

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

————— * —————

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI

**XÂY DỰNG ỨNG DỤNG BÁO THỨC TRÊN
HỆ ĐIỀU HÀNH ANDROID**

Sinh viên thực hiện: **Đặng Xuân Hưng**

Lớp: CN-CNTT K57

Giáo viên hướng dẫn: ThS **Hoàng Anh Việt**

Hà Nội 05-2016

Lời cảm ơn

Em xin cảm ơn thầy giáo – Thạc sỹ Hoàng Anh Việt đã chỉ bảo, giúp đỡ, tạo điều kiện cho em hoàn thành đồ án này.

Em xin cảm ơn các thầy cô trong Viện Công nghệ thông tin & Truyền thông và trong toàn trường Đại học Bách Khoa Hà Nội đã giúp đỡ em trong suốt các năm theo học tại trường, dạy bảo những kiến thức để làm nền tảng cho em thực hiện Đồ án này.

Hà Nội, ngày 17 tháng 5 năm 2016

Sinh viên thực hiện:

Đặng Xuân Hưng

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Đặng Xuân Hưng

Điện thoại liên lạc: 01632968610

Email: hungsoccerprince@gmail.com

Lớp: CN-CNTT K57

Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại: Bộ môn Công nghệ phần mềm, viện Công Nghệ Thông Tin và Truyền Thông, Đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 01/02/2016 đến 17/05/2016

2. Mục đích nội dung của ĐATN

Xây dựng ứng dụng báo thức cho di động trên hệ điều hành Android, cách tắt báo thức là nghe nhạc, xem ảnh từ thư viện của điện thoại và trả lời các câu hỏi liên quan.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu công nghệ Android
- Xây dựng ứng dụng báo thức

4. Lời cam đoan của sinh viên

Tôi - *Đặng Xuân Hưng* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS Hoàng Anh Việt*

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 17 tháng 05 năm 2016

Tác giả ĐATN

Đặng Xuân Hưng

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

.....

.....

.....

.....

.....

.....

.....

Hà Nội, Ngày Tháng Năm

Giáo viên hướng dẫn:

Mục lục

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	3
Danh mục hình vẽ	9
CHƯƠNG I: ĐẶT VẤN ĐỀ.....	11
1.1 Khảo sát thực tế.....	11
1.1.1 Khảo sát chung	11
1.1.2 Khảo sát các ứng dụng liên quan.....	11
1.1.3 Kết luận giải pháp cho ứng dụng.....	15
1.2 Nội dung đề tài	15
1.3 Bố cục đồ án.....	16
Kết chương:	16
Chương II: CƠ SỞ LÝ THUYẾT VÀ ĐỊNH HƯỚNG GIẢI PHÁP.....	17
2.1. Giới thiệu hệ điều hành Android.....	17
2.2. Kiến trúc Android.....	18
2.3. Phát triển ứng dụng trên Android.....	20
2.3.1. Ngôn ngữ lập trình.....	20
2.3.2. Vấn đề phân mảnh phiên bản Android trên thị trường:.....	21
2.3.3. Vấn đề kích thước màn hình.....	23
2.3.4. Môi trường lập trình cho Android	23
2.3.5. Android Studio	24
2.3.6. Các thành phần có bản của một project trên Android Studio.....	25
Kết chương:	26
CHƯƠNG III: XÂY DỰNG ỨNG DỤNG BÁO THỨC.....	27

3.1 Phân tích các chức năng	27
3.2. Biểu đồ Use case	32
3.2.1. Use case hiển thị danh sách báo thức	33
3.2.2. Use case thêm mới một báo thức.....	33
3.2.3. Use case chỉnh sửa báo thức	33
3.2.4. Use case xóa báo thức	34
3.2.5. Use case tắt báo thức	34
3.2.6. Use case thay đổi trạng thái on/off của báo thức.....	35
3.3. Biểu đồ tuần tự	36
3.3.1. Biểu đồ tuần tự chức năng hiển thị danh sách báo thức	36
3.3.2. Biểu đồ tuần tự chức năng thêm mới báo thức.....	37
3.3.3. Biểu đồ tuần tự chức năng chỉnh sửa báo thức.....	38
3.3.4 Biểu đồ tuần tự chức năng xóa báo thức	39
3.3.5. Biểu đồ tuần tự chức năng thay đổi trạng thái on/off báo thức	40
3.3.6. Biểu đồ tuần tự chức năng nghe nhạc tắt báo thức.....	41
3.3.7. Biểu đồ tuần tự chức năng xem ảnh tắt báo thức	42
3.3.8. Biểu đồ tuần tự chức năng tắt báo thức mặc định	43
3.4 Biểu đồ hoạt động	44
3.4.1 .Biểu đồ hoạt động cho chức năng xem danh sách báo thức.....	44
3.4.2 Biểu đồ hoạt động cho chức năng thêm mới báo thức	45
3.4.3 Biểu đồ hoạt động cho chức năng sửa báo thức	46
3.4.4 Biểu đồ hoạt động cho chức năng xóa báo thức.....	47
3.4.5. Biểu đồ hoạt động cho chức năng thay đổi trạng thái báo thức	48

3.4.6 Biểu đồ hoạt động cho chức năng tắt báo thức.....	49
3.5 Chi tiết thành phần, thuật toán sử dụng.....	50
3.5.1 PendingIntent.....	50
3.5.2 AlarmManager.....	50
3.5.3 BroadcastReceiver.....	50
3.5.4 Service.....	50
3.5.5 Chi tiết hệ thống cài đặt và phát báo thức.....	51
3.5.6 Thuật toán cài đặt báo thức gần nhất.....	53
3.6 Thiết kế CSDL.....	54
Kết chương:.....	56
CHƯƠNG IV: TRIỂN KHAI ỨNG DỤNG.....	57
4.1 Môi trường triển khai.....	57
4.2. Cách thức triển khai và sử dụng hệ thống.....	57
4.3. Kịch bản demo.....	57
4.4 Kết quả thử nghiệm.....	62
KẾT LUẬN.....	63
5.1 Các kết quả đạt được.....	63
5.1.1.Ưu điểm.....	63
5.1.2 Nhược điểm.....	63
5.2 Hướng phát triển trong tương lai.....	63
TÀI LIỆU THAM KHẢO.....	64

Danh mục các bảng

Bảng 3.1 Chi tiết chức năng xem danh sách báo thức	27
Bảng 3.2 Chi tiết chức năng thêm mới báo thức	29
Bảng 3.3 Chi tiết chức năng sửa báo thức	30
Bảng 3.4 Chi tiết chức năng xóa báo thức	30
Bảng 3.5 Chi tiết chức năng thay đổi trạng thái on/off báo thức.....	30
Bảng 3.6 Chi tiết chức năng nghe nhạc tắt báo thức	31
Bảng 3.7 Chi tiết chức năng xem ảnh tắt báo thức	32
Bảng 3.8 Use case hiển thị danh sách báo thức	33
Bảng 3.9 Use case thêm mới báo thức	33
Bảng 3.10 Use case chỉnh sửa báo thức	34
Bảng 3.11 Use case xóa báo thức	34
Bảng 3.12 Use case tắt báo thức	35
Bảng 3.13 Use case thay đổi trạng thái báo thức	36
Bảng 3.14 Bảng Alarm trong cơ sở dữ liệu	54
Bảng 3.15 Bảng Day_Alarm trong cơ sở dữ liệu.....	55
Bảng 4.1 Đánh giá kết quả thử nghiệm trên các phiên bản	62

Danh mục hình vẽ

Hình 1.1 Giao diện chính ứng dụng I Can't Wake Up! Alarm Clock	12
Hình 1.2 Giao diện cài đặt báo thức ứng dụng I Can't Wake Up! Alarm Clock.....	12
Hình 1.3 Màn hình chính ứng dụng Alarm Clock 3	13
Hình 1.4 Màn hình báo thức ứng dụng Alarm Clock 3	14
Hình 2.1 Mô hình kiến trúc nền tảng Android.....	18
Hình 2.2 Biểu đồ so sánh tốc độ của máy ảo Java giữa các phiên bản Android (Càng lớn càng tốt).....	21
Hình 2.3 So sánh giao diện nguyên bản của Android và giao diện Sense của riêng hãng HTC thiết kế.....	22
Hình 2.4 Thống kê số lượng điện thoại sử dụng các phiên bản Android	22
Hình 2.5 Cấu trúc thư mục và file của một dự án phần mềm Android trên Android Studio	25
Hình 3.1 Biểu đồ Use case của ứng dụng	32
Hình 3.2 Biểu đồ tuần tự chức năng hiển thị danh sách báo thức	36
Hình 3.3 Biểu đồ tuần tự chức năng thêm mới báo thức	37
Hình 3.4 Biểu đồ tuần tự chức năng chỉnh sửa báo thức	38
Hình 3.5 Biểu đồ tuần tự chức năng xóa báo thức.....	39
Hình 3.6 Biểu đồ tuần tự chức năng thay đổi trạng thái báo thức	40
Hình 3.7 Biểu đồ tuần tự chức năng nghe nhạc tắt báo thức	41
Hình 3.8 Biểu đồ tuần tự chức năng xem ảnh tắt báo thức.....	42
Hình 3.9 Biểu đồ tuần tự chức năng tắt báo thức mặc định.....	43
Hình 3.10 Biểu đồ hoạt động chức năng xem danh sách báo thức	44
Hình 3.11 Biểu đồ hoạt động chức năng thêm mới báo thức	45

Hình 3.12 Biểu đồ hoạt động chức năng xem sửa báo thức	46
Hình 3.13 Biểu đồ hoạt động chức năng xóa báo thức	47
Hình 3.14 Biểu đồ hoạt động chức năng thay đổi trạng thái báo thức	48
Hình 3.15 Biểu đồ hoạt động chức năng tắt báo thức.....	49
Hình 3.16 Quá trình cài đặt báo thức mới nhất.....	51
Hình 3.17 Sơ đồ thuật toán chọn báo thức gần nhất.....	53
Hình 3.18 Biểu đồ thiết kế CSDL	56
Hình 4.1 Demo chức năng hiển thị danh sách báo thức	58
Hình 4.2 Demo chức năng thêm mới hoặc chỉnh sửa báo thức	59
Hình 4.3 Demo chức năng phát âm báo thức.....	60
Hình 4.4 Demo chức năng nghe nhạc và tắt báo thức	61
Hình 4.5 Demo chức năng xem ảnh và tắt báo thức	62

CHƯƠNG I: ĐẶT VẤN ĐỀ

Nội dung chương này nhằm trình bày những vấn đề sau

- ✓ Khảo sát nhu cầu thực tế.
- ✓ Khảo sát các ứng dụng liên quan tới đề tài ứng dụng báo thức.
- ✓ Kết luận giải pháp cho ứng dụng

1.1 Khảo sát thực tế

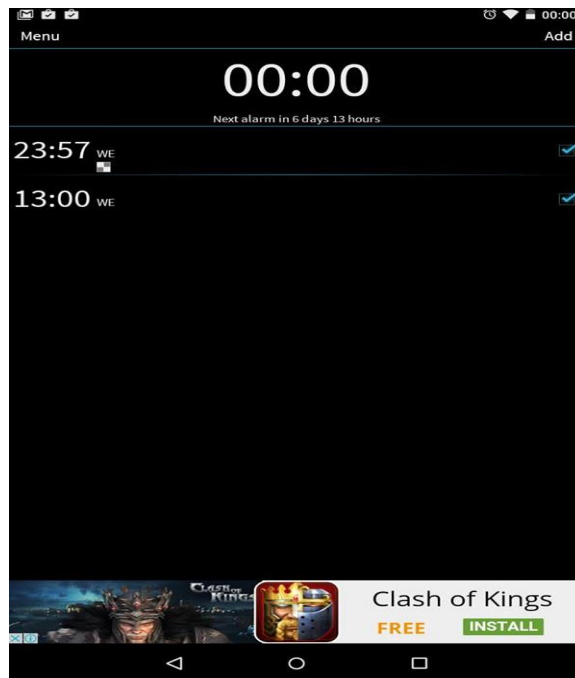
1.1.1 Khảo sát chung

Hiện nay, với sự phát triển về kinh tế, xã hội, gần như mỗi người từ học sinh, sinh viên, các bậc phụ huynh đến cả những người cao tuổi đều có thể sở hữu một chiếc Smart Phone. Vì vậy nhu cầu công việc, giải trí từ các ứng dụng trên di động là rất lớn và gần như là vô tận. Trong đó nhu cầu đặt báo thức, hẹn giờ cũng là khá phổ biến. Mỗi chiếc Smart Phone đều được trang bị sẵn một ứng dụng báo thức mặc định. Nhưng gần như nó quá nhàm chán, và đôi khi không đáp ứng đúng nhu cầu là báo thức của người dùng. Tình trạng học sinh, sinh viên, hay người đi làm thức đêm để rồi sang hôm sau “ngủ nướng” là rất phổ biến. Một báo thức thông thường chỉ khiến họ tỉnh dậy và tắt báo thức bởi một thao tác chạm đơn giản rồi tiếp tục giấc ngủ. Đây là một thói quen không hề tốt. Vấn đề đặt ra: Có thể thay đổi báo thức nhàm chán bằng một báo thức thú vị, đồng thời thay đổi được thói quen không tốt đó của người dùng?

1.1.2 Khảo sát các ứng dụng liên quan

Ứng dụng báo thức: I Can't Wake Up! Alarm Clock

Một số hình ảnh minh họa:



Hình 0.1 Giao diện chính ứng dụng I Can't Wake Up! Alarm Clock



Hình 0.2 Giao diện cài đặt báo thức ứng dụng I Can't Wake Up! Alarm Clock

Đánh giá:

❖ Ưu điểm

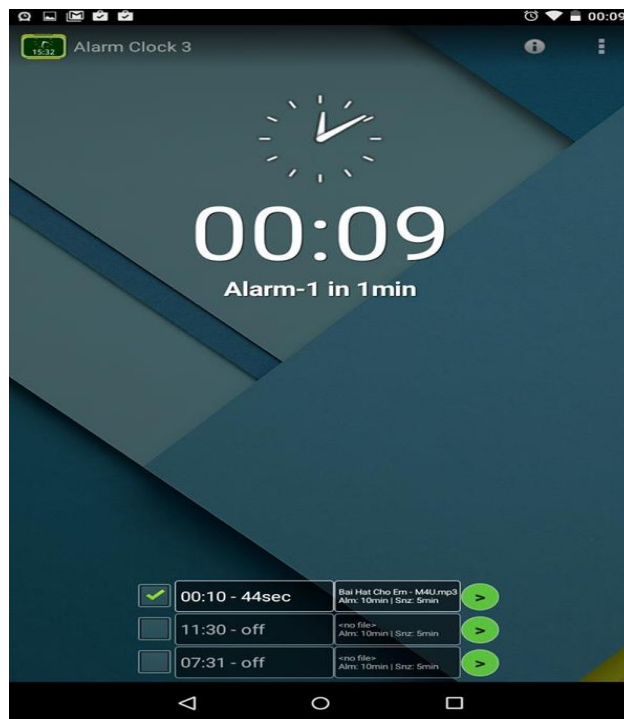
- Thiết kế :
 - ✓ Giao diện đơn giản, dễ thao tác, cài đặt báo thức
- Chức năng:
 - ✓ Có nhiều cách tắt báo thức hay: giải toán, kiểm tra trí nhớ, lắc điện thoại...

❖ Nhược điểm:

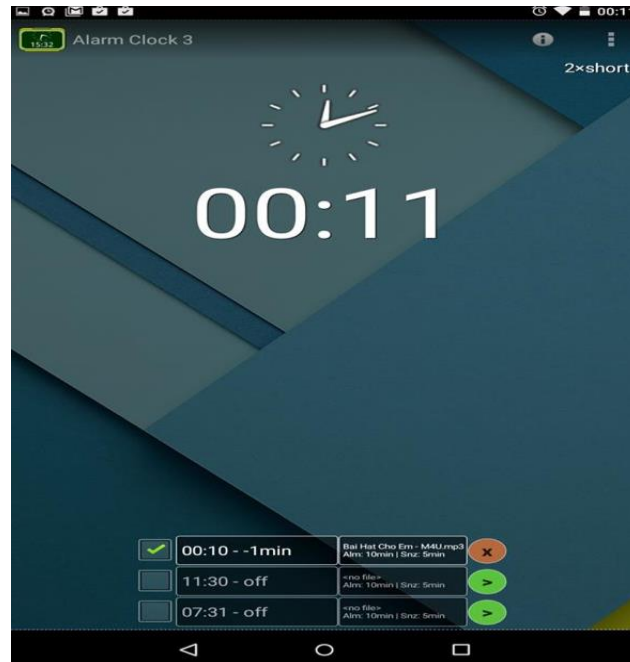
- Chức năng:
 - Báo thức không có title, đôi khi người dùng muốn cài đặt báo thức và nhắc nhở làm một việc gì đó thì title là một gợi ý rất hay.
- Thiết kế:
 - Button để thêm báo thức quá nhỏ, phía trên góc màn hình, khó phát hiện và tương tác.

Ứng dụng báo thức: Alarm Clock 3

Một số hình ảnh minh họa:



Hình 0.3 Màn hình chính ứng dụng Alarm Clock 3



Hình 0.4 Màn hình báo thức ứng dụng Alarm Clock 3

Đánh giá:

❖ Ưu điểm:

- Chức năng:

- ✓ Có thể phát âm báo thức bình thường ngay cả khi âm lượng điện thoại đang ở mức thấp nhất.
- ✓ Có thể cập nhật khoảng thời gian tới báo thức gần nhất liên tục.

❖ Nhược điểm:

- Chức năng:

- Chỉ cài đặt được tối đa 3 báo thức, điều này đôi khi là gây khó chịu với người dùng có nhu cầu cài đặt nhiều báo thức.
- Chức năng tắt báo thức đơn giản, như ứng dụng báo thức mặc định của điện thoại.

- Thiết kế:

- Button để click tắt báo thức quá nhỏ, khó phát hiện.

1.1.3 Kết luận giải pháp cho ứng dụng

❖ Chức năng:

- Để cài đặt báo thức, chỉ cần các thông số: thời gian, tiêu đề, âm báo, cách báo thức. Giảm thiểu tối đa các thông số không thực sự cần thiết đối với nhu cầu báo thức của người dùng.
- Cách tắt báo thức: ngoài cách tắt mặc định thì có 2 cách khác:
Cách 1: nghe nhạc (ngẫu nhiên) lấy từ thư viện của điện thoại và đoán tên bài hát để tắt báo thức.
Cách 2: Xem ảnh (ngẫu nhiên) lấy từ thư viện ảnh của điện thoại và đoán tên ảnh, thời gian chụp để tắt báo thức.
Yêu cầu phải trả lời đúng 3 bản nhạc hoặc 3 hình ảnh thì báo thức mới tắt. nếu không sẽ phát mãi mãi.
- Báo thức có thể được phát lại theo ngày trong mỗi tuần cho đến khi người dùng xóa hoặc đặt trạng thái off.

❖ Thiết kế:

- Giao diện cần hiển thị rõ ràng, ưu tiên các thành phần quan trọng sẽ chiếm vị trí dễ phát hiện và thao tác cùng diện tích lớn.
- Bố trí các thành phần đơn giản, dễ thao tác nhất cho người dùng.

