

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

Xây dựng nhóm tiện ích ứng dụng trên đồng hồ
thông minh

Sinh viên thực hiện : Nguyễn Văn Đông

Lớp KTMT&TT 2

Giáo viên hướng dẫn: [ThS] Bùi Trọng Tùng

HÀ NỘI 05-2015

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Nguyễn Văn Đông

Điện thoại liên lạc 01674599982 Email: dongnv1009@gmail.com

Lớp: KTMT & TT 2 Hệ đào tạo: Kỹ sư

Đồ án tốt nghiệp được thực hiện tại: Bộ phận Software, công ty SmartHome, tập đoàn BKAV, số ... Trung Kính, Cầu Giấy, Hà Nội.

Thời gian làm ĐATN: Từ ngày 15 /1 /2015 đến 29 /05 /2015

2. Mục đích nội dung của ĐATN

Xây dựng nhóm tiện ích ứng dụng chạy trên đồng hồ thông minh để phục vụ nhu cầu của người dùng.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu công nghệ giao tiếp giữa đồng hồ thông minh và smartphone.
- Cách thức đồng bộ dữ liệu, thông báo, cảnh báo.
- Các chức năng cơ bản của một đồng hồ thông thường.
- Các chức năng thông minh tương tác với smartphone.
- Đo và tính toán các thông tin sức khỏe người dùng.
- Đưa ra cảnh báo và lời khuyên hữu ích.

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Văn Đông* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Thạc sĩ. Bùi Trọng Tùng*

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 29 tháng 5 năm 2015

Tác giả ĐATN

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

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Các nhiệm vụ chính đề ra trong đồ án tốt nghiệp:

- Tìm hiểu việc giao tiếp giữa đồng hồ thông minh và smartphone.
- Phân tích và thiết kế hệ thống.
- Lập trình nhóm tiện ích ứng dụng chạy trên đồng hồ thông minh.

Môi trường thực hiện đồ án tốt nghiệp:

Công ty cổ phần SmartHome, Bộ phận Phần mềm, Trung Kính, Cầu Giấy, Hà Nội.

Bố cục đồ án: Bao gồm phần Mở đầu, Nội dung, Kết luận, Tài liệu tham khảo.

Phần mở đầu: Giới thiệu tóm tắt nhiệm vụ, đề tài, mục tiêu và phạm vi thực hiện đồ án.

Phần nội dung: gồm 2 phần

- *Đặt vấn đề và định hướng giải quyết:* Phần này mô tả bài toán xây dựng nhóm tiện ích ứng dụng chạy trên đồng hồ thông minh, các vấn đề cần đáp ứng; đưa ra các hướng giải quyết và việc lựa chọn các công nghệ.
- *Các kết quả đạt được:* Phần này trình bày kết quả phân tích, thiết kế hệ, cài đặt thử nghiệm ứng dụng và đưa ra những đánh giá chung về ưu nhược điểm, khả năng ứng dụng trong thực tế.

Phần kết luận: Kết luận chung của ĐATN, đánh giá các công việc đã làm và chưa làm được trong khuôn khổ thực hiện đồ án tốt nghiệp.

LỜI CẢM ƠN

Em xin được gửi lời cảm ơn chân thành tới các thầy cô giáo trong trường đại học Bách khoa Hà Nội nói chung và trong Viện Công Nghệ Thông Tin và Truyền Thông, bộ môn Kỹ Thuật Máy Tính và Truyền Thông nói riêng đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt thời gian 5 năm học vừa qua.

Đặc biệt em xin gửi lời cảm ơn chân thành đến Ths. Bùi Trọng Tùng đã trực tiếp tận tình giúp đỡ, chỉ bảo, hướng dẫn em trong công việc nghiên cứu và trong suốt quá trình thực hiện đồ án tốt nghiệp. Những kiến thức đó đã giúp em rất nhiều trong quá trình học tập và nghiên cứu, và sẽ còn tiếp tục bên em trong công việc sau này. Em cũng gửi lời cảm ơn tới Công ty trách nhiệm hữu hạn đầu tư và phát triển công nghệ Ngôi nhà thông minh (Bkav SmartHome Technology Company Limited) đã tạo điều kiện cũng như môi trường để em có thể học tập những kiến thức mới cũng như giúp đỡ em trong quá trình thực tập và làm đồ án này.

