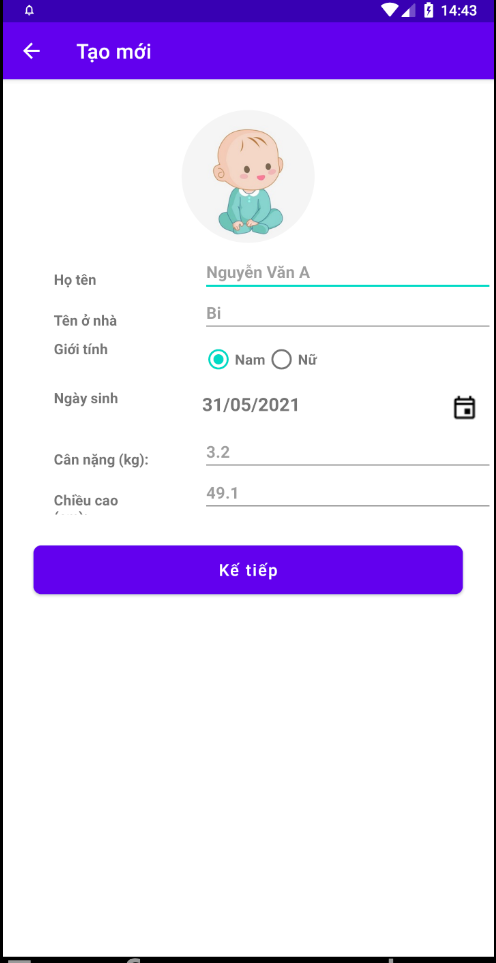
Màn hình này hỗ trợ trong việc dặt lại ứng trong trường hợp gặp lỗi (có thể về dữ cũng có thể vì một số lý do chưa xác định)



Hình 4. Màn hình cài đặt

## 4.5 Giao diện Thêm trẻ em

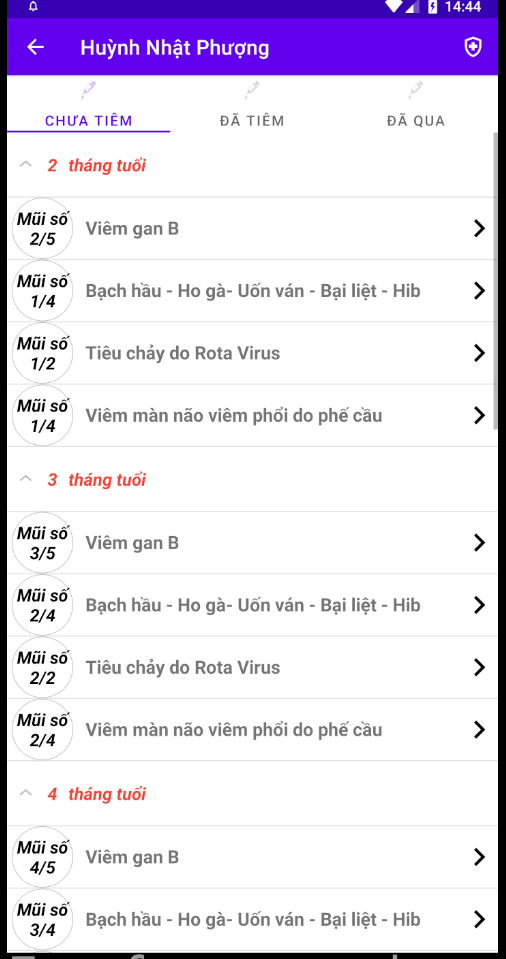
Màn hình này có chức năng tạo mới thông tin trẻ em. Khi chọn lưu, thông tin vừa tạo sẽ lưu lại trong cơ sở dữ liệu.