1.2 Nội dung đề tài

Tên đề tài

Xây dựng ứng dụng báo thức cho di động trên hệ điều hành Android

Nhiệm vụ của đề tài

Đề tài nhằm xây dựng một ứng dụng báo thức có tính giải trí, và mang lại hiệu quả báo thức cho người dùng. Người dùng có thể thêm, sửa, xóa một báo thức như bình thường. Và khi chuông báo thức rung lên, người dùng sẽ được nghe những bản nhạc từ chính list nhạc trong điện thoại của mình, hoặc xem những bức ảnh từ thư viện ảnh của điện thoại. Nhiệm vụ của người dùng là phải trả lời đúng tên của 3 bản nhạc, hoặc 3 câu hỏi về thời gian chụp, tên của những bức ảnh. Nếu không những bản nhạc hoặc âm báo thức này sẽ phát mãi mãi. Tất nhiên giấc ngủ sẽ rất khó được tiếp tục.

Đây là ứng dụng trải nghiệm đầu tiên của em về hệ điều hành Android, do còn hạn chế về kiến thức nên ứng dụng sẽ không tránh khỏi những vấn đề về thiết kế, chức năng... Em mong

được sự góp ý của thầy cô để hoàn thiện ứng dụng cũng như có kinh nghiệm cho những ứng dụng tiếp theo.

Em xin chân thành cảm ơn thầy Hoàng Anh Việt đã hướng dẫn, giúp đỡ em trong quá trình thực hiện đề tài này.

1.3 Bố cục đồ án

Đồ án gồm các phần chính sau:

Chương 1: Đặt vấn đề

Trình bày tổng quát về đề tài và bố cục của đề tài

Chương 2: Cơ sở lý thuyết và định hướng giải pháp

Chương này trình bày vấn đề về nền tảng cơ sở lý thuyết khi thực hiện đề tài, cũng như định hướng giải pháp chung cho đề tài và môi trường, công cụ, cơ sở lý thuyết và các công nghệ sử dụng trong đề tài.

Chương 3: Xây dựng ứng dụng báo thức

Chương này mô tả chi tiết phân tích thiết kế, mô hình hoạt động, các công nghệ, thành phần và thuật toán sử dụng

Chương 4: Triển khai ứng dụng

Chương này trình bày chi tiết môi trường triển khai, cách thức triển khai và kịch bản demo hệ thống.

Chương 5: Kết luận

Rút ra kết luận chung cho đề tài, ưu điểm nhược điểm của chương trình và các vấn đề làm được và chưa làm được từ đó đưa ra định hướng phát triển đề tài.

Kết chương:

Chương này đã đưa ra những đánh giá về nhu cầu người dùng qua các ứng dụng liên quan, định hướng đề tài và tóm tắt bố cục đồ án. Chương tiếp theo sẽ trình bày chi tiết cơ sở lý thuyết và định hướng giải pháp cho đề tài.

Chương II: CƠ SỞ LÝ THUYẾT VÀ ĐỊNH HƯỚNG GIẢI PHÁP

Nội dung chương này nhằm trình bày những vấn đề sau

- ✓ Giới thiệu lịch sử phát triển hệ điều hành Android, kiến trúc Android
- ✓ Môi trường lập trình Android
- ✓ Giới thiệu công cụ phát triển ứng dụng

2.1. Giới thiệu hệ điều hành Android

Tháng 7/2005, Google mua lại công ty phát triển phần mềm điện thoại Android, nhưng chưa công bố sẽ sử dụng cho mục đích gì. Để trả lời cho câu hỏi đó, tháng 11/2007, sau 2 năm phát triển, Google công bố hệ điều hành điện thoại di động mã nguồn mở Android, cùng với sự thành lập “Liên minh di động mở” (Open Handset Alliance) bao gồm hơn 65 nhà sản xuất phần cứng điện thoại lớn trên thế giới như Intel, HTC, China Mobile, T-Mobile,....

Android là hệ điều hành điện thoại di động mã nguồn mở miễn phí do Google phát triển dựa trên nền tảng của Linux. Bất kỳ một hãng sản xuất phần cứng nào cũng đều có thể tự do sử dụng hệ điều hành Android cho thiết bị của mình, miễn là các thiết bị ấy đáp ứng được các tiêu chuẩn cơ bản do Google đặt ra (có cảm ứng chạm, GPS, 3G,...)(Xem thêm: **Android Compatibility Definition Document**)

Các nhà sản xuất có thể thay đổi phiên bản Android trên máy của mình một cách tự do mà không cần phải xin phép hay trả bất kỳ khoản phí nào nhưng phải đảm bảo tính tương thích ngược (backward compatibility) của phiên bản chế riêng đó.

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

Những tính năng mà nền tảng Android hỗ trợ:

Application framework: Cho phép tái sử dụng và thay thế các thành phần sẵn có của Android.

Dalvik virtual machine: Máy ảo java được tối ưu hóa cho thiết bị di động.

Integrated browser: Trình duyệt web tích hợp được xây dựng dựa trên WebKit engine.

Optimized graphics: Hỗ trợ bộ thư viện 2D và 3D dựa vào đặc tả OpenGL ES 1.0.

SQLite: DBMS dùng để lưu trữ dữ liệu có cấu trúc.

Hỗ trợ các định dạng media phổ biến như: MPEG4, H.264, MP3, AAC, ARM, JPG, PNG, GIF.

Hỗ trợ thoại trên nền tảng GSM (Phụ thuộc vài phần cứng thiết bị).

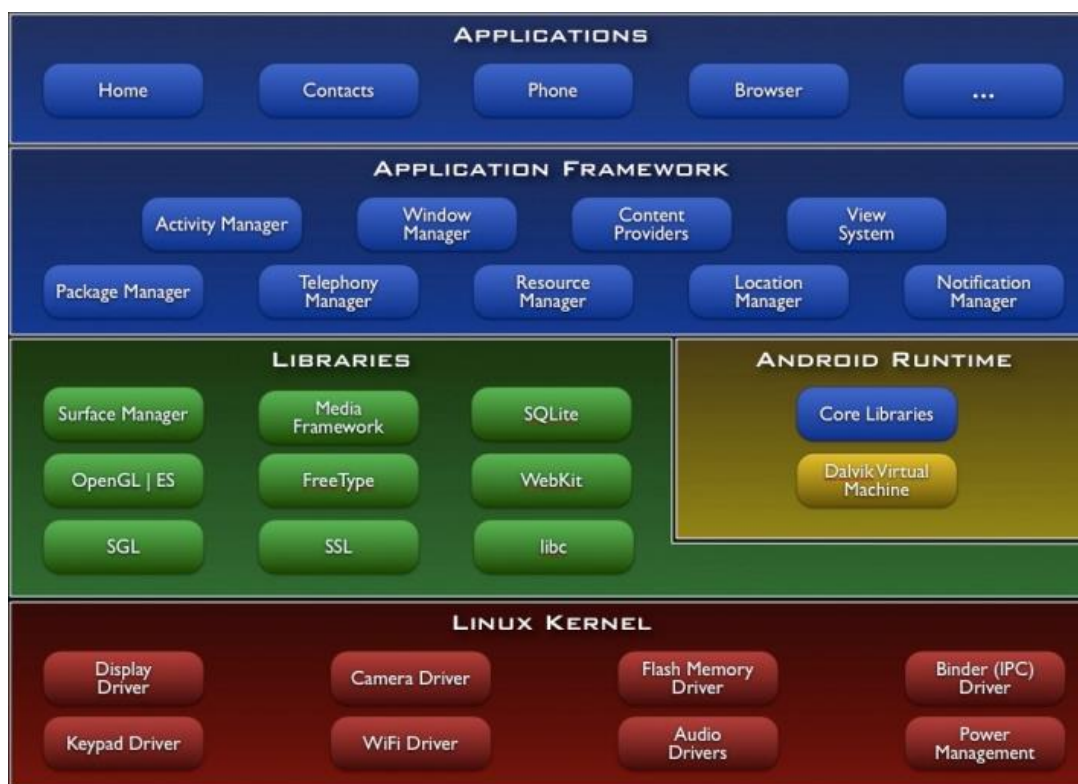
Bluetooth, EDGE, 3G và WiFi (Phụ thuộc vài phần cứng thiết bị).

Camera, GPS, la bàn và cảm biến (Phụ thuộc vài phần cứng thiết bị).

Bộ công cụ phát triển ứng dụng mạnh mẽ.

2.2. Kiến trúc Android

Lược đồ sau thể hiện các thành phần của hệ điều hành Android:



Hình 0.1 Mô hình kiến trúc nền tảng Android

Applications

Hệ điều hành Android tích hợp sẵn một số ứng dụng cơ bản như email client, SMS, lịch điện tử, bản đồ, trình duyệt web, sổ liên lạc và một số ứng dụng khác. Ngoài ra tầng này cũng chính là tầng chứa các ứng dụng được phát triển bằng ngôn ngữ Java.

Application Framework

Tầng này của hệ điều hành Android cung cấp một nền tảng phát triển ứng dụng mở qua đó cho phép nhà phát triển ứng dụng có khả năng tạo ra các ứng dụng vô cùng sáng tạo và phong phú. Các nhà phát triển ứng dụng được tự do sử dụng các tính năng cao cấp của thiết bị phần cứng như: thông tin định vị địa lý, khả năng chạy dịch vụ dưới nền, thiết lập đồng hồ báo thức, thêm notification vào status bar của màn hình thiết bị...

Người phát triển ứng dụng được phép sử dụng đầy đủ bộ API được dùng trong các ứng dụng tích hợp sẵn của Android. Kiến trúc ứng dụng của Android được thiết kế nhằm mục đích đơn giản hóa việc tái sử dụng các component. Qua đó bất kì ứng dụng nào cũng có thể công bố các tính năng mà nó muốn chia sẻ cho các ứng dụng khác (VD: Ứng dụng email có muốn các ứng dụng khác có thể sử dụng tính năng gửi mail của nó). Phương pháp tương tự cho phép các thành phần có thể được thay thế bởi người sử dụng.

Tầng này bao gồm một tập các services và thành phần sau:

Một tập phong phú và có thể mở rộng bao gồm các đối tượng View được dùng để xây dựng ứng dụng như: list, grid, text box, button và thậm chí là một trình duyệt web có thể nhúng vào ứng dụng.

Content Provider: Cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác hoặc chia sẻ dữ liệu của chúng.

Resource Manager: Cung cấp khả năng truy xuất các tài nguyên non-code như hình ảnh hoặc file layout.

Notification Manager: Cung cấp khả năng hiển thị custom alert trên thanh status bar.

Activity Manager: Giúp quản lý vòng đời của một ứng dụng.

Libraries

Hệ điều hành Android bao gồm một tập các bộ thư viện C/C++ được sử dụng bởi nhiều thành phần của Android system. Những tính năng này được cung cấp cho các lập trình viên thông qua bộ framework của Android. Dưới đây là một số thư viện cốt lõi:

System C library: một thể hiện được xây dựng từ BSD của bộ thư viện hệ thống C chuẩn (libc), được điều chỉnh để tối ưu hóa cho các thiết bị chạy trên nền Linux.

Media libraries: Bộ thư viện hỗ trợ trình diễn và ghi các định dạng âm thanh và hình ảnh phổ biến.

Surface manager: Quản lý hiển thị nội dung 2D và 3D.

LibWebCore: Một web browser engine hiện đại được sử dụng trong trình duyệt của Android lẫn trong trình duyệt nhúng web view được sử dụng trong ứng dụng.

SGL: Engine hỗ trợ đồ họa 2D.

3D libraries: Một thể hiện được xây dựng dựa trên các APIs của OpenGL ES 1.0. Những thư viện này sử dụng các tăng tốc 3D bằng phần cứng lẫn phần mềm để tối ưu hóa hiển thị 3D.

FreeType: Bitmap and vector font rendering.

SQLite: Một DataBaseManagement System nhỏ gọn và mạnh mẽ.

Android Runtime

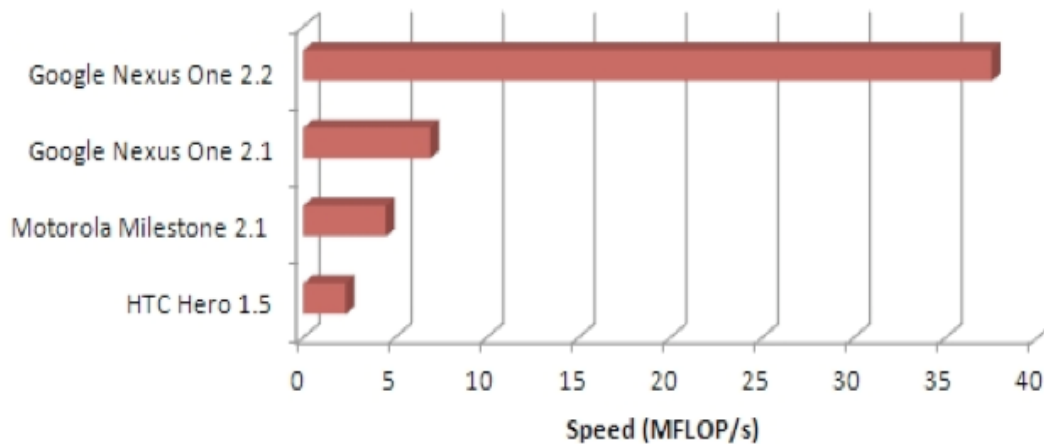
Hệ điều hành Android tích hợp sẵn một tập hợp các thư viện cốt lõi cung cấp hầu hết các chức năng có sẵn trong các thư viện lõi của ngôn ngữ lập trình Java. Mọi ứng dụng của Android chạy trên một tiến trình của riêng nó cùng với một thể hiện của máy ảo Dalvik. Máy ảo Dalvik thực tế là một biến thể của máy ảo Java được sửa đổi, bổ sung các công nghệ đặc trưng của thiết bị di động. Nó được xây dựng với mục đích làm cho các thiết bị di động có thể chạy nhiều máy ảo một cách hiệu quả. Trước khi thực thi, bất kỳ ứng dụng Android nào cũng được convert thành file thực thi với định dạng nén Dalvik Executable (.dex). Định dạng này được thiết kế để phù hợp với các thiết bị hạn chế về bộ nhớ cũng như tốc độ xử lý. Ngoài ra máy ảo Dalvik sử dụng bộ nhân Linux để cung cấp các tính năng như thread, low-level memory management.

2.3. Phát triển ứng dụng trên Android

2.3.1. Ngôn ngữ lập trình

Ngôn ngữ lập trình chính thức của Android là Java. Mặc dù các ứng dụng trên Android được phát triển dựa trên nền tảng Java, nhưng Android không hỗ trợ J2ME và J2SE, là hai ngôn ngữ lập trình phổ dụng cho các thiết bị di động.

Dựa trên máy ảo Java của Sun, Google đã tinh chỉnh và phát triển nên máy ảo Dalvik để biên dịch mã Java với tốc độ biên dịch nhanh hơn và nhẹ hơn. Đến phiên bản Froyo 2.2, Android đã hỗ trợ Just-in-time Compiler (JIT) làm tăng tốc độ biên dịch Java lên gấp 2-5 lần so với các phiên bản trước.



Hình 0.2 Biểu đồ so sánh tốc độ của máy ảo Java giữa các phiên bản Android (Càng lớn càng tốt)

Bằng cách sử dụng các frameworks của hãng thứ ba (VD: PhoneGap, Titanium,...), các nhà lập trình web cũng có thể phát triển ứng dụng Android một cách dễ dàng bằng các ngôn ngữ web phổ dụng như HTML, CSS, Javascript. Tuy nhiên số lượng các ứng dụng đi theo chiều hướng này chưa nhiều.

Ngoài ra các ứng dụng của Android sử dụng SQLite để quản lý cơ sở dữ liệu.

2.3.2. Vấn đề phân mảnh phiên bản Android trên thị trường:

Do Android là hệ điều hành mã nguồn mở và miễn phí, bất kỳ một nhà sản xuất phần cứng nào cũng có thể sử dụng và tùy biến Android để cài đặt trên các thiết bị của mình. Điều này dẫn đến vấn đề phân mảnh (fragmentation) khi trên thị trường tồn tại nhiều mẫu điện thoại Android, mỗi mẫu chạy một phiên bản Android khác nhau. Đôi khi các phiên bản Android này còn khác nhau về giao diện hay chức năng, do các nhà sản xuất tự chế thêm vào mã nguồn của Android. Một ví dụ tiêu biểu chính là giao diện Sense UI của HTC.

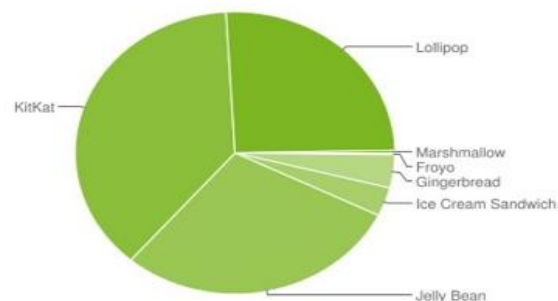


Hình 0.3 So sánh giao diện nguyên bản của Android và giao diện Sense của riêng hãng HTC thiết kế

Mặt khác, do việc cập nhật hệ điều hành phụ thuộc hoàn toàn vào các nhà sản xuất, thế nên sau khi Google công bố phiên bản Android mới, người dùng sẽ buộc phải chờ khá lâu mới được cập nhật.

Do vấn đề phân mảnh này, trên thị trường một lúc tồn tại nhiều phiên bản Android khác nhau, và người phát triển ứng dụng Android buộc phải xem xét đến khả năng ứng dụng do mình viết ra có thể sẽ không chạy được hay thiếu đi một số chức năng quan trọng khi chạy trên các máy chạy phiên bản Android cũ.

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.3%
4.1.x	Jelly Bean	16	11.0%
4.2.x		17	13.9%
4.3		18	4.1%
4.4	KitKat	19	37.8%
5.0	Lollipop	21	15.5%
5.1		22	10.1%
6.0	Marshmallow	23	0.3%



Hình 0.4 Thống kê số lượng điện thoại sử dụng các phiên bản Android

2.3.3. Vấn đề kích thước màn hình

Một lần nữa, do trên thị trường có rất nhiều nhà sản xuất điện thoại Android, và mặc dù Google yêu cầu tất cả mọi điện thoại Android phải đáp ứng 1 số tiêu chí chung về phần cứng (màn hình cảm ứng, GPS, 3G,...), các nhà sản xuất hoàn toàn tự do quyết định kích cỡ và độ phân giải điện thoại.

Như vậy, các nhà phát triển ứng dụng Android phải chuẩn bị trước khả năng ứng dụng của mình sẽ chạy trên một loạt các kích cỡ màn hình lớn nhỏ khác nhau.

Tuy nhiên, có một điều chắc chắn: 100% các điện thoại Android đều sử dụng màn hình cảm ứng đa điểm, còn bàn phím vật lý hay cảm ứng quang thì tùy máy.

2.3.4. Môi trường lập trình cho Android

Android SDK bao gồm các công cụ riêng lẻ như: debugger, các thư viện, trình giả lập điện thoại Android, các tài liệu hỗ trợ và code mẫu. Hiện Android cung cấp bộ công cụ này trên nhiều nền tảng hệ điều hành khác nhau (Windows, Linux, Mac,...), miễn là có sẵn Java Development Kit, Apache Ant và Python 2.2 trở lên.

Môi trường lập trình (IDE) chính thức của Android là Eclipse (từ phiên bản 3.2) với sự hỗ trợ của plugin Android Development Tools (ADT). Tuy nhiên, người lập trình có thể sử dụng bất kỳ IDE hay trình soạn thảo văn bản nào để viết code Java và XML rồi biên dịch nên ứng dụng hoàn chỉnh bằng cách sử dụng dòng lệnh (command lines). Và gần đây đây, Google đã nói gót ông lớn Microsoft để tự tạo ra bộ công cụ lập trình chuẩn riêng, đó là Android Studio.