Cuối cùng em xin gửi lời cảm ơn chân thành tới gia đình, bạn bè đã quan tâm, động viên, đóng góp ý kiến và giúp đỡ trong quá trình học tập, nghiên cứu để hoàn thành đồ án tốt nghiệp này. Do những hạn chế về kiến thức và thời gian của bản thân, đồ án tốt nghiệp của em chắc chắn còn những hạn chế nhất định, vì vậy rất mong các thầy cô và bạn đọc đóng góp để em có thể hoàn thiện hơn nữa cho sản phẩm của mình.

Hà Nội, ngày 29 tháng 05 năm 2015

Nguyễn Văn Đông

Mục lục

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	2
TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP	4
LỜI CẢM ƠN.....	5
DANH MỤC CÁC HÌNH VẼ	8
DANH MỤC BẢNG BIỂU	10
PHẦN I	11
CHƯƠNG I - ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP	12
I. Đặt vấn đề	12
1. Mô tả bài toán.....	12
2. Các vấn đề đặt ra và định hướng giải quyết.....	12
II. Cơ sở lí thuyết và công cụ được lựa chọn	13
1. Hệ điều hành Android	13
2. Các thành phần trong giao diện Android	17
3. Kết nối bluetooth.....	21
4. Mô hình đồng bộ dữ liệu	21
PHẦN II.....	23
CHƯƠNG II - PHÂN TÍCH VÀ XÂY DỰNG ỨNG DỤNG	24
I. Phân tích ứng dụng	24
1. Tổng quan về đồng hồ thông minh	24
3. Biểu đồ usecase và đặc tả.....	25
4. Biểu đồ hoạt động	38
5. Biểu đồ trình tự	44
II. Thiết kế ứng dụng.....	46
1.1. Thiết kế lớp	46

1.2.	Thiết kế cơ sở dữ liệu	51
1.3.	Thiết kế giao diện	52
III.	Thiết kế kiểm thử.....	53
IV.	Các kịch bản thử nghiệm	55
V.	Cài đặt chương trình và kết quả đạt được.....	57
1.	Giao diện khi sử dụng	57
2.	Đánh giá chung về ứng dụng.....	62
	CHƯƠNG III – KẾT LUẬN	63
1.	Kết luận chung.....	63
2.	Định hướng phát triển của ứng dụng.....	63
	TÀI LIỆU THAM KHẢO	64

DANH MỤC CÁC HÌNH VẼ

Hình 1. Các phiên bản android.....	14
Hình 2. Sơ đồ kiến trúc của hệ điều hành Android.....	14
Hình 3. Mô hình đồng hồ thông minh tổng quan.....	24
Hình 4. Biểu đồ usecase đồng bộ giao diện đồng hồ, thông tin cuộc gọi, tin nhắn, email, sự kiện, notification.....	26
Hình 5. Biểu đồ usecase đồng bộ thông số sức khỏe.....	27
Hình 6. Biểu đồ usecase báo thức.....	28
Hình 7. Biểu đồ usecase trả lời, từ chối cuộc gọi, tắt chuông.....	29
Hình 8. Use case thực hiện cuộc gọi.....	30
Hình 9. Biểu đồ use case gửi tin nhắn nhanh.....	30
Hình 10. Biểu đồ use case play, pause, next, previous nhạc.....	31
Hình 11. Biểu đồ usecase thông báo cuộc gọi đến, tin nhắn mới, notification.....	32
Hình 12. Biểu đồ use case cảnh báo vận động.....	33
Hình 13. Biểu đồ usecase đo nhịp tim.....	34
Hình 14. Biểu đồ use case đo bước chân, tính quãng đường, lượng calo tiêu thụ.....	35
Hình 15. Biểu đồ usecase bật tắt bluetooth.....	36
Hình 16. Biểu đồ hoạt động kết nối.....	39
Hình 17. Biểu đồ hoạt động gọi điện, nhắn tin nhanh.....	40
Hình 18. Biểu đồ hoạt động thông báo, trả lời cuộc gọi.....	41
Hình 19. Biểu đồ hoạt động thông báo notification có trên điện thoại.....	42
Hình 20. Biểu đồ hoạt động nhập thông tin.....	43
Hình 21. Biểu đồ trình tự nhập thông tin.....	44
Hình 22. Biểu đồ trình tự gửi tin nhắn nhanh.....	44
Hình 23. Biểu đồ trình tự gọi điện.....	45
Hình 24. Biểu đồ lớp tổng quan.....	46
Hình 25. Biểu đồ các lớp màn hình cuộc gọi.....	47
Hình 26. Biểu đồ các lớp màn hình đồng hồ.....	48
Hình 27. Biểu đồ các lớp dialog thông báo.....	49
Hình 28. Biểu đồ lớp luồng đo, cảnh báo sức khỏe.....	50
Hình 29. Biểu đồ các lớp setting.....	51
Hình 30. Cấu trúc cây thư mục watch.....	52
Hình 31. Cấu trúc cây thư mục sms.....	52