Hình 4. Thêm trẻ em

## 4.6 Giao diện Phác đồ cá nhân của trẻ

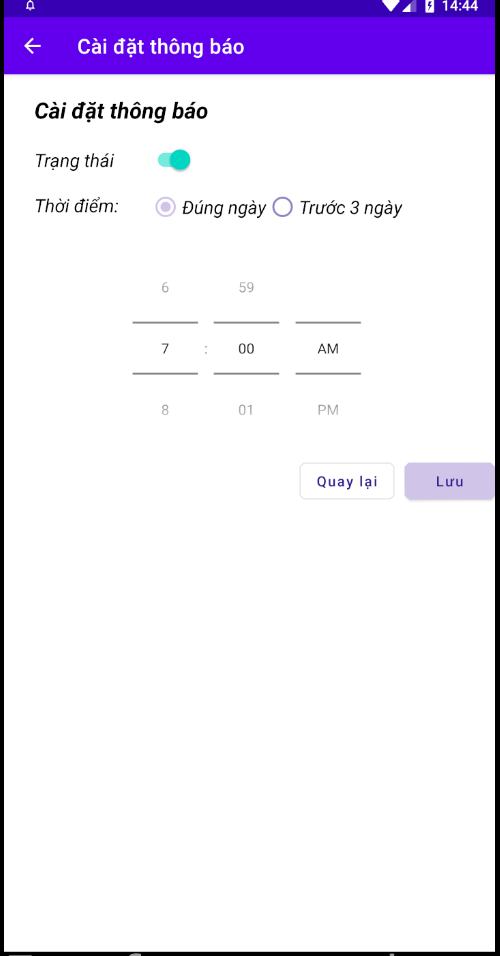
Giao diện này với mục đích hiển thị phác đồ tương úng với từng trẻ. Thông qua 3 tab: chưa tiêm, đã tiêm, đã qua



Hình 4. Phác đồ điều trị cá nhân

## 4.7 Giao diện Cài Đặt Thông Báo

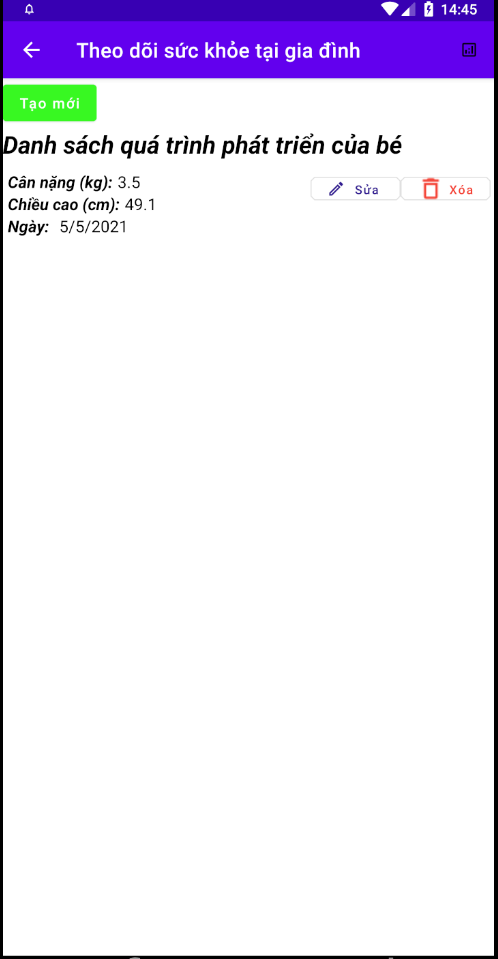
Màn hình này giúp người dùng quản lý được thời gian thông báo cho các mũi tiêm. Người dùng tự mình cài đặt thời gian báo thức cho chính họ.



Hình 4. Cài đặt thông báo

## 4.8 Giao diện Theo Dõi Sức Khỏe Tại Nhà

Màn hình này dùng để theo dõi tình hình sức khỏe của trẻ em. Khi chọn icon biểu đồ thống kê được trên thanh task bar. Chuyển sang màn hình thống kê, hệ thống dự trên bộ dữ liệu về sức khỏe của trẻ mà tạo nên biểu đồ (thống kế qua từng tháng) – chi tiết xem tại ứng dụng.