Ứng dụng Android được đóng gói thành các file .apk và được lưu trong thư mục /data/app của hệ điều hành Android. Java Development Kit (JDK) 5.0.

Một số công cụ hỗ trợ lập trình Android tiêu biểu:

SQLite Manager: Là một addon của Firefox giúp quản lý cơ sở dữ liệu SQLite của Android.

DroidDraw: Giúp thiết kế file XML giao diện ứng dụng.

Balsamiq Mockups và Adobe Fireworks: Giúp nhanh chóng phác thảo ý tưởng và giao diện sơ bộ của ứng dụng.

StarUML: Vẽ các lược đồ UML hỗ trợ phân tích thiết kế.

2.3.5. Android Studio

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Bộ công cụ Android Studio cung cấp những trình soạn thảo riêng biệt tương ứng với hầu hết các file cấu hình và Layout của ứng dụng Android với định dạng XML. Với những file Layout giao diện, Android Studio cho phép các lập trình viên dễ dàng chuyển đổi chế độ chỉnh sửa giữa trình biên soạn nội dung XML hoặc trình biên soạn dưới dạng giao diện (GUI). Ngoài ra, bộ phát triển Android Studio còn tích hợp bên trong những tiện ích hỗ trợ phát triển ứng dụng như sau:

Bộ xây dựng ứng dụng Gradle với nhiều cấu hình linh động.

Cho phép xây dựng ứng dụng tùy biến và tự động tạo ra file .apk tương thích với thông tin tùy biến.

Bộ code mẫu giúp chúng ta xây dựng các chức năng phổ biến của các ứng dụng.

Trình biên soạn Layout GUI cho ứng dụng Android phong phú và tiện lợi, cho phép người dùng dễ dàng tạo ra giao diện màn hình bằng cách kéo thả các Component mẫu có sẵn và chỉnh sửa giao diện themes (kích thước, màu sắc...) tùy ý.

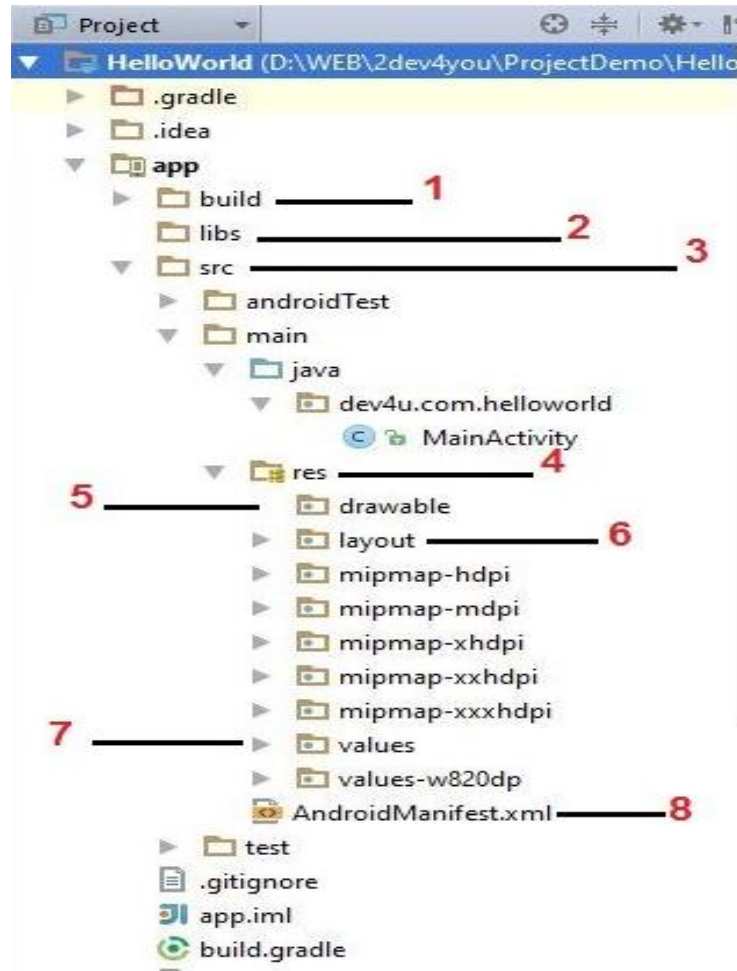
Tích hợp **lint** – Một ứng dụng tích hợp giúp các developer kiểm soát về hiệu suất (performance), tính khả dụng (usability), khả năng tương thích của các phiên bản API sử dụng, và những vấn đề tiềm ẩn bên trong có thể xảy ra lúc Runtime.

ProGuard (tiện ích tối ưu và mã hóa code khi build ứng dụng) và Android app-signing.

Bộ tích hợp hỗ trợ phát triển ứng dụng Android dễ dàng với các dịch vụ của nền tảng đám mây của Google

Mới đây, Google đã cho ra mắt phiên bản 2.0, và ngay sau đó là 2.1 với các tính năng mới nổi bật là **Instant Run**, trình giả lập Android nhanh hơn, Cloud Test Lab và GPU Debugger.

2.3.6. Các thành phần cơ bản của một project trên Android Studio



Hình 0.5 Cấu trúc thư mục và file của một dự án phần mềm Android trên Android Studio

build: Thư mục này chứa các tập tin được tạo tự động như Aidl, Build configuration, R(R.JAVA) và thư mục outputs chứa file apk.

libs: Đây là một thư mục để thêm các thư viện để phát triển các ứng dụng Android.

src: Thư mục này chứa các file nguồn .java cho cho project của bạn. Mặc định, nó bao gồm file nguồn MainActivity.java.

res: Thư mục này chứa các thư mục con khác như drawable , layout, values....

res/drawable: Thư mục này chứa các hình ảnh hoặc các file xml được sử dụng trong ứng dụng, ví dụ 1 hình ảnh để làm background cho app...

res/layout: Đây là một thư mục chứa các tập tin định nghĩa giao diện người dùng trong ứng dụng của bạn.

res/values: Đây là một thư mục chứa các tập tin XML khác nhau chứa một tập hợp các nguồn lực, chẳng hạn như các chuỗi string trong có trong ứng dụng và màu sắc .

AndroidManifest.xml: Đây là file trong đó mô tả các đặc điểm cơ bản của ứng dụng và xác định từng thành phần của nó.

Kết chương:

Chương này đã trình bày về hệ điều hành Android, môi trường và công cụ phát triển ứng dụng. Chương tiếp theo sẽ mô tả chi tiết phân tích thiết kế, công nghệ, thuật toán sử dụng thực hiện đề tài.

CHƯƠNG III: XÂY DỰNG ỨNG DỤNG BÁO THỨC

Nội dung chương này nhằm trình bày những vấn đề sau

- ✓ Phân tích các chức năng hoạt động của ứng dụng.
- ✓ Biểu đồ trình tự mô tả trình tự hoạt động của từng chức năng.
- ✓ Biểu đồ hoạt động mô tả cách thức hoạt động, các tác nhân trong từng chức năng
- ✓ Phân tích các thành phần, thuật toán sử dụng
- ✓ Thiết kế cơ sở dữ liệu cho ứng dụng

3.1 Phân tích các chức năng

Ứng dụng thiết kế cho người dùng có thể cài đặt báo thức như những ứng dụng báo thức mặc định. Điểm khác biệt là ở cách tắt báo thức.

Cụ thể người dùng có thể:

- Xem danh sách các báo thức đã cài đặt
- Thêm báo thức mới.
- Sửa báo thức đã cài đặt
- Xóa báo thức
- Thay đổi trạng thái cho báo thức đã cài đặt : on/off
- Nghe nhạc và đoán tên khi muốn tắt báo thức.
- Xem ảnh, trả lời câu hỏi để tắt báo thức.

Chi tiết chức năng xem danh sách báo thức:

Thứ tự	Thao tác người dùng	Màn hình	Mô tả hệ thống
1	Mở ứng dụng		- Ứng dụng mở màn hình Main Activity. - Hiện thị đồng hồ. - Lấy danh sách báo thức từ cơ sở dữ liệu và hiển thị ra màn hình - Với mỗi item hiển thị thời gian, title và một Switch Button thể hiện trạng thái on/off của báo thức.

Bảng 0.1 Chi tiết chức năng xem danh sách báo thức

Chi tiết chức năng thêm mới báo thức

Thứ tự	Thao tác người dùng	Màn hình	Mô tả hệ thống, Control
1	Click Add Button	MainActivity	- Ứng dụng mở màn hình AddAlarmActivity. - Hiển thị thông tin cần cài đặt cho báo thức: + TextView: hiển thị thời gian. + EditText :nhập Title + TextView: hiển thị loại báo thức (mặc định hoặc chơi game) + CheckBox: lựa chọn báo rung hoặc không. + TextView: hiển thị âm báo thức lựa chọn.
2	Click TextView hiển thị thời gian	AddAlarmActivity	- Ứng dụng hiển thị TimeDialog cho người dùng cài đặt thời gian báo thức
3	Nhập Title cho báo thức vào EditText	AddAlarmActivity	- Hiển thị bàn phím cho người dùng nhập Title báo thức
4	Click TextView hiển thị loại báo thức	AddAlarmActivity	- Hiển thị AlertDialog cho người dùng lựa chọn loại báo thức: mặc định hoặc chơi game.
5	Check vào CheckBox chọn/bỏ chọn chế độ rung	AddAlarmActivity	- Xác nhận chế độ rung tương ứng trạng thái của checkbox.
6	Click TextView hiển thị âm báo	AddAlarmActivity	- Mở màn hình SelectRingActivity để người dùng chọn âm báo. - Tại màn hình này, người dùng có thể lựa chọn âm báo có sẵn của ứng dụng, hoặc lấy từ danh sách bài hát trong điện thoại.
7	Chọn một âm báo thức	SelectRingActivity	- Trở về màn hình AddAlarmActivity và hiển thị tên âm báo đã chọn lên TextView âm báo.
8	Click Save Button	AddAlarmActivity	- Lưu thông tin báo thức vào cơ sở dữ liệu.

			- Start Service SetAlarmService để cài đặt báo thức gần nhất cho điện thoại. - Trở về màn hình MainActivity
9	Click Back Button	AddAlarmActivity	- Trở về màn hình MainActivity

Bảng 0.2 Chi tiết chức năng thêm mới báo thức**Chi tiết chức năng chỉnh sửa báo thức**

Thứ tự	Thao tác người dùng	Màn hình	Mô tả hệ thống, Control
1	Click chọn một item trên danh sách báo thức.	MainActivity	- Ứng dụng mở màn hình EddAlarmActivity. - Hiện thị thông tin đã cài đặt cho báo thức: + TextView: hiển thị thời gian. + EditText :nhập Title + TextView: hiển thị loại báo thức (mặc định hoặc chơi game) + CheckBox: báo rung hoặc không. + TextView: hiển thị âm báo thức lựa chọn.
2	Click TextView hiển thị thời gian	EditAlarmActivity	- Ứng dụng hiển thị TimeDialog cho người dùng chỉnh sửa thời gian báo thức
3	Nhập Title cho báo thức vào EditText	EditAlarmActivity	- Hiện thị bàn phím cho người dùng chỉnh sửa Title báo thức
4	Click TextView hiển thị loại báo thức	EditAlarmActivity	- Hiện thị AlertDialog cho người dùng lựa chọn loại báo thức: mặc định, music, photo
5	Check vào CheckBox chọn/bỏ chọn chế độ rung	EditAlarmActivity	- Xác nhận chế độ rung tương ứng trạng thái của checkbox.
6	Click TextView hiển thị âm báo	EditAlarmActivity	- Mở màn hình SelectRingActivity để người dùng chọn âm báo. - Tại màn hình này, người dùng có thể lựa chọn âm báo có sẵn của ứng dụng, hoặc lấy từ danh sách bài hát trong điện thoại.

7	Chọn một âm báo thức	SelectRingActivity	- Trở về màn hình EditAlarmActivity và hiển thị tên âm báo đã chọn lên TextView âm báo.
8	Click Save Button	AddAlarmActivity	- Update báo thức vào cơ sở dữ liệu. - Start Service SetAlarmService để cài đặt báo thức gần nhất cho điện thoại. - Trở về màn hình MainActivity
9	Click Back Button	AddAlarmActivity	- Trở về màn hình MainActivity

Bảng 0.3 Chi tiết chức năng sửa báo thức**Chi tiết chức năng xóa báo thức:**

Thứ tự	Thao tác người dùng	Màn hình	Mô tả hệ thống
1	Click chọn một item trên danh sách báo thức.	MainActivity	- Ứng dụng mở màn hình EditAlarmActivity - Hiển thị đồng hồ và thông tin báo thức đã cài đặt.
2	Click delete button	EditAlarmActivity	- Xóa báo thức trong cơ sở dữ liệu. - Start Service SetAlarmService để cài đặt báo thức gần nhất cho điện thoại. - Trở về màn hình MainActivity

Bảng 0.4 Chi tiết chức năng xóa báo thức**Chi tiết chức năng thay đổi trạng thái cho báo thức đã cài đặt: on/off**

Thứ tự	Thao tác người dùng	Màn hình	Mô tả hệ thống
1	Vuốt Switch button tương ứng với mỗi báo thức trong danh sách báo thức để thay đổi trạng thái on/off.	MainActivity	- Update báo thức trong cơ sở dữ liệu. - Start Service SetAlarmService để cài đặt báo thức gần nhất cho điện thoại.

Bảng 0.5 Chi tiết chức năng thay đổi trạng thái on/off báo thức

Chi tiết chức năng nghe nhạc và đoán tên khi muốn tắt báo thức

Thứ tự	Thao tác người dùng	Màn hình	Mô tả hệ thống
1		StopAlarmActivity	- Khi đến thời gian đã cài đặt của báo thức, hiển thị màn hình StopAlarmActivity. - Đồng thời phát âm báo thức để báo hiệu cho người dùng.
2	Click Stop button	StopAlarmActivity	- Nếu báo thức là loại “Music”, ngừng báo thức, mở màn hình AudioActiviy và phát một bản nhạc được lấy từ danh sách trong điện thoại, kèm theo hiển thị 4 phương án tên bản nhạc cho người dùng lựa chọn.
3	Click 1 button chọn đáp án	AudioActivity	- Nếu đáp án chọn là đúng, sẽ được cộng 1 điểm, sai thì giữ nguyên. - Cứ lần lượt phát từng bản nhạc và trả lời đến khi đạt đủ 3 điểm, thông báo hoàn thành, ngừng phát bản nhạc và sau 5 giây sẽ thoát màn hình - Start Service SetAlarmService để cài đặt báo thức mới gần nhất cho điện thoại.

Bảng 0.6 Chi tiết chức năng nghe nhạc tắt báo thức**Chi tiết chức năng xem ảnh, trả lời câu hỏi tắt báo thức**

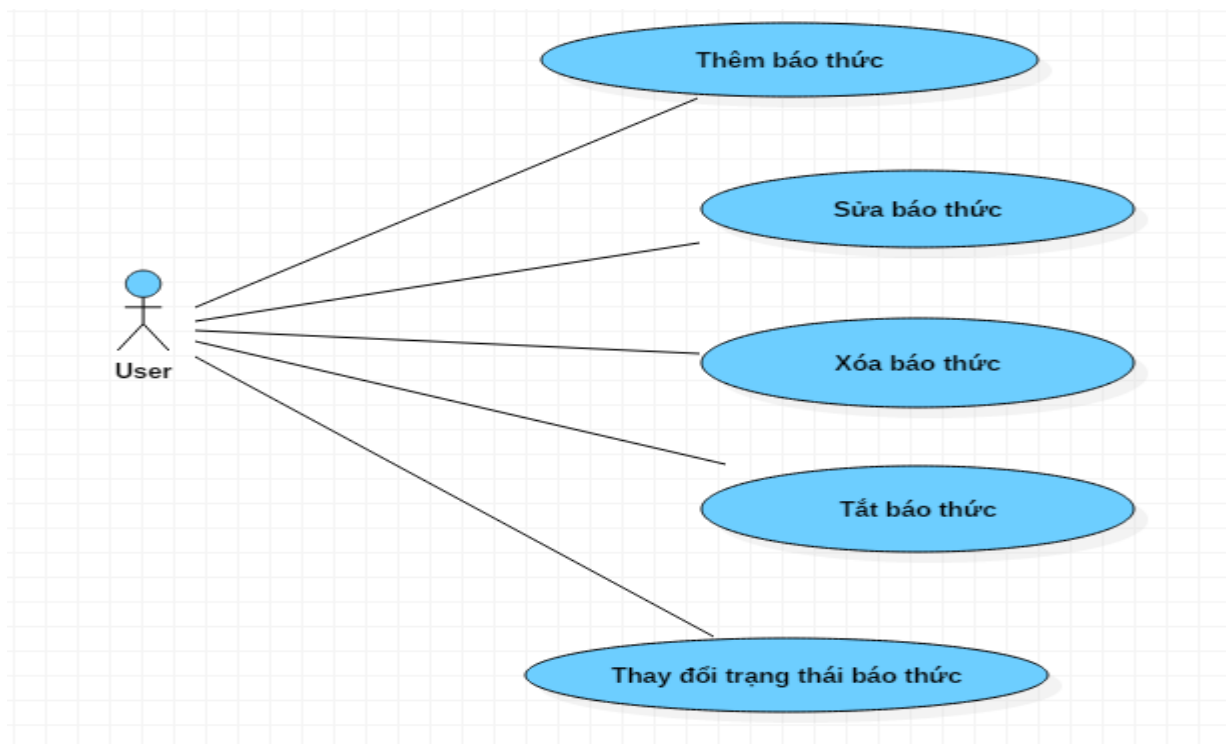
Thứ tự	Thao tác người dùng	Màn hình	Mô tả hệ thống
1		StopAlarmActivity	- Khi đến thời gian đã cài đặt của báo thức, hiển thị màn hình StopAlarmActivity. - Đồng thời phát âm báo thức để báo hiệu cho người dùng.
2	Click Stop button	StopAlarmActivity	- Nếu báo thức là loại “Photo”, ngừng báo thức, mở màn hình PhotoActiviy hiển thị một bức ảnh được lấy từ danh sách trong điện thoại, kèm theo câu hỏi và 4

			phương án trả lời liên quan đến bức ảnh cho người dùng lựa chọn.
3	Click 1 button chọn đáp án	AudioActivity	<ul style="list-style-type: none">- Nếu đáp án chọn là đúng, sẽ được cộng 1 điểm, sai thì giữ nguyên.- Cứ lần lượt hiển thị ngẫu nhiên các bức ảnh cho người dùng trả lời đến khi đạt đủ 3 điểm, thông báo hoàn thành, tắt âm báo thức và sau 5 giây sẽ thoát màn hình- Start Service SetAlarmService để cài đặt báo thức mới gần nhất cho điện thoại.

Bảng 0.7 Chi tiết chức năng xem ảnh tắt báo thức

3.2. Biểu đồ Use case

Qua biểu đồ này, ta sẽ thấy được quá trình thực hiện các chức năng của chương trình cần thông qua những thao tác nào và quan hệ giữa những thao tác đó ra sao.