Hình 32. Màn hình khởi động	57
Hình 33. Màn hình yêu cầu và xác thực kết nối.....	57
Hình 34. Màn hình đồng hồ	58
Hình 35. Màn hình cuộc gọi.....	59
Hình 36. Màn hình tin nhắn, email	59
Hình 37. Màn hình setting.....	60
Hình 38. Màn hình vận động và nhịp tim	60
Hình 39. Màn hình đặt báo thức.....	61
Hình 40. Màn hình đồng hồ bấm giờ, hẹn giờ	61

DANH MỤC BẢNG BIỂU

Bảng 1. Đặc tả use case đồng bộ giao diện đồng hồ, thông tin cuộc gọi, tin nhắn, email, sự kiện, notification.....	27
Bảng 2. Đặc tả use case đồng bộ thông số sức khỏe.....	27
Bảng 3. Đặc tả use case báo thức.....	28
Bảng 4. Đặc tả use case trả lời, từ chối cuộc gọi, tắt chuông	29
Bảng 5. Đặc tả use case thực hiện cuộc gọi	30
Bảng 6. Đặc tả use case gửi tin nhắn nhanh.....	31
Bảng 7. Đặc tả use case: play, pause, next, previous nhạc	32
Bảng 8. Đặc tả use caase thông báo cuộc gọi đến, tin nhắn mới, notification.....	33
Bảng 9. Đặc tả use case cảnh báo vận động.....	34
Bảng 10. Đặc tả use case đo nhịp tim	34
Bảng 11. Đặc tả use case đo số bước chân, tính quãng đường, lượng calo tiêu thụ.....	35
Bảng 12. Đặc tả use case bật tắt bluetooth.....	37
Bảng 13. Đặc tả use case thay đổi ngôn ngữ.....	37
Bảng 14. Đặc tả use case nhập thông tin người dùng và cài đặt thông số cảnh báo sức khỏe	38
Bảng 15. Thiết kế kiểm thử tính đúng đắn.....	54
Bảng 16. Thiết kế kiểm thử tính tin cậy.....	55
Bảng 17. Thiết kế kiểm thử tính dùng được	55

PHẦN I

CHƯƠNG I - ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

I. Đặt vấn đề

1. Mô tả bài toán

Với sự phát triển bùng nổ của công nghệ thông tin, cũng như sự tiến bộ vượt bậc của các công nghệ liên quan trong thời gian gần đây đã đẩy phương thức thông tin liên lạc của con người lên một tầm cao mới. Cuộc sống con người ngày càng được cải thiện thì những nhu cầu phục vụ cho cuộc sống con người ngày càng được nâng cao.

Năm 2013 - 2014 là năm mà người yêu thích công nghệ được chào đón bởi hàng loạt các sản phẩm độc đáo, trong số đó không thể không kể đến đồng hồ thông minh - SmartWatch. Đây là một sản phẩm được người dùng rất quan tâm và thu hút sự tham gia của rất nhiều hãng sản xuất đồ công nghệ nổi tiếng như Samsung, LG, Sony, Apple... Với rất nhiều tính năng ưu việt, đồng hồ thông minh đã làm thay đổi hoàn toàn cách nhìn truyền thống của mọi người về đồng hồ đeo tay.