Hình 4. Theo dõi sức khỏe tại nhà

# CHƯƠNG 5: KẾT LUẬN

### Kết quả đạt được

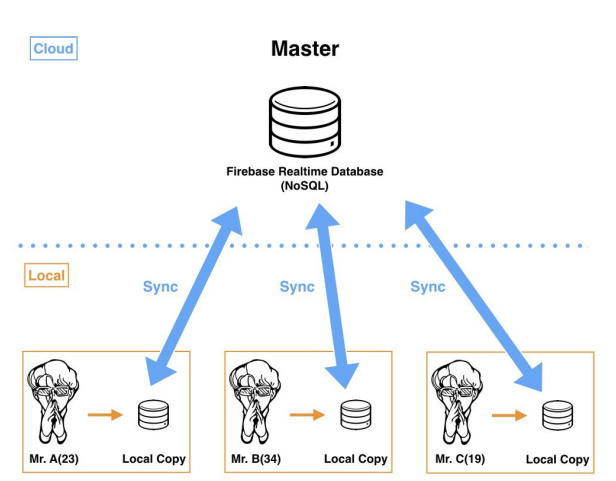
Các chức năng chính trong use case gần như đã được thực hiện gần như là đầy đủ. Ứng dụng áp dụng một số thuật toán như sắp xếp, tìm kiếm,....

Thông qua ứng dụng, nhóm đã hiểu được cách hoạt động của phần mềm trên di động, thấy được sự tiện ích trên thiết bị di động, cũng như hiểu rõ được cách làm một phần mềm trên di động. Không chỉ áp dụng các công cụ mà nhóm còn sử dụng các thuật toán như: sắp sắp CollectionSort(), cách kiểm tra thời gian trong dữ liệu với ngày thực tế, kiểm tra ràng buộc giữa các mũi tiêm trong cùng 1 vaccine.

Bên cạnh đó ứng dụng còn có một số ít hạn chế như vài chức năng chưa được hoàn thiện: đánh giá ứng dụng, chia sẻ ứng dụng thông qua quét mã QR Code,... Bộ dữ liệu chưa được tối ưu hóa vaccine và các mũi tiêm, vẫn đang trong mức tổng quát chưa thiết kế chi tiết. Việc xử lý các ràng buộc về dữ liệu còn rườm rà, chưa tối ưu hóa.Dữ liệu chưa có tính bảo mật thông tin cao. Ứng dụng còn chưa thân thiện với người dùng, chưa thực sự là dễ sử dụng.

### Hướng phát triển

Sau khi thực hiện đồ án tại trường, ứng dụng sẽ được mở rộng them. Cho phép ứng dụng không chỉ dùng offline mà còn được kết nối trực tuyến. Chuyển sang sử dụng CSDL là firebase database nhằm mục đích quản lý, lưu lại dữ liệu của người dùng. Vì app hiện tại đang sử dụng dữ liệu là bộ nhớ của máy, nên khi gỡ cài đặt ứng dụng dữ liệu sẽ tự động mất theo. Tại sao lại chọn lưu trữ dữ liệu bằng firebase? Firebase realtime database lưu trữ dữ liệu theo định dạng JSON, nó phù hợp với cách xử lý dữ liệu mà ứng dụng đang sử dụng. Với khi sử dụng Firebase database ta có được lợi thế đó là real time, khi dữ liệu thay đổi, mọi thiết bị kết nối sẽ nhận được trong vài mili giây. Dữ liệu lưu trữ ở dạng local trong khi offline và thực hiện đồng bộ khi được kết nối. Vậy nên khi sử dụng Firebase realtime database tiện lợi trong việc phát triển/mở rộng ứng dụng (gắn thêm) mà không cần chỉnh sửa quá nhiều về mã nguồn.