Hình 0.1 Biểu đồ Use case của ứng dụng

3.2.1. Use case hiển thị danh sách báo thức

Use Case	List Alarm
Actor	User
Brief Description	Hiển thị danh sách báo thức
Main Flow	<ol style="list-style-type: none">1. Mở ứng dụng2. Hệ thống truy vấn CSDL và trả về kết quả danh sách báo thức3. Nhận và hiển thị kết quả cho người dùng lên màn hình MainActivity

Bảng 0.8 Use case hiển thị danh sách báo thức

3.2.2. Use case thêm mới một báo thức

Use Case	Add Alarm
Actor	User
Brief Description	Thêm mới một báo thức
Main Flow	<ol style="list-style-type: none">1. Click add button, mở màn hình AddAlarmActivity2. Cài đặt các thuộc tính cho báo thức3. Click save button4. Hệ thống lưu báo thức vào cơ sở dữ liệu, start Service cài đặt báo thức gần nhất. Trở về màn hình MainActivity
Alternative Flows	4.1. Nếu không thành công thông báo cho người dùng biết.
Pre-condition	

Bảng 0.9 Use case thêm mới báo thức

3.2.3. Use case chỉnh sửa báo thức

Use Case	Edit Alarm
Actor	User
Brief Description	Chỉnh sửa báo thức

Main Flow	1. Người dùng chọn 1 báo thức đã cài đặt, mở màn hình EditAlarmActivity. 2. Người dùng chỉnh sửa thông tin báo thức. 3. Click save button 4. Hệ thống update vào cơ sở dữ liệu, start service cài đặt báo thức gần nhất. Trở về màn hình MainActivity
Alternative Flows	4.1. Nếu không thành công thông báo cho người dùng biết.
Pre-condition	

Bảng 0.10 Use case chỉnh sửa báo thức

3.2.4. Use case xóa báo thức

Use Case	Delete Alarm
Actor	User
Brief Description	Xóa báo thức
Main Flow	1. Người dùng chọn 1 báo thức đã cài đặt, mở màn hình EditAlarmActivity. 2. Click delete button. 3. Hệ thống kết nối cơ sở dữ liệu và xóa báo thức, start service cài đặt báo thức gần nhất. Trở về màn hình MainActivity
Alternative Flows	3.1 Nếu không thành công, thông báo cho người dùng
Pre-condition	

Bảng 0.11 Use case xóa báo thức

3.2.5. Use case tắt báo thức

Use Case	Stop Alarm
Actor	User
Brief Description	Tắt báo thức

Main Flow	<ol style="list-style-type: none">1. Đến thời gian báo thức, hệ thống tự động mở màn hình StopAlarm và phát âm báo thức báo hiệu cho người dùng.2. Người dùng click stop button. Nếu báo thức là mặc định thì ngừng báo thức. Nếu báo thức là “Music”, mở màn hình AudioActivity để người dùng chơi minigame, tương tự nếu là “Photo” thì mở màn hình PhotoActivity để người dùng chơi minigame.3. Tại màn hình AudioActivity, hệ thống sẽ phát 1 từng bản nhạc bất kì trong điện thoại và hiển thị câu hỏi và đáp án cho người dùng lựa chọn. Khi người dùng hoàn thành trò chơi, thông báo cho người dùng và thoát khỏi màn hình.4. Tương tự, tại màn hình PhotoActivity, ứng dụng hiển thị các bức ảnh trong thư viện ảnh của điện thoại, người dùng sẽ trả lời các câu hỏi liên quan đến những bức ảnh để tắt báo thức.5. Người dùng trả lời cho đến khi hoàn thành điều kiện. Tắt âm báo và đồng thời start Service để cài đặt báo thức mới gần nhất.
Alternative Flows	
Pre-condition	

Bảng 0.12 Use case tắt báo thức

3.2.6. Use case thay đổi trạng thái on/off của báo thức

Use Case	Change State Alarm
Actor	User
Brief Description	Thay đổi trạng thái báo thức (on/off)
Main Flow	<ol style="list-style-type: none">1. Người dùng vuốt thay đổi trạng thái on/off của từng báo thức trên danh sách báo thức.2. Hệ thống update cơ sở dữ liệu, sau đó start Service cài đặt báo thức mới gần nhất cho ứng dụng.
Alternative Flows	

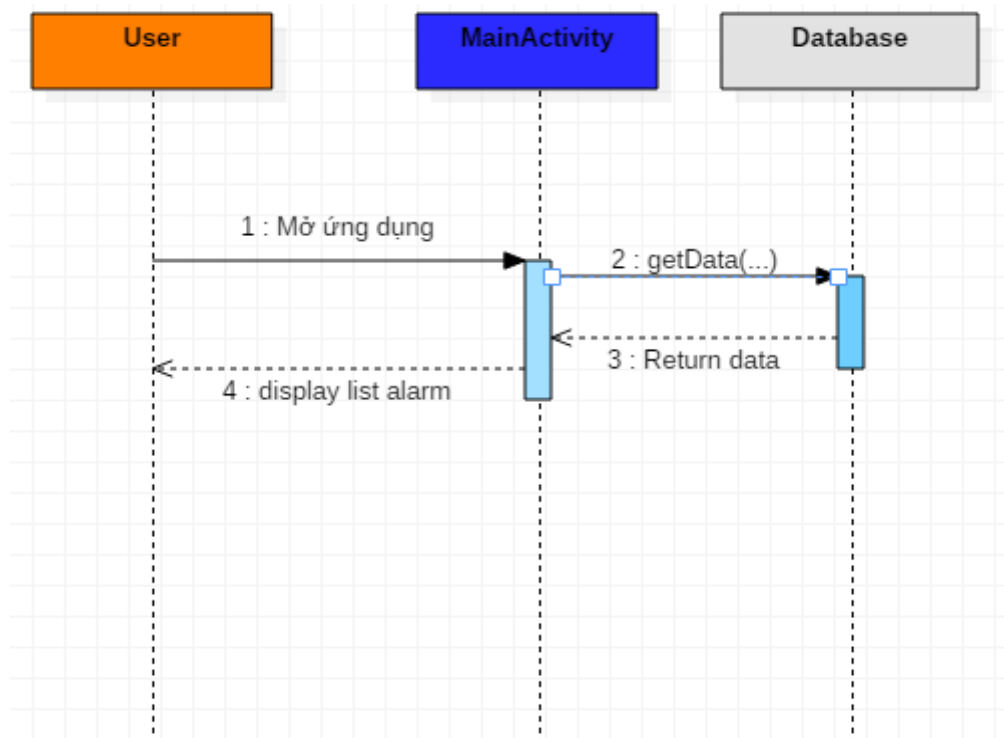
Pre-condition	
---------------	--

Bảng 0.13 Use case thay đổi trạng thái báo thức

3.3. Biểu đồ tuần tự

Biểu đồ trong phần này sẽ nói rõ hơn về quá trình thực hiện các chức năng của chương trình. Sử dụng biểu đồ này sẽ giúp cho người lập trình đơn giản hơn trong quá trình cài đặt các thiết kế.

3.3.1. Biểu đồ tuần tự chức năng hiển thị danh sách báo thức

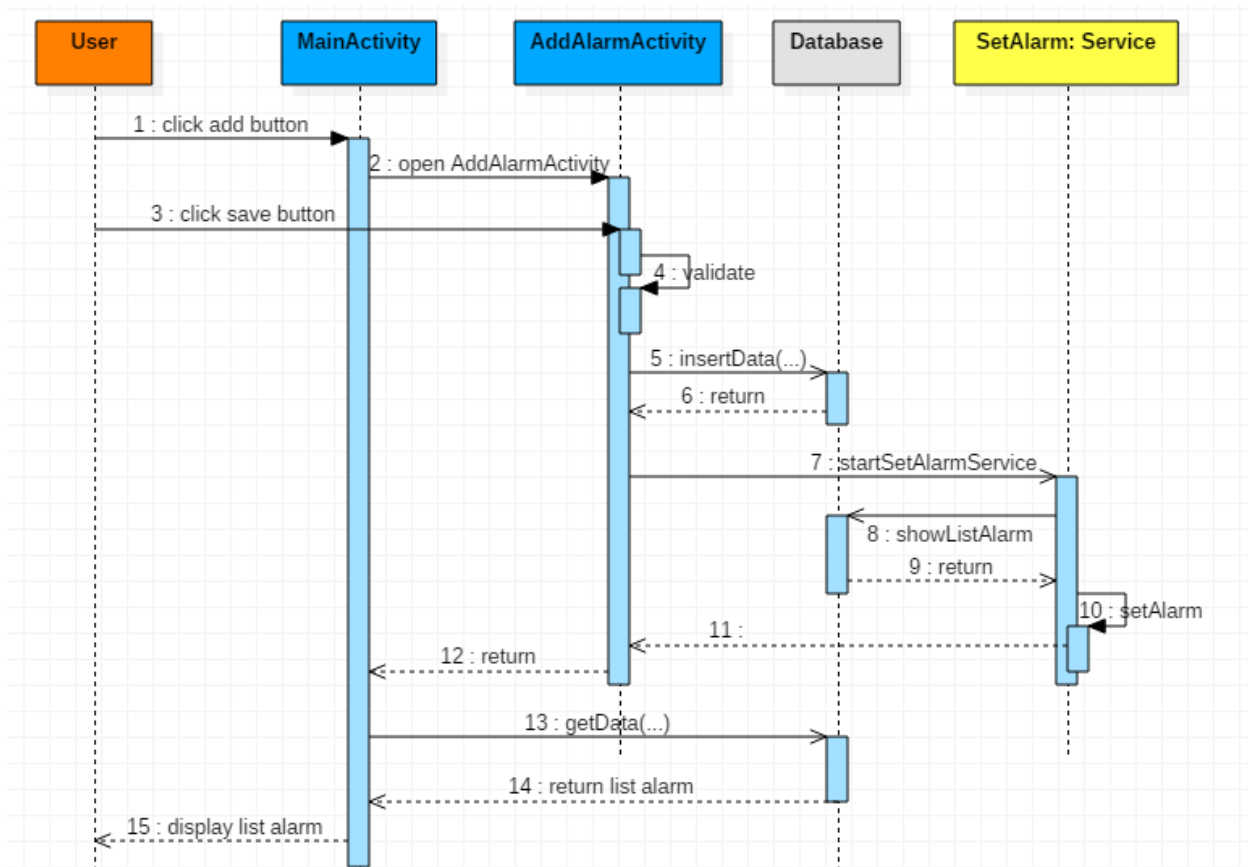


Hình 0.2 Biểu đồ tuần tự chức năng hiển thị danh sách báo thức

Các bước được thực hiện tuần tự như sau:

- 1. User vào ứng dụng
- 2-3-4: lấy tất cả các báo thức, trả kết quả, gán adapter vào RecyclerView và hiển thị ra màn hình

3.3.2. Biểu đồ tuần tự chức năng thêm mới báo thức

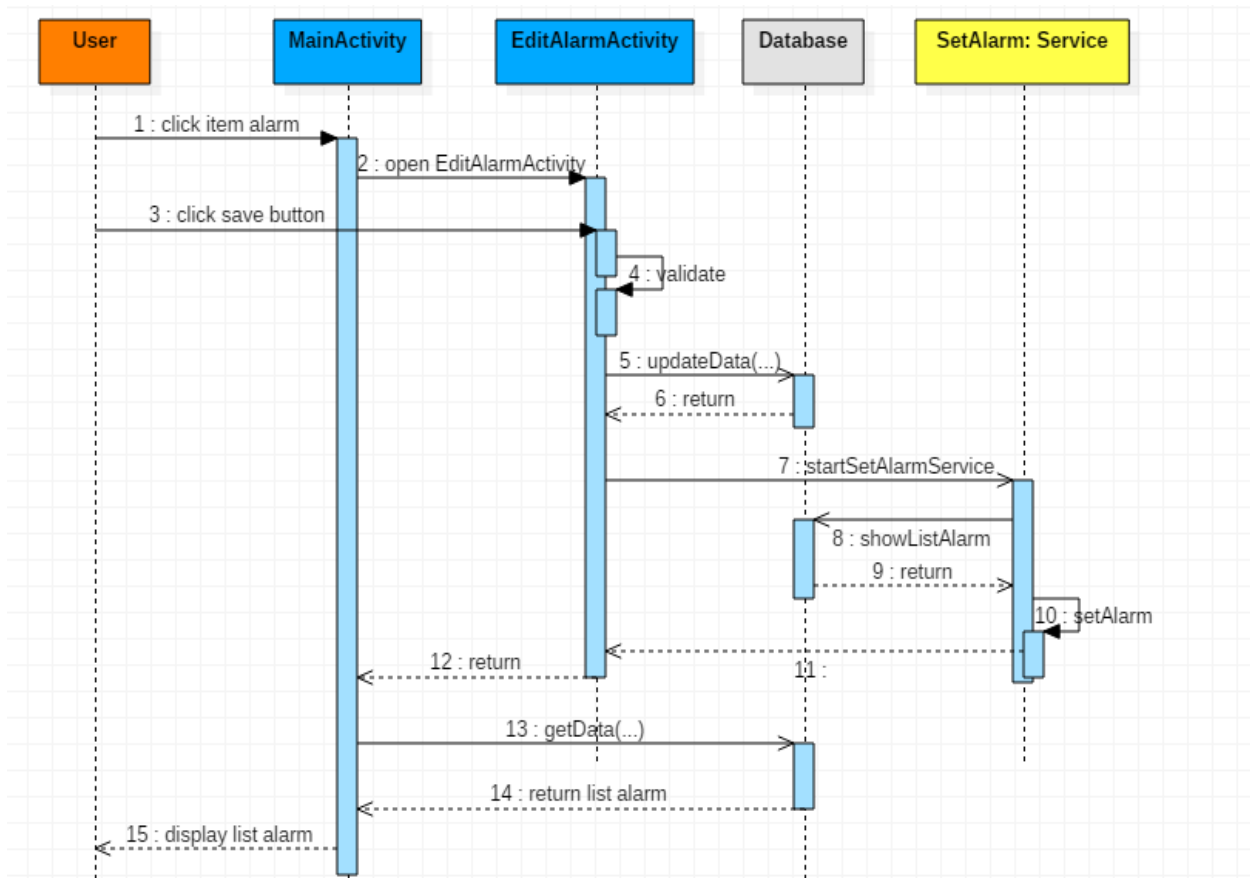


Hình 0.3 Biểu đồ tuần tự chức năng thêm mới báo thức

Các bước được thực hiện tuần tự như sau:

- 1-2-3 người dùng click add button, mở màn hình AddAlarmActivity, cài đặt thông tin báo thức và click save button.
- 4-5-6: hệ thống kiểm tra thông tin cài đặt báo thức và lưu vào cơ sở dữ liệu.
- 7-8-9-10-11: start Service thực hiện cài đặt báo thức gần nhất cho điện thoại.
- 12-13-14-15: trở về màn hình MainActivity, truy vấn cơ sở dữ liệu lấy danh sách báo thức và hiển thị ra màn hình

3.3.3. Biểu đồ tuần tự chức năng chỉnh sửa báo thức

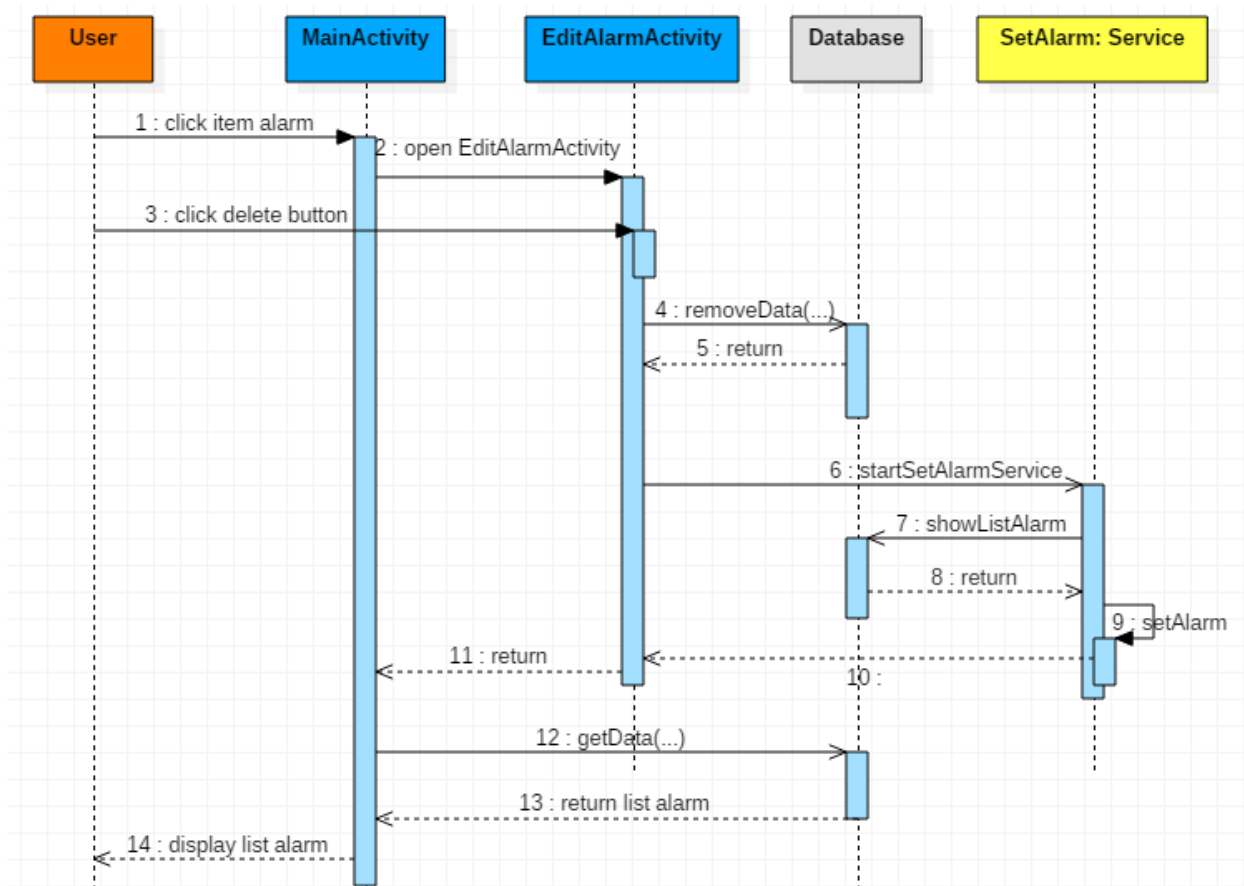


Hình 0.4 Biểu đồ tuần tự chức năng chỉnh sửa báo thức

Các bước được thực hiện tuần tự như sau:

- 1-2-3 người dùng click chọn một báo thức trong danh sách báo thức trên màn hình MainActivity, mở màn hình EditAlarmActivity, chỉnh sửa thông tin báo thức và click save button.
- 4-5-6: hệ thống kiểm tra thông tin cài đặt báo thức và cập nhật vào cơ sở dữ liệu.
- 7-8-9-10-11: start Service thực hiện cài đặt báo thức gần nhất cho điện thoại.
- 12-13-14-15: trở về màn hình MainActivity, truy vấn cơ sở dữ liệu lấy danh sách báo thức và hiển thị ra màn hình