Không chỉ đơn giản là một chiếc đồng hồ, Smart Watch còn được trang bị công nghệ kết nối không dây, cảm biến chuyển động đếm số bước đi, đo nhịp tim...nhờ đó nó có thể hỗ trợ rất nhiều cho người dùng trong việc chăm sóc sức khỏe. Với công nghệ kết nối không dây bluetooth kết hợp với smartphone, đồng hồ thông minh đã đảm nhận tốt vai trò thông báo nhanh nhất đến người dùng các thông tin cuộc gọi, tin nhắn hay nhắc nhở các ghi nhớ mà người dùng đặt ra.

2. Các vấn đề đặt ra và định hướng giải quyết

Các vấn đề đặt ra:

- + Đồng hồ phải kết nối được với điện thoại.
- + Tổ chức một cấu trúc dữ liệu chung để có thể đồng bộ giữa điện thoại và đồng hồ.
- + Đo và tính toán được các thông số sức khỏe người dùng.

Định hướng giải quyết:

- + Sử dụng kết nối bluetooth để làm giao thức kết nối cho điện thoại và đồng hồ.
- + Quy định cấu trúc tổ chức các dữ liệu về danh bạ, cuộc gọi, tin nhắn, email, notification, thông số sức khỏe một cách thống nhất giữa điện thoại và đồng hồ để có thể đồng bộ dữ liệu một cách dễ dàng.

+ Sử dụng các thuật toán để đo và tính toán các thông số, cảnh báo sức khỏe cho người dùng.

II. Cơ sở lý thuyết và công cụ được lựa chọn

1. Hệ điều hành Android

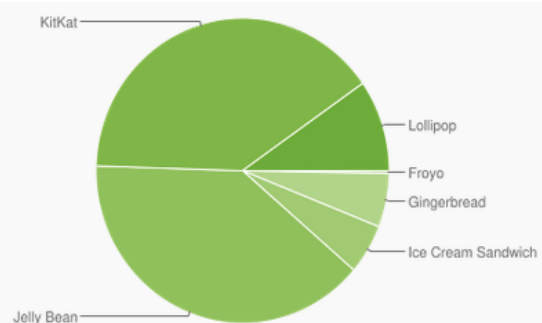
- Giới thiệu chung

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cảm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

Android là hệ điều hành trên điện thoại phổ biến nhất trên thế giới. Hiện tại có hàng trăm triệu thiết bị chạy hệ điều hành này và phân bố trên hơn 190 quốc gia. Android có sự tăng trưởng rất nhanh về số thiết bị cài đặt. Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi.

Hệ điều hành Android phát triển ngày càng lớn mạnh và kể từ khi nó ra đời tới giờ nó đã ra khá nhiều các phiên bản khác nhau.

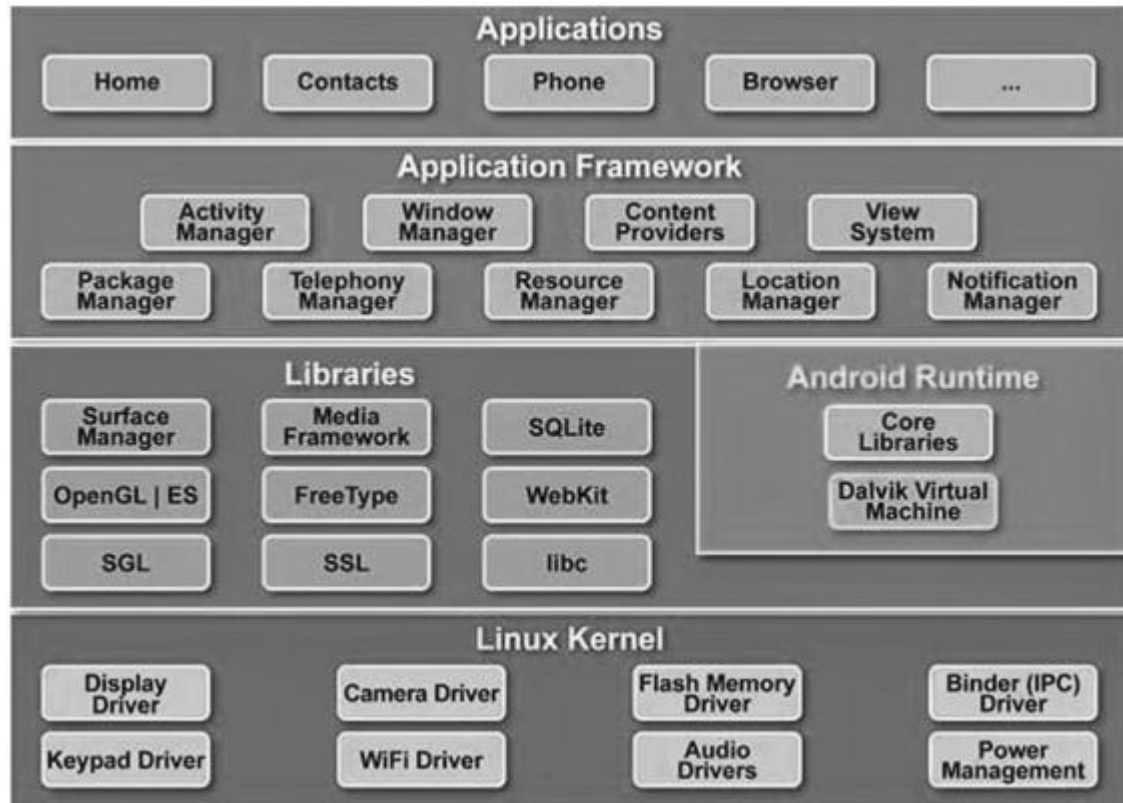
Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.3%
4.1.x	Jelly Bean	16	15.6%
4.2.x		17	18.1%
4.3		18	5.5%
4.4	KitKat	19	39.8%
5.0	Lollipop	21	9.0%
5.1		22	0.7%