Hình 5. Hình ảnh mô tả Firebase Realtime Database

Điều này giúp ứng dụng đưa ra thêm hướng phát triển đó là việc đồng bộ và chia sẻ dữ với nhiều máy khác nhau thông qua quét mã QR Code. Trong một gia đình không chỉ nhắc nhở một cá nhân mà còn đồng bộ hóa dữ liệu cho hộ gia đình, những người liên quan có thể sử dụng chung dữ liệu trẻ em với nhau. Việc làm này không chỉ nhắc cá nhân mà nhắc nhở cho tập thể gia đình, để mọi người trong gia đình nắm được tình hình tiêm chủng của trẻ em nhà họ.

Không chỉ kết nối giữa những người trong cùng gia đình. Mà ứng dụng mở rộng cho việc kết nối với các cơ sở tiêm chủng, hỗ trợ người dùng tương tác trực tiếp đến các bác sĩ có trình độ chuyên môn. Không giới hạn ứng dụng chỉ quản lý về việc nhắc nhở tiêm phòng mà còn giúp người dùng giải quyết được vấn đề chưa từng gặp phải trong tiêm chủng thông qua tư vấn, giúp đỡ của của bác sĩ tại một thời điểm xác định.

Quản lý thông báo như các ứng dụng mạng xã hội (facebook, weibo,.. ) để người dùng dễ dàng theo dõi dòng sự kiện thông báo xảy ra của ứng dụng.

Chỉnh sửa lại giao diện cho đẹp mắt và hợp lý hơn cho người dùng.

# TÀI LIỆU THAM KHẢO

Các tài liệu tiếng việt

[1] Lê Hoàng Sơn, Nguyễn Hữu Thọ (2018) Giáo trình lập trình android

[2] Giáo trình lập trình Android - ĐH Khoa Học Tự Nhiên HCM

[3] Thạc Bình Cường, Phân tích và thiết kế hệ thống thông tin, nhà xuất bản Khoa học và Kỹ thuật, 2002.

Các tài liệu từ internet

[4] “ImageView”: <https://developer.android.com/reference/android/widget/ImageView>

[5]”Intend”: <https://developer.android.com/reference/android/content/Intent>

[6] “SQLite”: <https://developer.android.com/training/data-storage/sqlite>

[7]”Install android studio”: <https://developer.android.com/studio/install>

[8]”AndroidManifest”: <https://developer.android.com/guide/topics/manifest/manifest-intro>

[9]”SDK”: <https://developer.android.com/studio/releases/platforms>

[10]”android 9”: <https://developer.android.com/about/versions/pie/android-9.0>

[11]”Application framework”: <https://developer.android.com/guide/components/fundamentals>

[12]”Button”: <https://developer.android.com/reference/android/widget/Button>

[13]”ContextMenu”: <https://developer.android.com/reference/android/view/ContextMenu.html?hl=pt-br>

[14]: “Activity”: <https://developer.android.com/reference/android/app/Activity>

[15]: “Alarm management”: <https://vncoder.vn/bai-hoc/alarmmanager-trong-android-123>

[16]: “Notification” : <https://developer.android.com/guide/topics/ui/notifiers/notifications>

[17]: “Broadcast Reciever”:

[18]: “Foreground services”: <https://developer.android.com/guide/components/foreground-services>

[19]: Service in android: https://developer.android.com/guide/components/services

# PHỤ LỤC

Từ kết khảo sát thực tế + nguồn thông tin tìm kiếm để ràng buộc chính xácdữ liệu được chia ra làm 2 loại vaccine với mũi tiêm (injection)

Dữ liệu vaccines.json: file này là nguồn dữ liệu này chứa thông tin của vaccice gồm:

* Tên vaccine (vaccineName):
* Căn bệnh (disease): field này mô tả căn bệnh, mức độ nguy hiểm, để đưa ra lý do tiêm chủng cho bệnh.
* Tiêm chủng (vaccination): field này mô tả tên bệnh
* Mô tả(desscription): mô tả lịch tiêm của vaccine
* Ghi chú/chú ý (note):