3.3.4 Biểu đồ tuần tự chức năng xóa báo thức

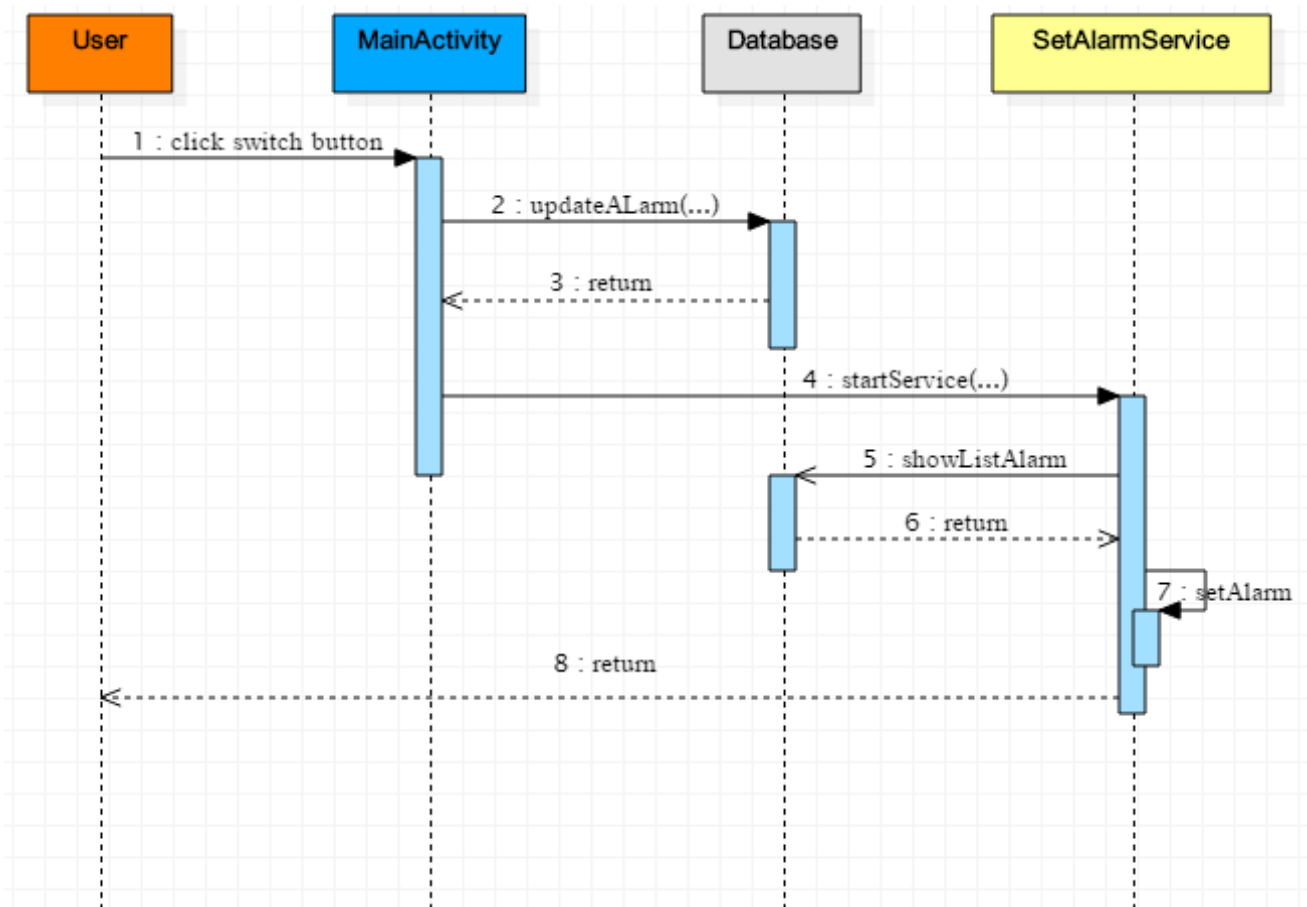


Hình 0.5 Biểu đồ tuần tự chức năng xóa báo thức

Các bước được thực hiện tuần tự như sau:

- 1-2-3 Người dùng click chọn một báo thức trong danh sách báo thức trên màn hình MainActivity, mở màn hình EditAlarmActivity, click delete button.
- 4-5: Hệ thống xóa báo thức trong cơ sở dữ liệu
- 6-7-8-9-10: start Service thực hiện cài đặt báo thức gần nhất cho điện thoại.
- 11-12-13-14: trở về màn hình MainActivity, truy vấn cơ sở dữ liệu lấy danh sách báo thức và hiển thị ra màn hình

3.3.5. Biểu đồ tuần tự chức năng thay đổi trạng thái on/off báo báo thức

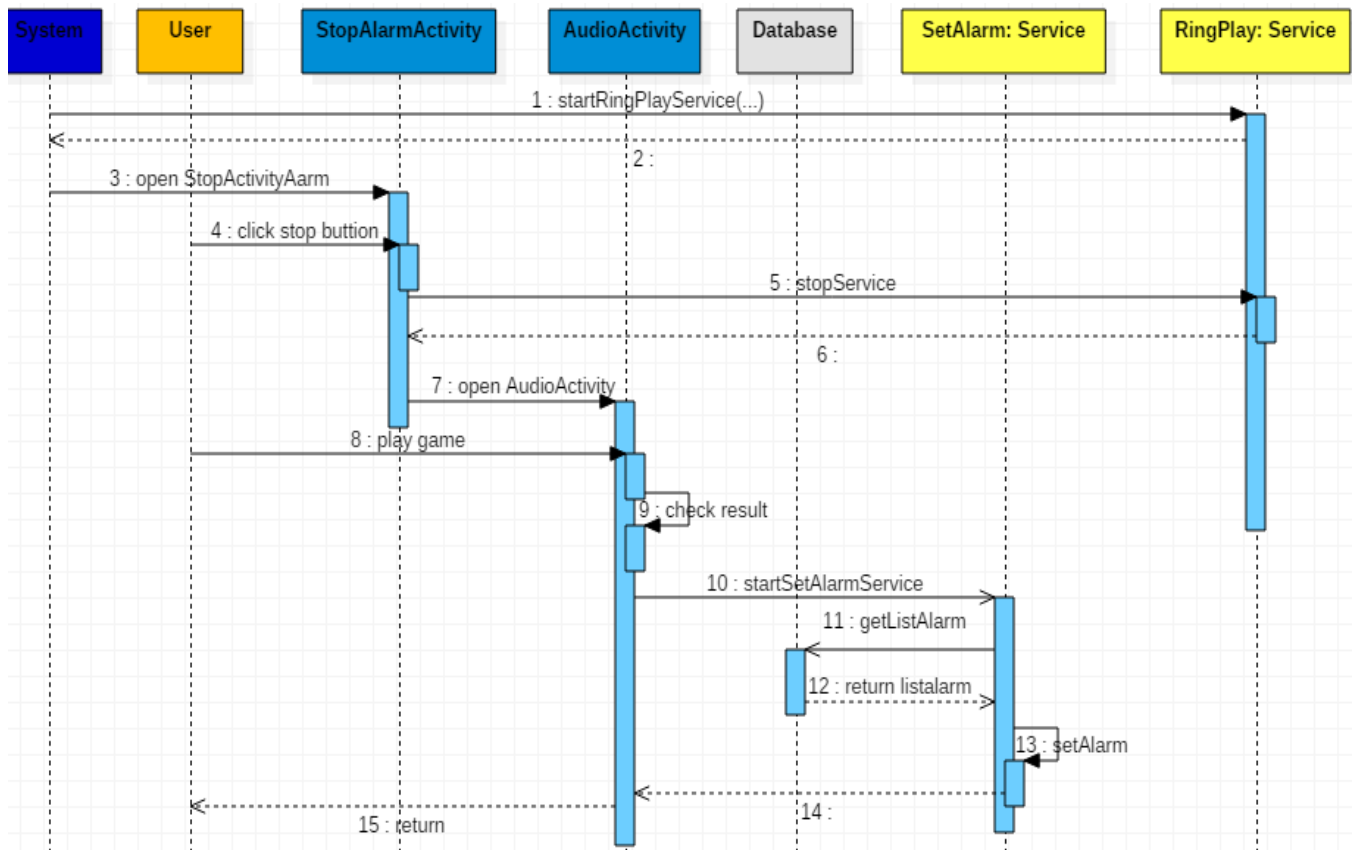


Hình 0.6 Biểu đồ tuần tự chức năng thay đổi trạng thái báo thức

Các bước được thực hiện tuần tự như sau:

- 1. Người dùng click SwitchButton từng item báo thức để thay đổi trạng thái on/off
- 2-3: Hệ thống cập nhật vào cơ sở dữ liệu
- 4-5-6-7-8: start Service thực hiện cài đặt báo thức gần nhất cho điện thoại, báo kết quả cho người dùng

3.3.6. Biểu đồ tuần tự chức năng nghe nhạc tắt báo thức

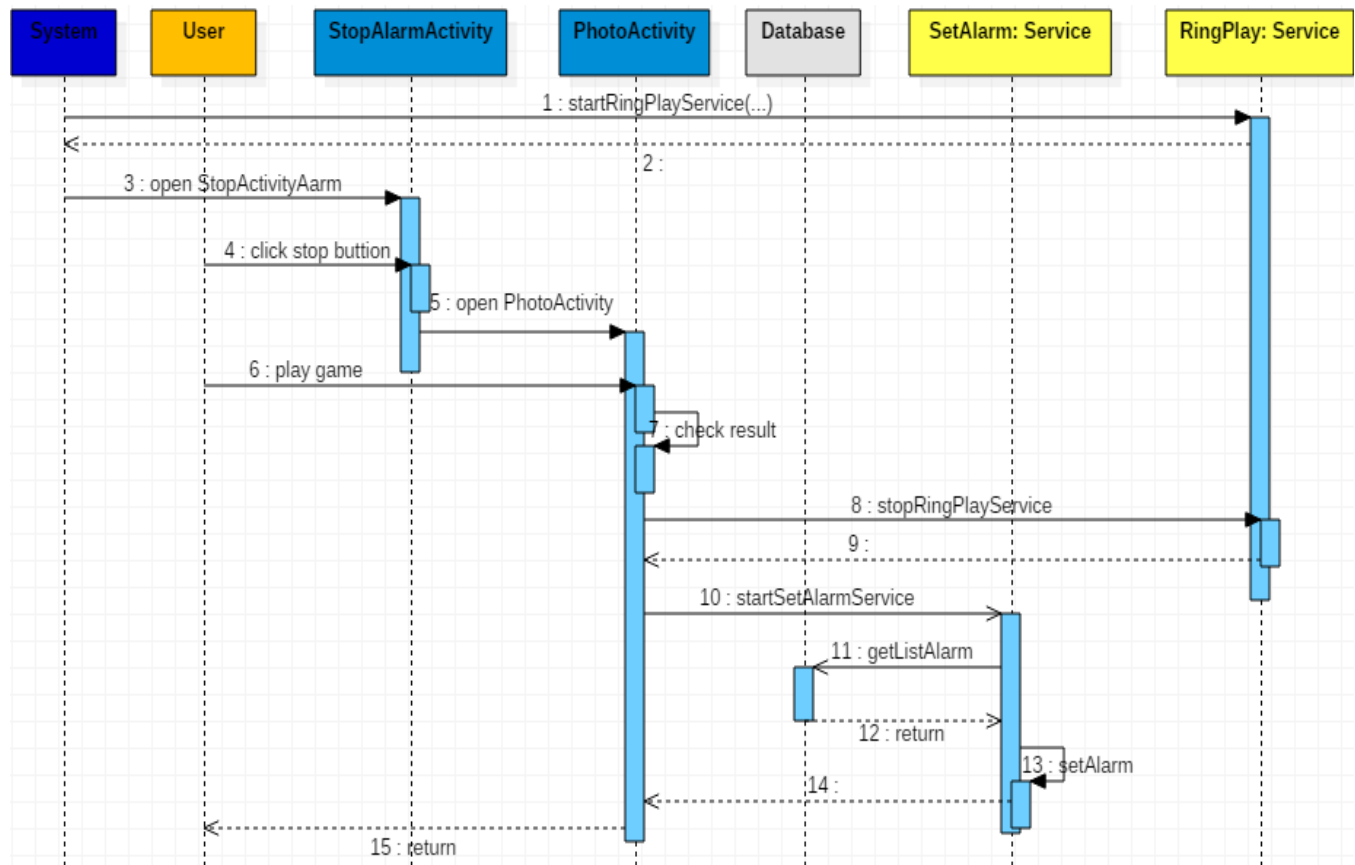


Hình 0.7 Biểu đồ tuần tự chức năng nghe nhạc tắt báo thức

Các bước được thực hiện tuần tự như sau:

- 1-2-3: Tới thời gian báo thức, hệ thống tự động gọi service phát âm báo thức, đồng thời mở màn hình StopAlarmActivity.
- 4-5-6-7: Người dùng click stop button tắt âm báo thức, mở AudioActivity
- 8-9: Tại màn hình này, hệ thống phát lần lượt từng bản nhạc ngẫu nhiên lấy từ điện thoại và hiển thị các đáp án cho người dùng lựa chọn tên bản nhạc. Người dùng lựa chọn đáp án, hệ thống kiểm tra kết quả và thông báo cho người dùng.
- 10-11-12-13-14-15: Khi hoàn thành điều kiện tắt báo thức, tiếp tục start Service để cài đặt báo thức mới cho điện thoại. Thông báo cho người dùng và thoát khỏi màn hình.

3.3.7. Biểu đồ tuần tự chức năng xem ảnh tắt báo thức

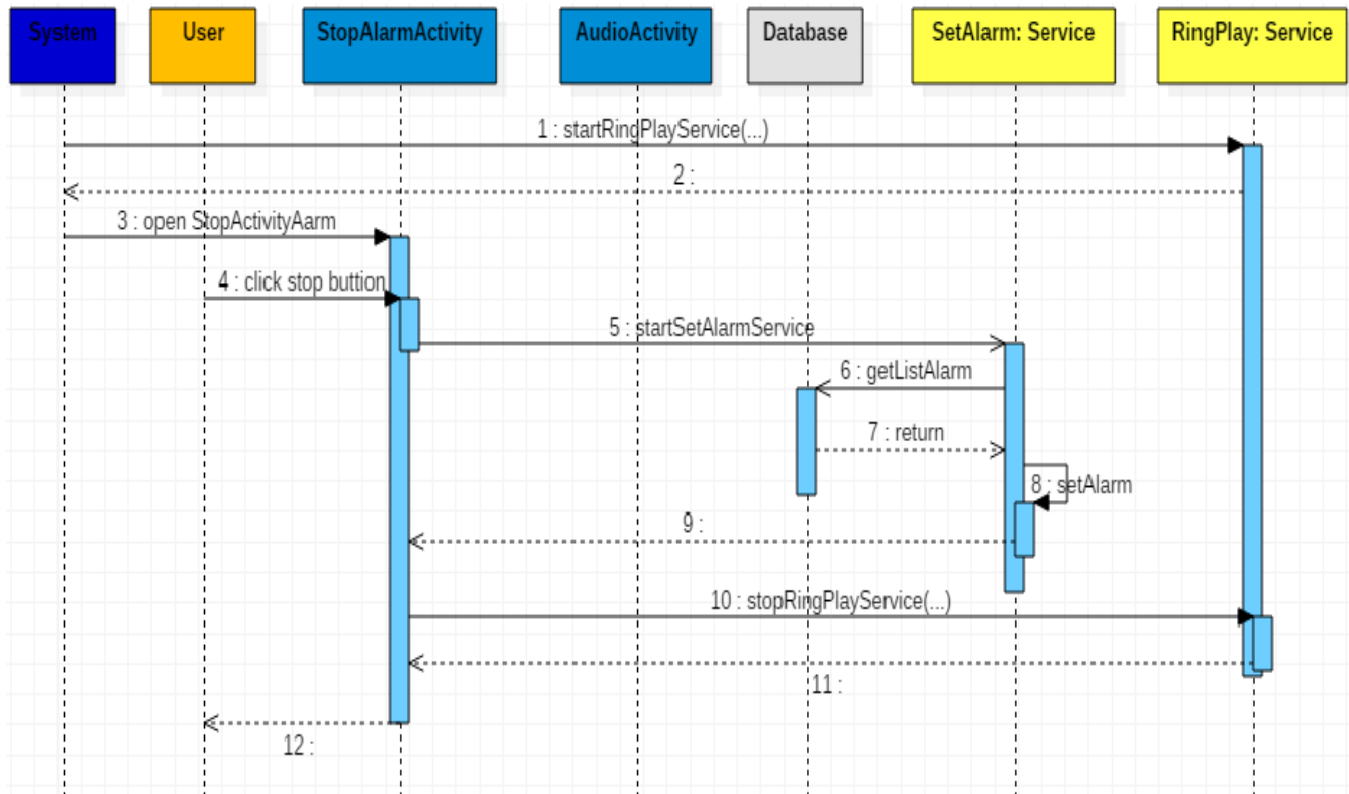


Hình 0.8 Biểu đồ tuần tự chức năng xem ảnh tắt báo thức

Các bước được thực hiện tuần tự như sau:

- 1-2-3: Tới thời gian báo thức, hệ thống tự động gọi service phát âm báo thức, đồng thời mở màn hình StopAlarmActivity.
- 4-5: Người dùng click stop button tắt âm báo thức, mở màn hình AudioActivity.
- 6-7: Người dùng trả lời những câu hỏi liên quan đến các bức ảnh trong thư viện ảnh của điện thoại. Hệ thống kiểm tra câu trả lời.
- 8-9-10-11-12-13-14-15: Khi hoàn thành điều kiện tắt báo thức, tắt báo thức và start Service để cài đặt báo thức mới cho điện thoại. Thông báo cho người dùng và thoát khỏi màn hình.

3.3.8. Biểu đồ tuần tự chức năng tắt báo thức mặc định



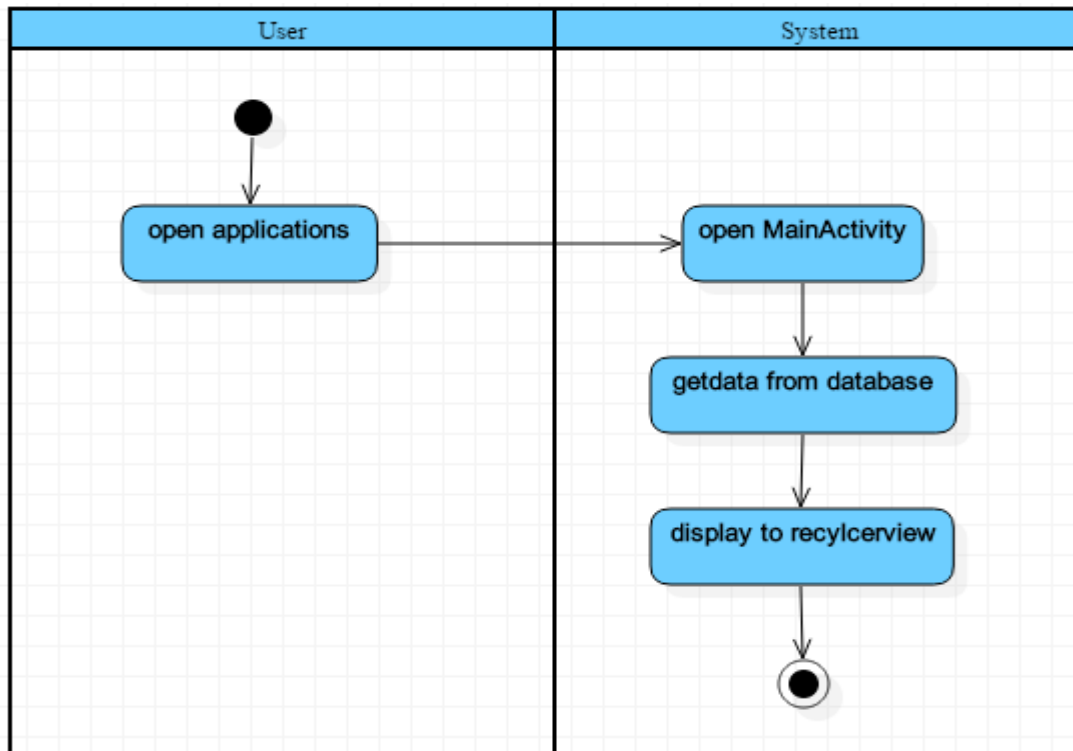
Hình 0.9 Biểu đồ tuần tự chức năng tắt báo thức mặc định

Các bước được thực hiện tuần tự như sau:

- 1-2-3: Tới thời gian báo thức, hệ thống tự động gọi service phát âm báo thức, đồng thời mở màn hình StopAlarmActivity.
- 4-5-6-7-8-9-10-11-12: Người dùng click stop button. Tắt âm báo thức, start Service cài đặt báo thức mới gần nhất cho điện thoại. Thông báo cho người dùng và thoát khỏi màn hình.

3.4 Biểu đồ hoạt động

3.4.1 .Biểu đồ hoạt động cho chức năng xem danh sách báo thức

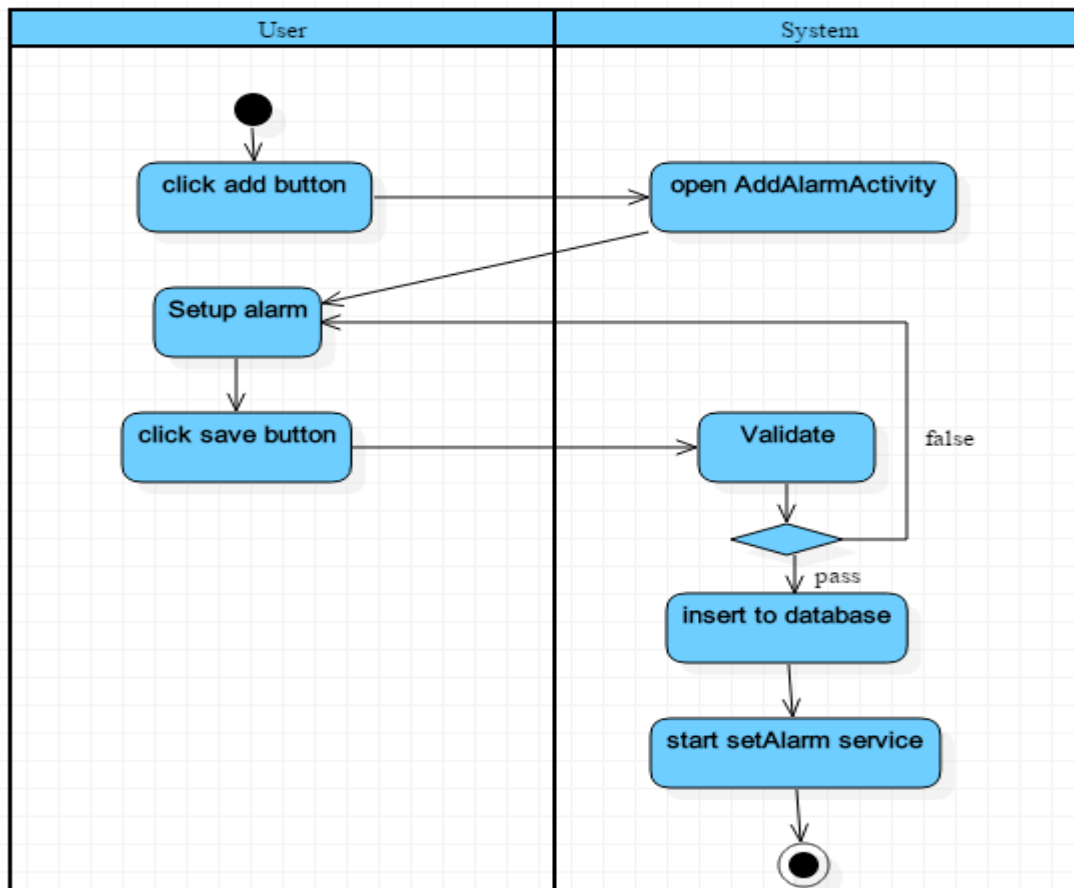


Hình 0.10 Biểu đồ hoạt động chức năng xem danh sách báo thức

Mô tả:

- Người dùng mở ứng dụng.
- Hệ thống truy vấn CSDL và trả về danh sách báo thức
- Hiển thị danh sách báo thức cho người dùng.

3.4.2 Biểu đồ hoạt động cho chức năng thêm mới báo thức

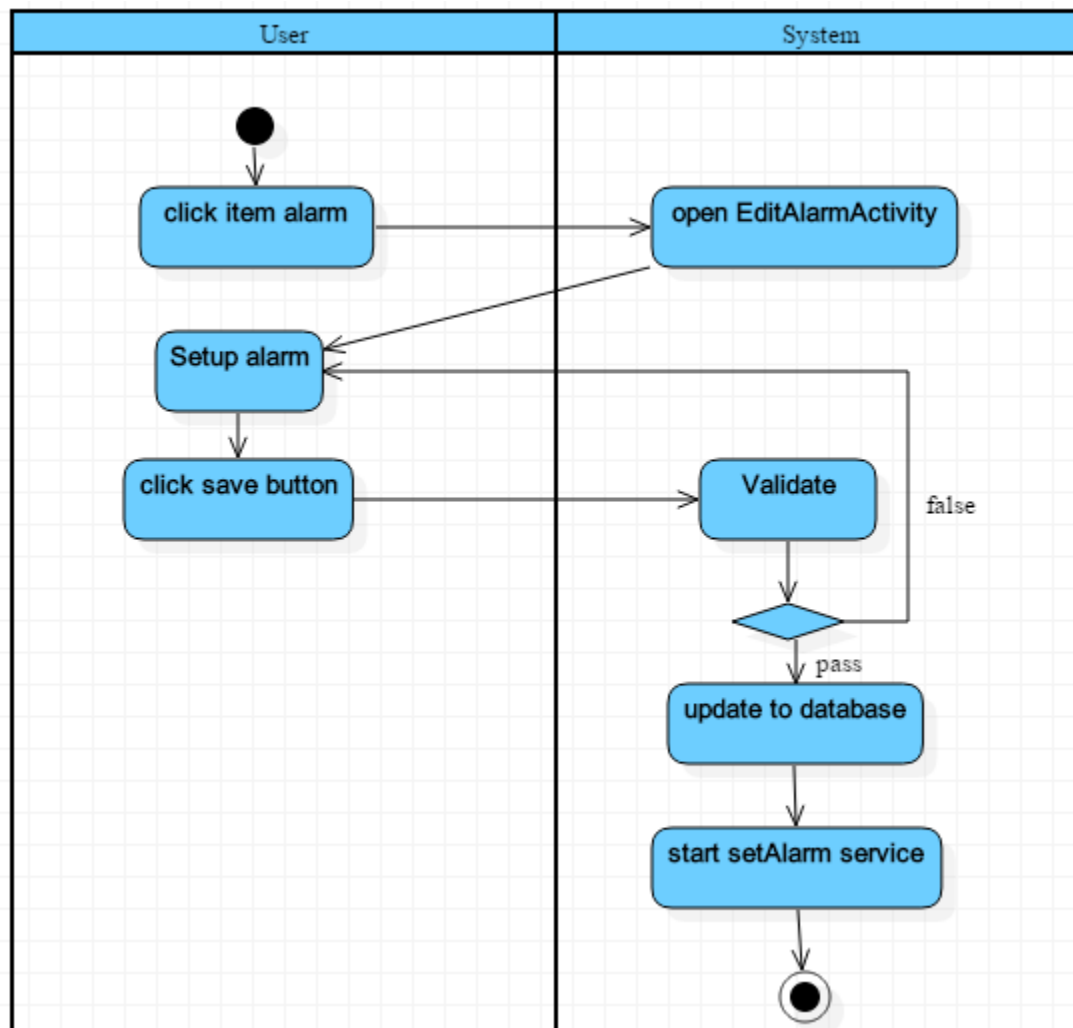


Hình 0.11 Biểu đồ hoạt động chức năng thêm mới báo thức

Mô tả:

- Người dùng click add button, mở màn hình AddAlarmActivity, cài đặt báo thức và click save button.
- Hệ thống thêm báo thức vào cơ sở dữ liệu, sau đó chạy service SetAlarmService để cài đặt báo thức gần nhất cho điện thoại.
- Trở về màn hình MainActivity

3.4.3 Biểu đồ hoạt động cho chức năng sửa báo thức

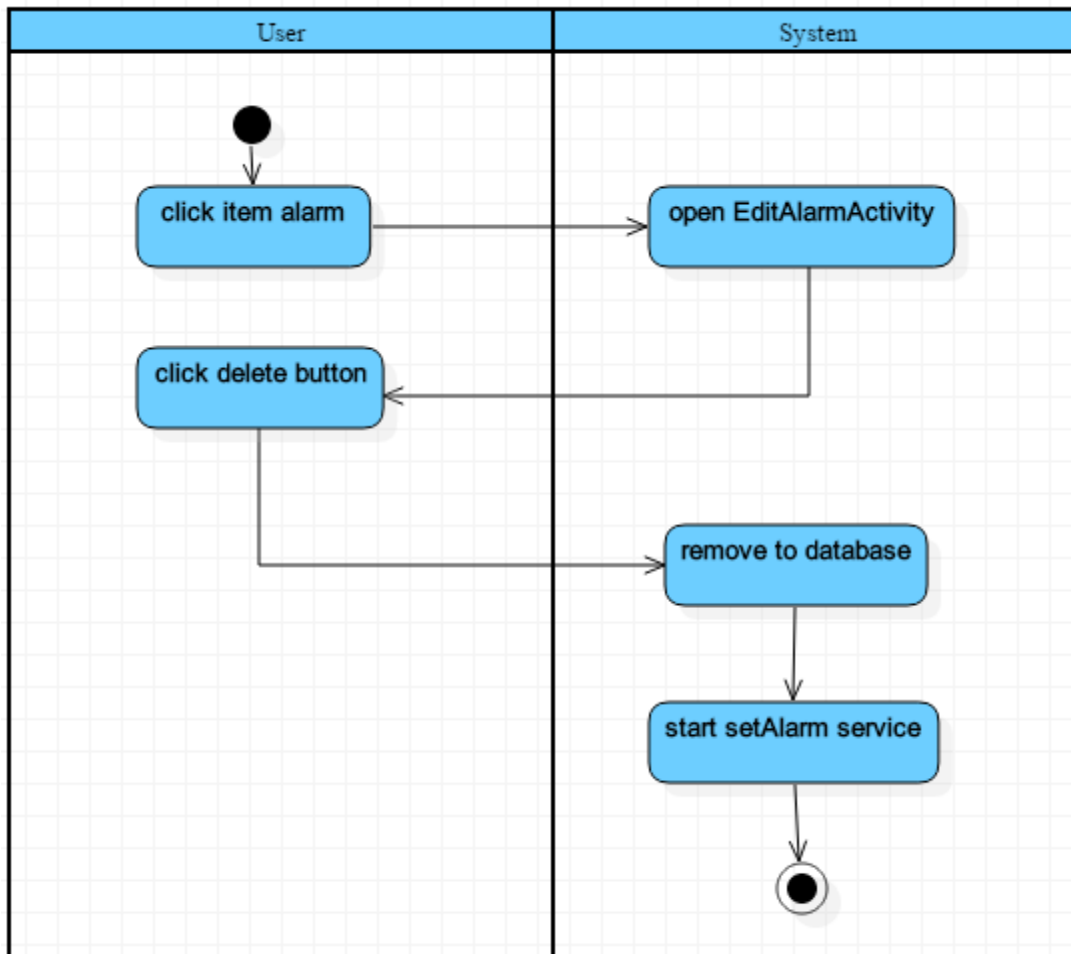


Hình 0.12 Biểu đồ hoạt động chức năng xem sửa báo thức

Mô tả:

- Người dùng chọn một báo thức, mở màn hình EditAlarmActivity, chỉnh sửa lại cài đặt báo thức và click save button.
- Hệ thống update vào cơ sở dữ liệu, sau đó chạy service SetAlarmService cài đặt báo thức gần nhất cho điện thoại.
- Trở về màn hình MainActivity

3.4.4 Biểu đồ hoạt động cho chức năng xóa báo thức

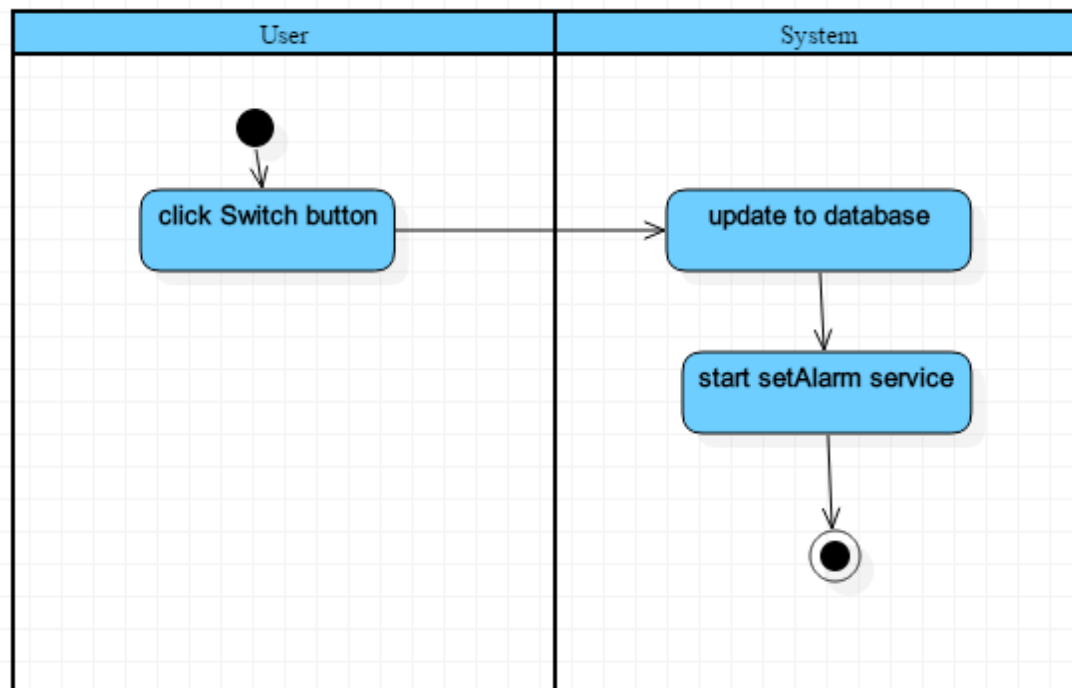


Hình 0.13 Biểu đồ hoạt động chức năng xóa báo thức

Mô tả:

- Người dùng chọn một báo thức, mở màn hình EditAlarmActivity và click delete button.
- Hệ thống xóa báo thức trong cơ sở dữ liệu, sau đó chạy service SetAlarmService cài đặt báo thức gần nhất cho điện thoại.
- Trở về màn hình MainActivity

3.4.5. Biểu đồ hoạt động cho chức năng thay đổi trạng thái báo thức

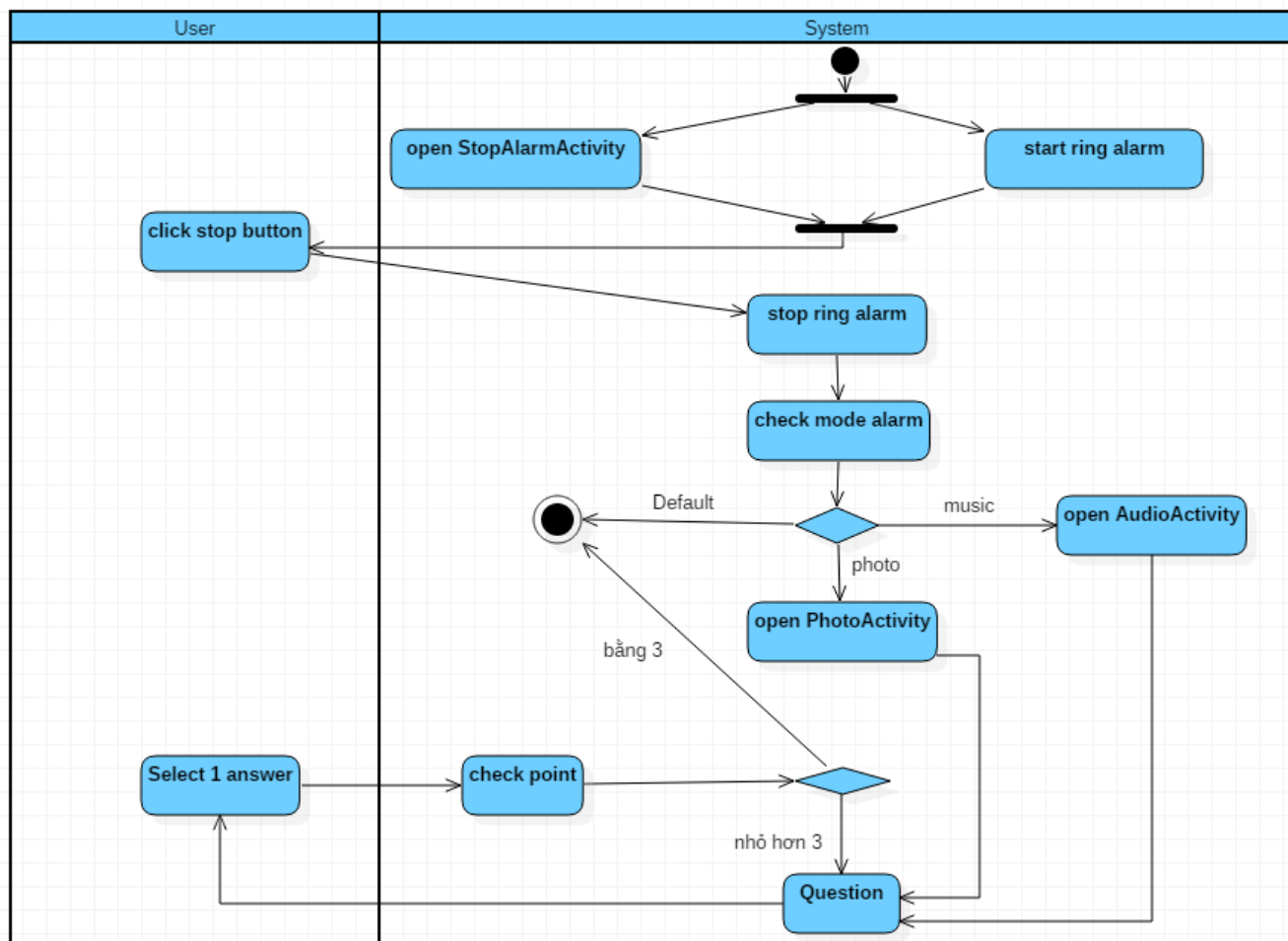


Hình 0.14 Biểu đồ hoạt động chức năng thay đổi trạng thái báo thức

Mô tả:

- Người dùng click Switch button ứng với mỗi báo thức để thay đổi trạng thái on/off.
- Ứng dụng update vào cơ sở dữ liệu, sau đó chạy service SetAlarmService để cài đặt báo thức gần nhất cho điện thoại

3.4.6 Biểu đồ hoạt động cho chức năng tắt báo thức



Hình 0.15 Biểu đồ hoạt động chức năng tắt báo thức

Mô tả:

- Khi đến thời gian báo thức, ứng dụng mở âm báo thức, đồng thời mở màn hình StopAlarmActivity
- Người dùng click stop button.
- Nếu báo thức là loại “Default”, hệ thống sẽ tắt âm báo và finish. Nếu báo thức là loại “Music”, tắt âm báo thức, mở tiếp màn hình AudioActivity để người dùng chơi mini game, tương tự nếu là loại “Photo” , mở màn hình PhotoActivity.
- Người dùng trả lời các câu hỏi về âm nhạc, hình ảnh trong thư viện điện thoại. Khi hoàn hành điều kiện, hệ thống start Service cài đặt báo thức tiếp theo cho điện thoại.