Sau đây là mô tả một phần dữ liệu của vaccines.json (thông tin chi tiết xem tại vaccines.json):

[  
 {  
 "vaccination": "Lao",  
 "vaccineName": "Vaccine BCG (Việt Nam)",  
 "disease": "\tMỗi năm trên thế giới có hơn 1 triệu người tử vong do bệnh Lao. Căn bệnh này được xếp vào top 10 căn bệnh truyền nhiễm nguy hiểm và gây tử vong nhiều nhất hiện nay.\n\tBệnh Lao phổi thường gặp nhất với những triệu chứng như ho liên tục kéo dài không khỏi, sốt cao, đau ngực, khó thở… Bệnh nhân nếu như không được phát hiện và điều trị điều trị kịp thời dễ gặp các biến chứng đe dọa trực tiếp đến sức khỏe cũng như là tính mạng như: ho ra máu, tràn khí màng phổi, suy hô hấp, giãn phế quản, u nấm phổi…",  
 "description": "Lịch tiêm chủng:\n\t• Vắc xin BCG được chỉ định tiêm cho trẻ trong vòng 30 ngày sau khi sinh và càng sớm càng tốt.\n\t• Vắc xin BCG chỉ tiêm một liều duy nhất, không cần tiêm nhắc lại.",  
 "note": "Tác dụng không mong muốn:\n\t• Phần lớn trẻ em đều có phản ứng tại vết tiêm, thông thường ngay sau khi tiêm BCG sẽ xuất hiện một nốt nhỏ tại chỗ tiêm và biến mất sau khoảng 30 phút. Sau khoảng 2 tuần sẽ xuất hiện một vết loét đỏ có kích thước bằng đầu bút chì. Sau đó 2 tuần vết loét tự lành tạo nên một vết sẹo nhỏ có đường kính 5mm.\nCác phản ứng có thể gặp như áp xe, nổi hạch nách thường do sử dụng bơm tiêm không vô trùng, tiêm quá nhiều hay tiêm không đúng kỹ thuật.\n\t• Các phản ứng nặng khi tiêm BCG thường rất ít gặp, chỉ có 1/1.000.000 người bị nhiễm lao sau khi tiêm BCG, hay xảy ra ở những trường hợp nhiễm HIV hoặc suy giảm miễn dịch mắc phải."  
 },...

]

Dữ liệu injections.json: file này là nguồn dữ liệu chứa thông tin của mũi tiêm tương ứng với từng vaccice gồm:

* Tháng tiêm (injectionMonth): trường này mô tả cho tháng tiêm
* Tên mũi tiêm (injectionName): field này mô tả căn bệnh, mức độ nguy hiểm, để đưa ra lý do tiêm chủng cho bệnh.
* Khoảng cách (vaccination): field này mô tả tên bệnh
* Mã vaccine (idVaccine): trường này là id vaccine. Vì vaccine không trùng nhau nên khi tạo mã vaccine(primary key) trong SQLite sử dụng bằng cách khai báo tăng tự động nên trường id này gần như là chính xác. Mã tăng tự động bắt đầu từ 1

Sau đây là mô tả một phần dữ liệu của vaccines.json (thông tin chi tiết xem tại injections.json):

[  
 {  
 "injectionMonth": 0,  
 "injectionName": "Mũi số 1/1",  
 "distance": 0,  
 "idVaccine": "1"  
 },  
 {  
 "injectionMonth": 0,  
 "injectionName": "Mũi số 1/5",  
 "distance": 0,  
 "idVaccine": "2"  
 },  
 {  
 "injectionMonth": 1,  
 "injectionName": "Mũi số 2/5",  
 "distance": 1,  
 "idVaccine": "2"  
 },...