3.5 Chi tiết thành phần, thuật toán sử dụng.

3.5.1 PendingIntent

PendingIntent là một loại Intent đặc biệt. Một Intent bình thường sẽ chỉ tồn tại từ khi được khởi tạo cho đến khi kết thúc nhiệm vụ của mình. PendingIntent có thể tồn tại trong suốt ứng dụng và ngay cả khi đã thoát ứng dụng.

Trong ứng dụng này, PendingIntent thực hiện gửi broadcast tới BroadcastReceiver để yêu cầu phát âm báo thức. Để biết khi nào nó được gửi, dùng AlarmManager.

3.5.2 AlarmManager

AlarmManager là thành phần không thể thiếu trong một ứng dụng báo thức. Android AlarmManager cho phép bạn truy cập vào hệ thống báo động. Với sự trợ giúp này, ta có thể lên lịch cho ứng dụng chạy hoặc làm bất cứ một hành động nào khác vào một thời điểm cụ thể trong tương lai. Nó hoạt động ngay cả khi thoát ứng dụng. Android AlarmManager chứa một khóa sau CPU cung cấp bảo lãnh không ngủ cho đến khi điện thoại phát sóng được sử lý.

Trong ứng dụng này, AlarmManager dùng để hẹn giờ gửi PendingIntent chứa thông tin báo thức tới BroadcastReceiver.

3.5.3 BroadcastReceiver

Broadcast Receiver là một trong các thành phần chính của android, có thể hiểu nó như một bộ thu các bản tin cần thiết cho apps. Các bản tin được thu ở đây chính là các Intent. Có thể thu các Intent sẵn có của hệ điều hành ví dụ như: tin nhắn đến, cuộc gọi đến, trạng thái của điện thoại, trạng thái của pin... Ngoài ra các ứng dụng cũng có thể thu Intent của chính các chúng ta gửi ra để làm một nhiệm vụ gì đó (có thể dùng để start service khi cần).

Trong ứng dụng này, Broadcast Receiver sẽ nhận Intent được gửi bởi AlarmManager. Khi nhận được thông tin về báo thức, Broadcast Receiver sẽ kiểm tra yêu cầu và khởi động service phát âm báo thức, đồng thời mở màn hình báo thức.

Để sử dụng được Broadcast Receiver, cần khai báo chúng trong Manifest.

3.5.4 Service

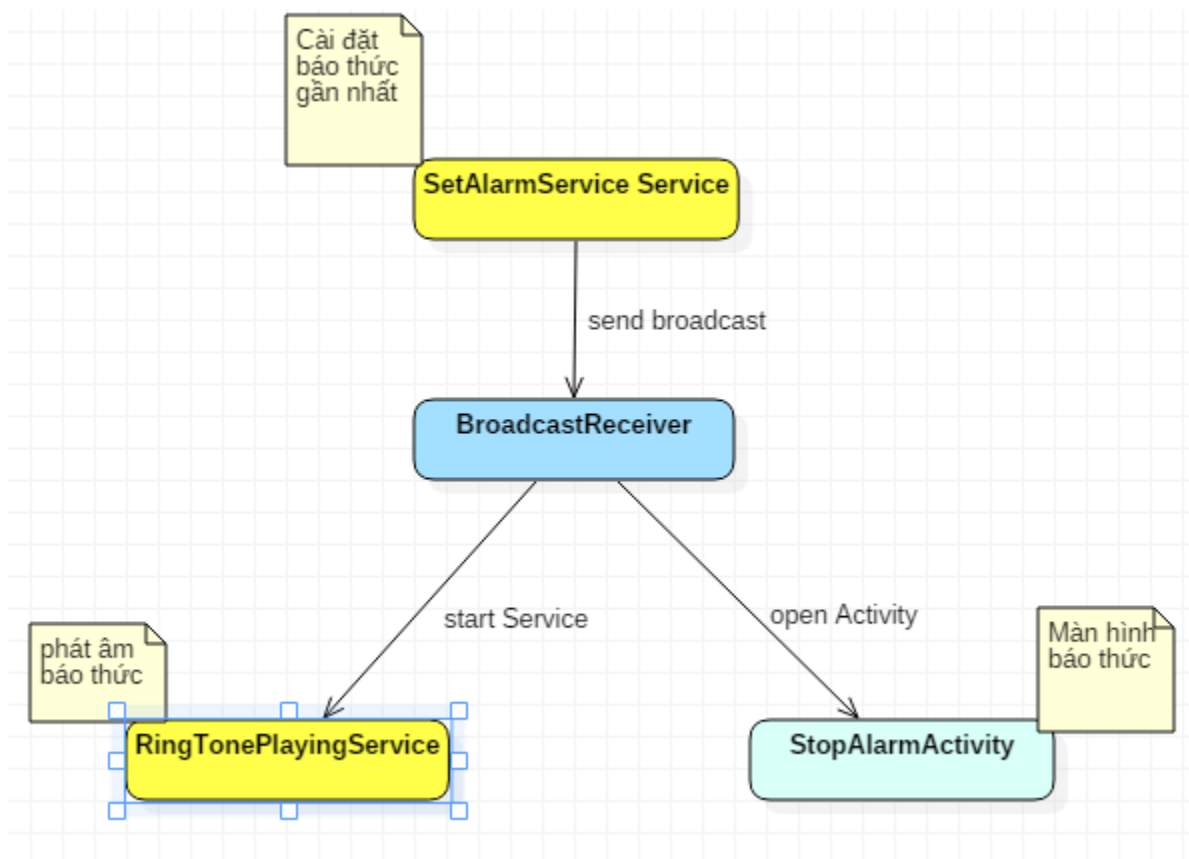
Service là một thành phần được chạy bên trong nền để xử lý các công việc trong thời gian dài. Một ứng dụng nghe nhạc có thể phát nhạc, trong khi đó người dùng đang ở giao diện của ứng

dụng khác. Hoặc ứng dụng download có thể tải dữ liệu trên mạng về máy mà không ngăn chặn người dùng tương tác với các ứng dụng khác...

Để sử dụng được Service, cần khai báo chúng trong Manifest

Trong ứng dụng này, khi Broadcast Receiver nhận được yêu cầu, nó sẽ start Service có nhiệm vụ phát âm báo thức. Hàm onStartCommand() của service sẽ được thực thi. Hàm này sẽ thực hiện kiểm tra yêu cầu âm báo cũng như trạng thái âm báo để thực hiện start hoặc stop âm báo.

3.5.5 Chi tiết hệ thống cài đặt và phát báo thức.



Hình 0.16 Quá trình cài đặt báo thức mới nhất

Mô tả:

Service SetAlarmService

- Service SetAlarmService: Đây là một service có nhiệm vụ lấy danh sách các báo thức trong cơ sở dữ liệu, thực hiện thuật toán tìm ra báo thức gần nhất trong tương lai để cài đặt thời gian cho AlarmManager. Khi đã tìm ra báo thức gần nhất, cài đặt AlarmManager đến đúng thời gian báo thức, Intent mang thông tin báo thức sẽ được gửi tới Broadcast Receiver.
- Như vậy khi mỗi người dùng thêm mới, chỉnh sửa, xóa, thay đổi trạng thái một báo thức, hoặc sau khi người dùng tắt một báo thức, hệ thống sẽ lưu/cập nhật vào cơ sở dữ liệu, sau đó sẽ start Service này để cài đặt báo thức.

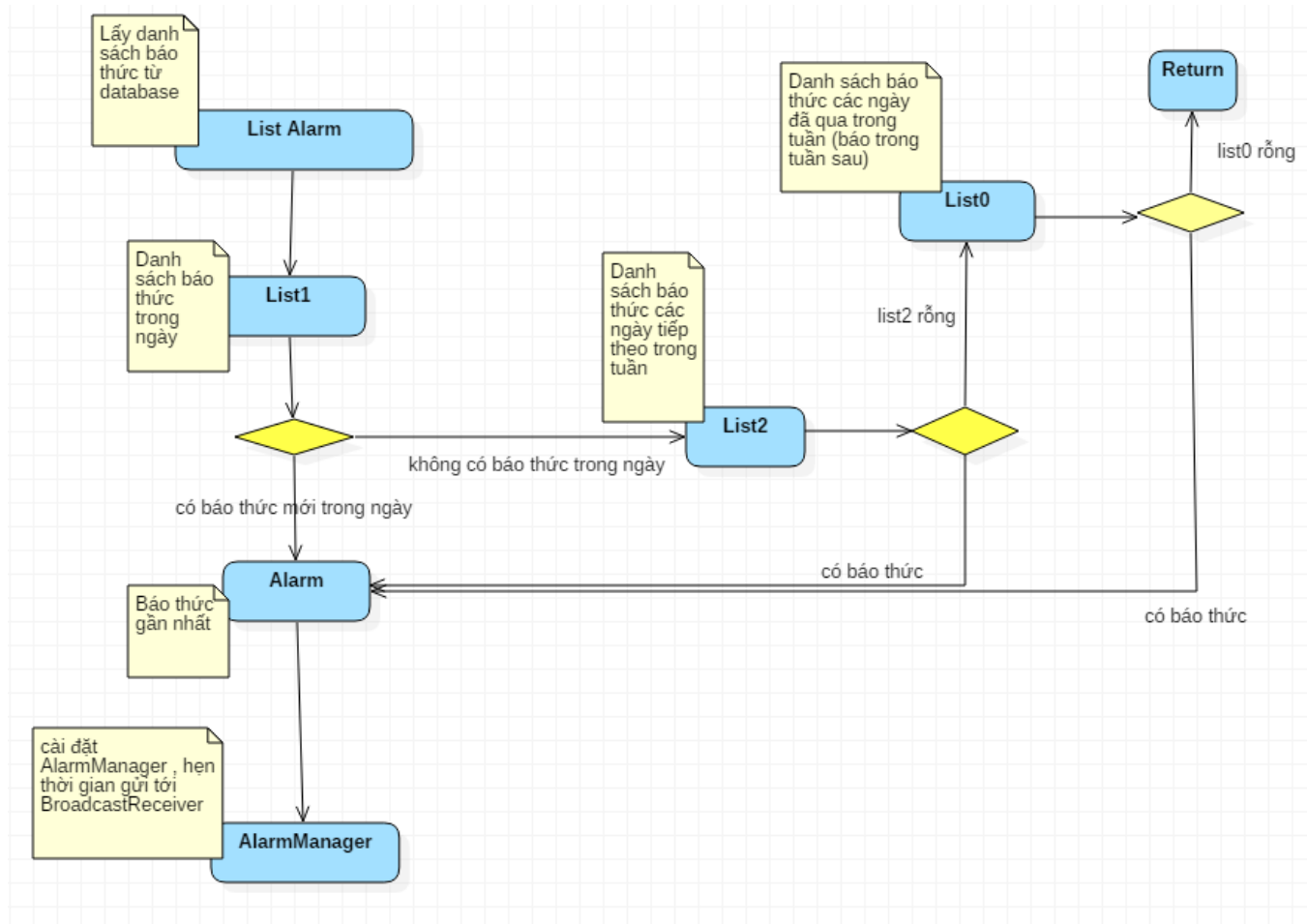
Broadcast Receiver

- Thành phần này nhận Intent được gửi từ Service phía trên, tại đây sẽ thực hiện kiểm tra thông tin về báo thức để start Service phát âm báo thức cũng như mở màn hình báo thức tương ứng.

RingtonePlayingService:

- Đây là service được khởi chạy từ thành phần Broadcast Receiver phía trên. Tại đây hệ thống thực hiện phát/ngừng âm báo thức tương ứng với yêu cầu nhận được.

3.5.6 Thuật toán cài đặt báo thức gần nhất



Hình 0.17 Sơ đồ thuật toán chọn báo thức gần nhất

Mô tả:

- Đầu tiên, truy vấn cơ sở dữ liệu, lấy ra danh sách các báo thức. Phân thành 3 danh sách: List0-danh sách báo thức tuần sau, List1-danh sách báo thức trong ngày, List2-danh sách báo thức các ngày tiếp theo trong tuần.
- Kiểm tra từng phần tử trong List1. Nếu phần tử là báo thức trước thời điểm hiện tại, thêm phần tử này vào List0. Nếu có báo thức sau thời điểm hiện tại, tìm ra báo thức gần thời điểm hiện tại nhất và cài đặt báo thức (tiêu chí so sánh: giờ, phút). Nếu List1 không có báo thức mới, kiểm tra List2.
- Nếu List2 có báo thức, chọn báo thức gần nhất và cài đặt báo thức (tiêu chí so sánh ngày, giờ, phút). Nếu List2 không có báo thức, kiểm tra List1.

- Nếu List1 có báo thức, chọn báo thức gần nhất và cài đặt báo thức (tiêu chí so sánh ngày, giờ, phút). Nếu List1 không có báo thức, không có báo thức tiếp theo.

3.6 Thiết kế CSDL .

Cơ sở dữ liệu gồm có 2 bảng:

- Bảng Alarm: lưu trữ danh sách báo thức

Name	DataType	Constraints	Documentation
id_alarm	Integer	PK	khóa chính
name_alarm	varchar(255)		Tiêu đề báo thức
ring_alarm	varchar(255)	NOT NULL	Âm báo thức
hour	Integer	NOT NULL	Giờ báo thức
minute	Integer	NOT NULL	Phút báo thức
arr_day	varchar(255)	NOT NULL	Danh sách ngày báo thức
state	varchar(255)	NOT NULL	Trạng thái báo thức(on/off)
mode	varchar(255)	NOT NULL	Kiểu báo thức
type	varchar(255)	NOT NULL	Loại âm báo (mặc định / từ điện thoại)
vibrate	varchar(255)	NOT NULL	Báo rung

Bảng 0.14 Bảng Alarm trong cơ sở dữ liệu

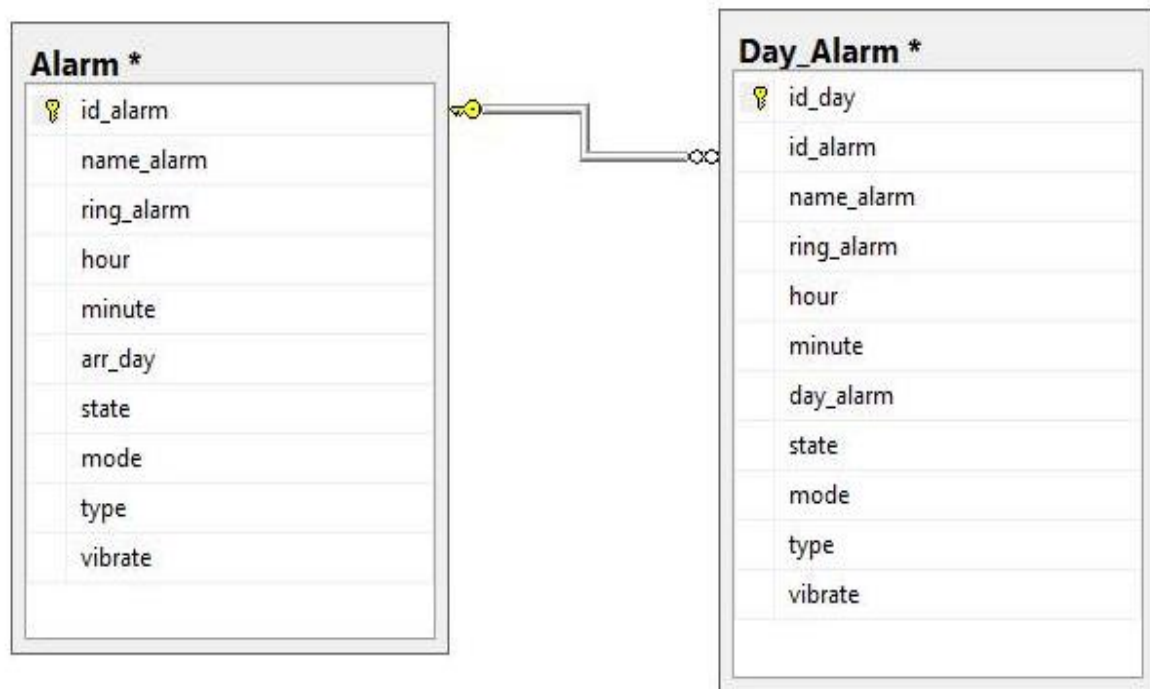
- Bảng Day_Alarm: ứng với mỗi ngày của 1 báo thức trong bảng Alarm sẽ có 1 bảng Day_Alarm, bảng này cũng lưu đầy đủ thông tin của báo thức.

Name	DataType	Constraints	Documentation
id_day	Integer	PK	Khóa chính
id_alarm	Integer	FOREIGN KEY	Khóa ngoại
name_alarm	varchar(255)		Tiêu đề báo thức
ring_alarm	varchar(255)	NOT NULL	Âm báo thức
hour	Integer	NOT NULL	Giờ báo thức
minute	Integer	NOT NULL	Phút báo thức
day_alarm	varchar(255)	NOT NULL	Ngày báo thức
state	varchar(255)	NOT NULL	Trạng thái báo thức(on/off)
mode	varchar(255)	NOT NULL	Kiểu báo thức
type	varchar(255)	NOT NULL	Loại âm báo (mặc định / từ điện thoại)
vibrate	varchar(255)	NOT NULL	Báo rung

Bảng 0.15 Bảng Day_Alarm trong cơ sở dữ liệu

Ứng với mỗi bản ghi của bảng Alarm có thể có nhiều bản ghi trong bảng Day_Alarm (tối đa là 7 bản ghi tương ứng 7 ngày trong tuần). Mỗi một ngày trong danh sách các ngày của một bản ghi ở bảng Alarm sẽ có một bản ghi tương ứng trong bảng Day_Alarm.

- Biểu đồ thiết kế CSDL



Hình 0.18 Biểu đồ thiết kế CSDL

Kết chương:

Chương này đã mô tả chi tiết về cấu trúc, phân tích thiết kế, cách thức hoạt động của ứng dụng. Chương tiếp theo sẽ mô tả cách thức triển khai và demo kịch bản ứng dụng.

CHƯƠNG IV: TRIỂN KHAI ỨNG DỤNG

Nội dung chương này nhằm trình bày những vấn đề sau

- ✓ Môi trường, cách thức triển khai và sử dụng.
- ✓ Kịch bản và hình ảnh demo.

4.1 Môi trường triển khai

- Ứng được cài đặt trên Android 5.0
- Môi trường phát triển là trên máy tính cá nhân (Desktop hoặc Laptop) sử dụng hệ điều hành Windows có cài đặt máy ảo Genymotion, công cụ Android Studio.
- Công cụ phát triển là Android Studio đã cài Android SDK.

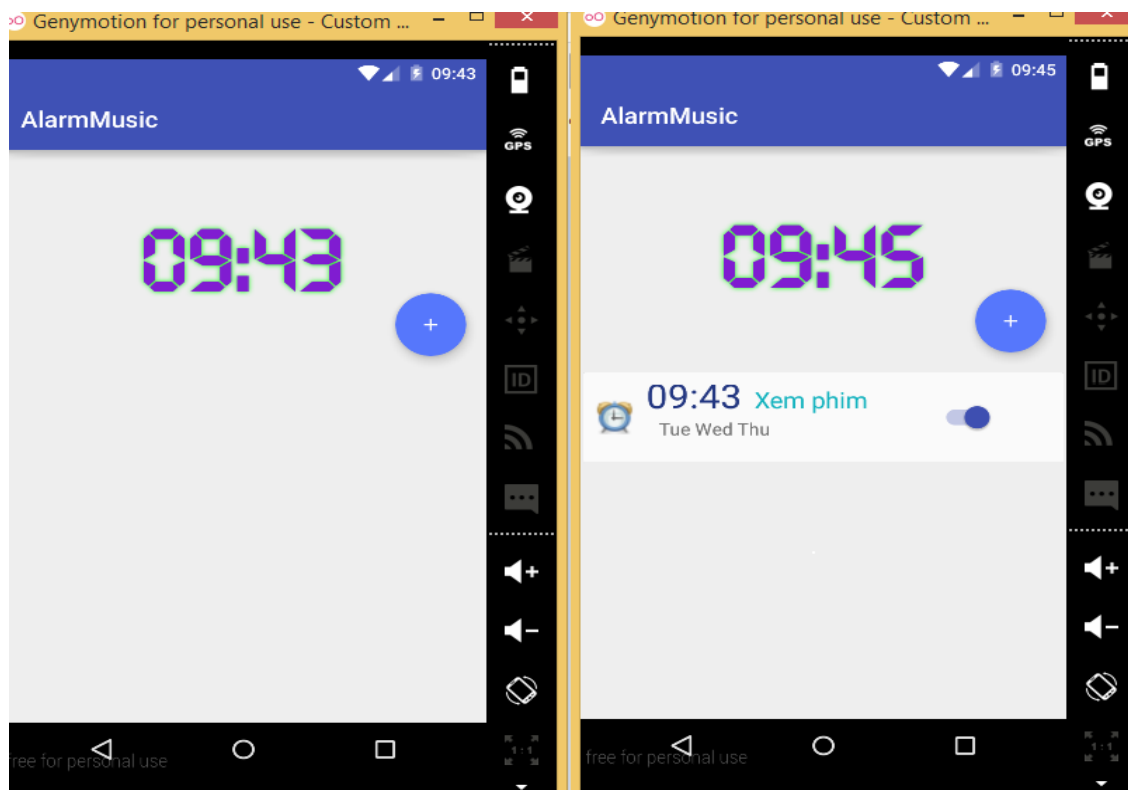
4.2. Cách thức triển khai và sử dụng hệ thống

- Chạy ứng dụng Android bằng máy ảo Genymotion hoặc máy điện thoại thật chạy hệ điều hành Android.

4.3. Kịch bản demo

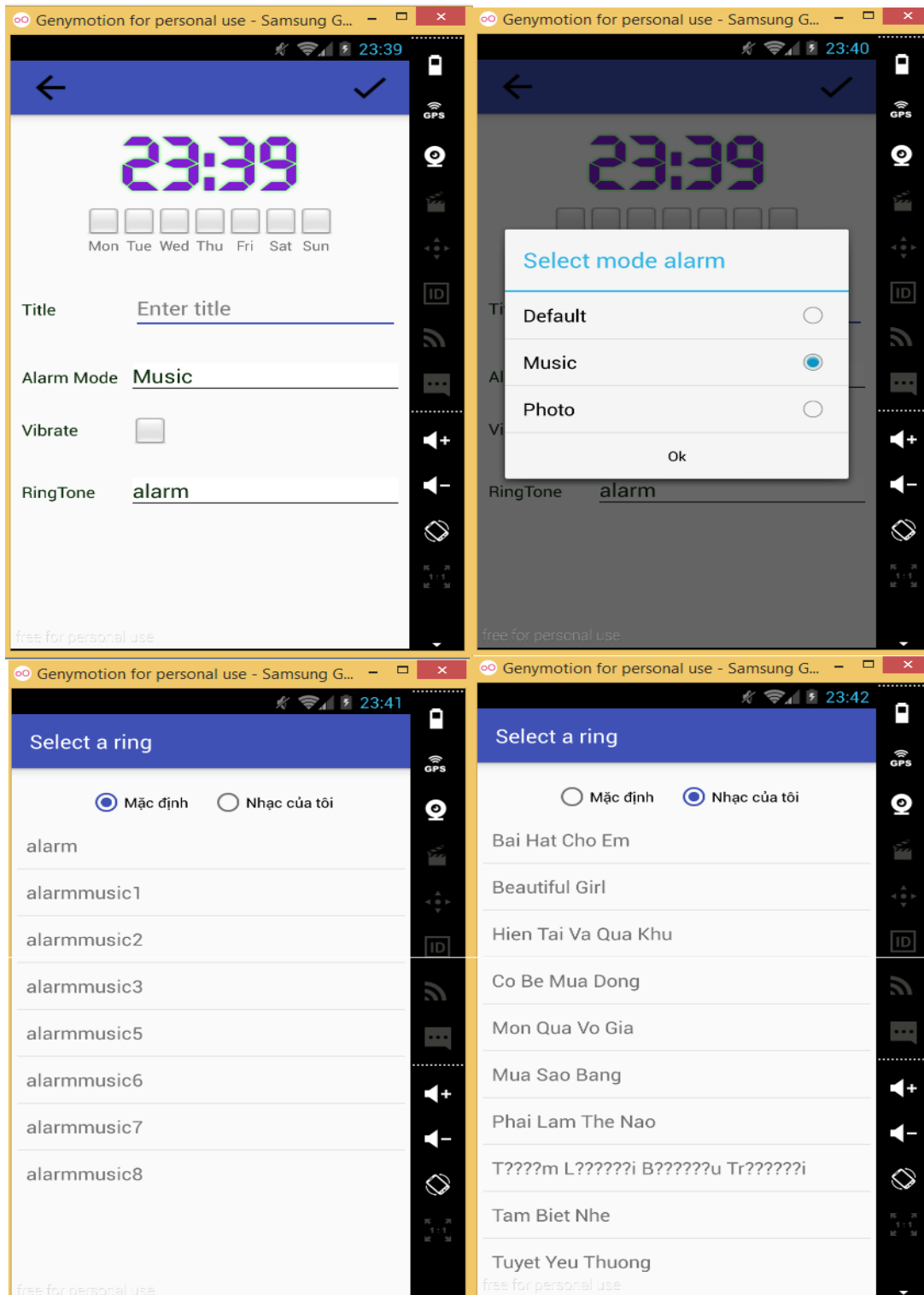
Kịch bản người dùng

Người dùng mở ứng dụng xem danh sách báo thức



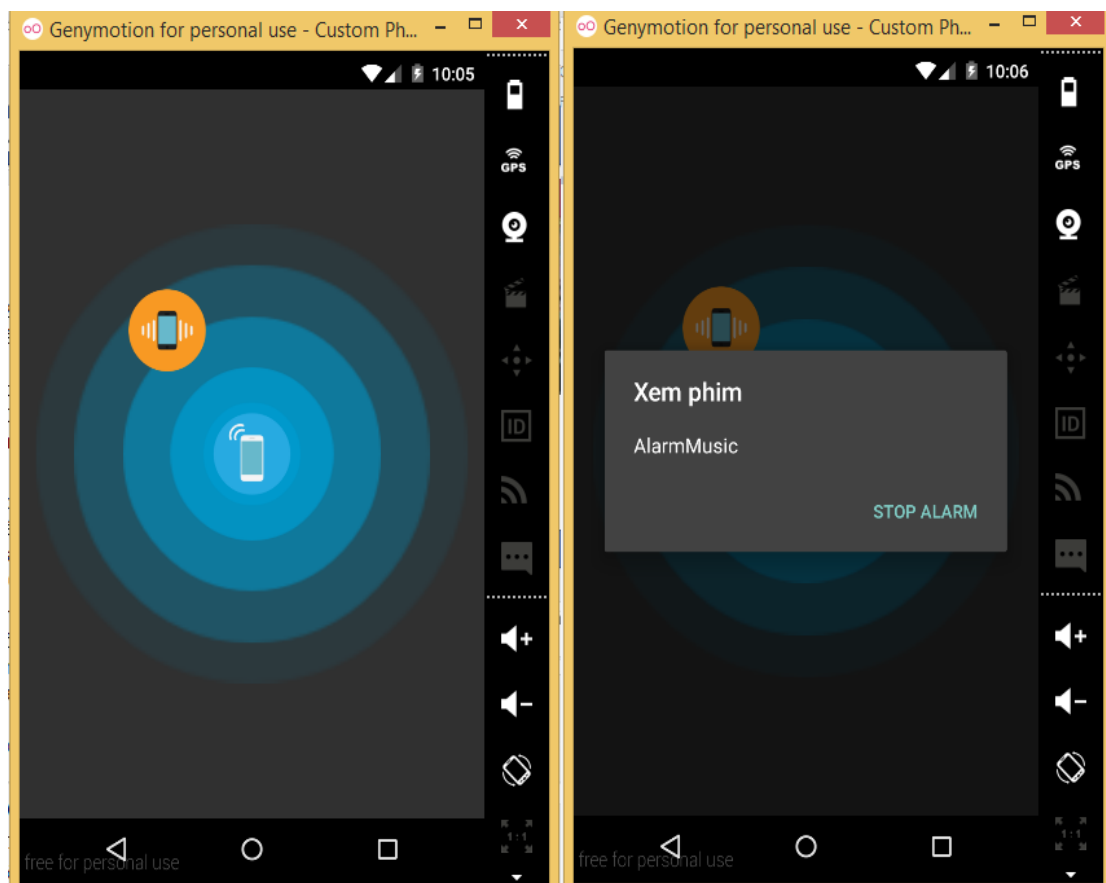
Hình 0.1 Demo chức năng hiển thị danh sách báo thức

Người dùng thêm mới hoặc chỉnh sửa báo thức



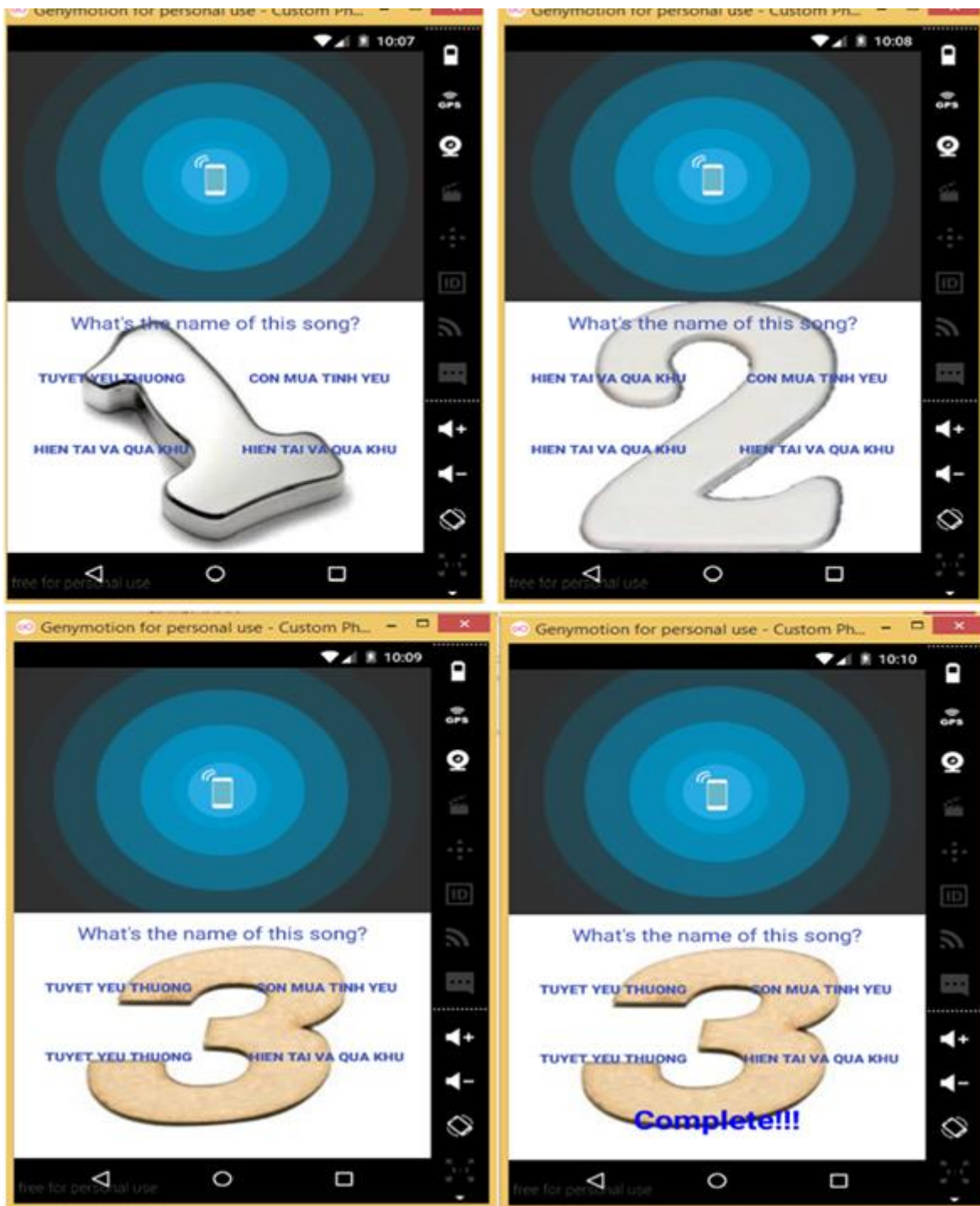
Hình 0.2 Demo chức năng thêm mới hoặc chỉnh sửa báo thức

Phát báo thức:



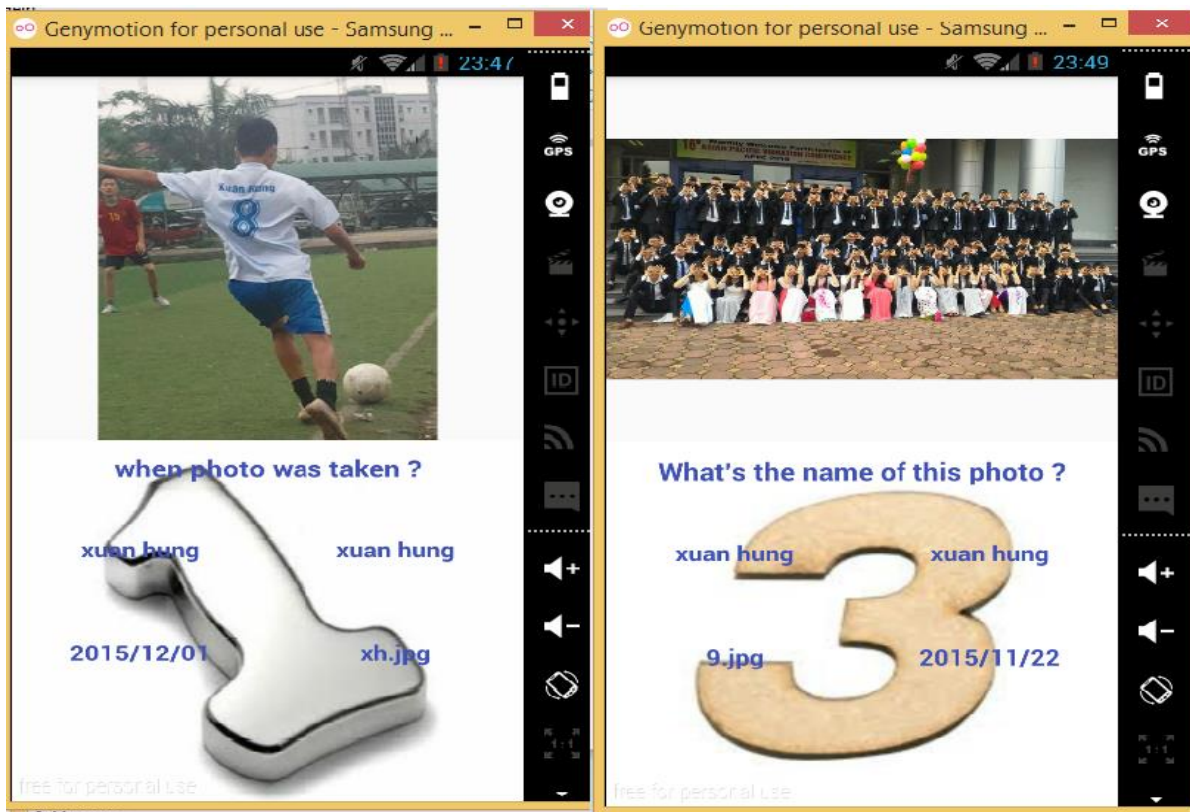
Hình 0.3 Demo chức năng phát âm báo thức

Người dùng nghe nhạc và đoán tên bài hát báo thức



Hình 0.4 Demo chức năng nghe nhạc và tắt báo thức

Xem ảnh và tắt báo thức



Hình 0.5 Demo chức năng xem ảnh và tắt báo thức

4.4 Kết quả thử nghiệm

Đánh giá kết quả thử nghiệm trên các phiên bản Android khác nhau:

Thứ tự	Dòng máy	Phiên bản hệ điều hành	Kết quả
1	LGL34C	Android 4.4	Hệ thống hoạt động ổn định, không crash, giao diện không bị vỡ.
2	Samsung galaxy s4	Android 5.0.1	Hệ thống hoạt động ổn định, không crash, giao diện mượt, đạt yêu cầu.
3	Tablet Asus Nexus 7	Android 6.0	Hệ thống hoạt động ổn định, không crash, giao diện mượt, đạt yêu cầu.

Bảng 0.1 Đánh giá kết quả thử nghiệm trên các phiên bản

KẾT LUẬN

5.1 Các kết quả đạt được

5.1.1.Ưu điểm

Đồ án đã hoàn thành các mục tiêu đề ra. Các vấn đề mà đồ án này đã đạt được như sau:

- Hiểu cách hoạt động của các thành phần cơ bản trong android, đặc biệt là Broadcast Receivers và Service để cài đặt báo thức.
- Xây dựng cơ bản hoàn chỉnh ứng dụng, đáp ứng đầy đủ các chức năng đã đề ra ban đầu.
- Giao diện đơn giản, dễ sử dụng. Phần báo thức khá thú vị.
- Chạy tốt với các bản từ android 4.0 và mới hơn.

5.1.2 Nhược điểm

- Giao diện còn đơn giản, chưa được sắc sảo.
- Chức năng còn đơn giản.

5.2 Hướng phát triển trong tương lai

Hệ thống đã đạt được các yêu cầu đề ra nhưng vẫn còn những hạn chế vì thế hướng phát triển tiếp theo cần cải thiện khắc phục các hạn chế trên:

- Thêm một số cách tắt báo thức độc đáo, mới lạ hơn
- Giao diện sinh động, tinh tế hơn để tạo ấn tượng với người dùng.

TÀI LIỆU THAM KHẢO

1. Ths Trương Thị Ngọc Phượng , *Lập trình Android* , Nhà xuất bản Thời Đại, 2014
2. Dawn Griffiths and David Griffiths, *Head First Android Development*, O'Reilly Media, 2015.
3. Ryan Hodson, *Android Programming Succinctly*, Syncfusion Inc, 2014.
4. Neil Smyth , *Android Studio Development Essentials – Android 6 Edition*, Payload Media , 2015
5. Ths Trần Duy Thành, Advanced Programing, Trường Đại học Khoa Học Tự Nhiên TP.HCM, <https://duythanhcse.wordpress.com>
6. [Google.com](https://www.google.com) , [github.com](https://www.github.com)
7. <https://khoapham.vn>