]

# KẾ HOẠCH THỰC HIỆN

**Tên đồ án**: Xây dựng phần mềm quản lý nhắc nhở tiêm phòng cho trẻ nhỏ

**Giảng viên hướng dẫn**: Huỳnh Hữu Nghĩa

**Thành viên nhóm**: Nguyễn Thanh Tường – 16028491

Nguyễn Nhật Trường - 16026511

1. Kế hoạch công việc (15 tuần)

|  |  |  |
| --- | --- | --- |
| **Thời gian** | **Nội dung công việc làm theo tuần** | **Ghi chú** |
| Tuần 1 | * Tìm hiểu nghiệp vụ * Tìm hiểu về quy trình TDD * Lập bản kế hoạch công việc cần thực hiên. * Tìm hiểu công nghệ liên quan để hiện thực đồ án | * Giai đoạn phân tích yêu cầu |
| Tuần 2 | * Phân tích yêu cầu( thiết kế Use case – mức tổng quát) * Viết tài liệu báo cáo đồ án * Tìm hiểu và đưa ra các service thích hợp cho từng giai đoạn | * Giai đoạn thiết kế hệ thống |
| Tuần 3 | * Phân tích yêu cầu (Xem lại mô hình lớp và mô hình cơ sở dữ liệu) * Thiết kế mockup * Cập nhập tài liệu báo cáo | * Giai đoạn thiết kế hệ thống |
| Tuần 4 | * Phân tích yêu cầu (thiết kế mô hình hoạt động, thiết kế mockup giao diện ) * Thiết kế test case * Đưa ra coding convention * Cập nhập file báo cáo | * Giai đoạn thiết kế hệ thống |
| Tuần 5 | * Thực thi chương trình (code) – xử lý giao diện * Cập nhập file báo cáo * Tìm hiểu về tạo chat head bubble * Tạo bộ dữ liệu vaccine | * Gai đoạn thực hiện và thử nghiệm |
| Tuần 6 | * Thực thi chương trình (code) * Cập nhập file báo cáo * Tìm hiểu về Android MPChart (thư viện vẽ biểu đồ) * Tìm hiểu về voice search | * Gai đoạn thực hiện và thử nghiệm |
| Tuần 7 | * Thực thi chương trình (code) * Cập nhập file báo cáo * Tìm hiểu về SQLite * Tìm hiểu về content provider | * Gai đoạn thực hiện và thử nghiệm * Update bộ dữ liệu vaccine, tạo bộ dữ liệu mũi tiêm |
| Tuần 8 | * Thực thi chương trình (code) - Hiện thực hóa chức năng * Cập nhập file báo cáo Thực hiện kết nối giao diện với database | * Gai đoạn thực hiện và thử nghiệm * Update bộ dữ liệu vaccine |
| Tuần 9 | * Thực thi chương trình (code) * Cập nhật file báo cáo * Tìm hiểu về Lottie Animations | * Gai đoạn thực hiện và thử nghiệm * Ôn lại các thuật toán sắp xếp về tìm kim61 |
| Tuần 10 | * Thực thi chương trình (code) * Cập nhập file báo cáo * Thiết kế test case | * Gai đoạn thực hiện và thử nghiệm |
| Tuần 11 | * Thực thi chương trình (code) * Cập nhập file báo cáo * Kiểm thử cho từng chức năng (Unit test) * Fix bug | * Gai đoạn thực hiện và thử nghiệm |
| Tuần 12 | * Cập nhập file báo cáo * Kiểm thử cho toàn bộ hệ thống (System test) * Fix bug | * Gai đoạn thực hiện và thử nghiệm |
| Tuần 13 | * Kiểm thử hệ thống và fix bug * Cập nhập file báo cáo | * Gai đoạn thực hiện và thử nghiệm |
| Tuần 14 | * Deploy hệ thống * Cập nhập file báo cáo | * Triển khai hệ thống |
| Tuần 15 | * Tổng kết và viết tài liệu báo cáo kết quả cho người hướng dẫn. * Cập nhập file báo cáo | * Triển khai hệ thống |

*Tp. Hồ Chí Minh, ngày… tháng … năm 2021*

**Giảng viên hướng dẫn**