Hình 1. Các phiên bản android

- Kiến trúc hệ điều hành Android

Hệ điều hành Android có các thành phần phần mềm tạm chia thành 5 phần với 4 tầng chính trong sơ đồ kiến trúc như hình dưới đây:



Hình 2. Sơ đồ kiến trúc của hệ điều hành Android

- Linux Kernel

Ở dưới cùng của các tầng trên là Linux - Linux 2.6 với khoảng 115 bản vá lỗi. Tầng này cung cấp chức năng hệ thống cơ bản như quản lý các tiến trình, quản lý bộ nhớ, quản lý thiết bị như máy ảnh, bàn phím, màn hình hiển thị, v.v... Ngoài ra, nhân Linux xử lý tất cả các vấn đề về Mạng và một loạt các trình điều khiển thiết bị giao tiếp với phần cứng ngoại vi.

- Libraries

Tầng này là một tập hợp các thư viện bao gồm trình duyệt web mã nguồn mở sử dụng WebKit engine, thư viện libc, cơ sở dữ liệu SQLite là một kho lưu

trữ hữu ích cho việc lưu trữ và chia sẻ dữ liệu của ứng dụng, có các thư viện để chơi/ghi lại âm thanh và video, các thư viện SSL chịu trách nhiệm về bảo mật Internet, v.v...

- Android Runtime

Đây là phần thứ ba của kiến trúc Android. Phần này cung cấp một thành phần quan trọng được gọi là Dalvik Virtual Machine (DVM) - một loại máy ảo Java được thiết kế đặc biệt và tối ưu hóa cho Android. DVM sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ và đa luồng, những tính năng này đều có trong ngôn ngữ Java. Dalvik VM cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó. Android Runtime cũng cung cấp một tập các thư viện lõi cho phép các nhà phát triển ứng dụng Android viết các ứng dụng Android sử dụng ngôn ngữ lập trình Java (J2SE).

- Application Framework

Tầng Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong ở dạng các class trong Java. Các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong các ứng dụng của họ. (Ví dụ: dịch vụ kết nối Internet, dịch vụ SMS,...).

- Applications

Bạn sẽ viết ứng dụng và các ứng dụng mà bạn viết được cài đặt ở tầng này. Ví dụ: Angry Bird, Facebook, Viber, v.v...

Các thành phần:

Việc hiểu được các thành phần (component) tạo nên một ứng dụng Android là rất cần thiết cho việc lập trình. Các thành phần này được chia làm 6 loại bao gồm:

- Activity: hiểu một cách đơn giản thì Activity là nền của 1 ứng dụng. Khi khởi động 1 ứng dụng Android nào đó thì bao giờ cũng có 1 main Activity được gọi, hiển thị màn hình giao diện của ứng dụng cho phép người dùng tương tác.

- Service: thành phần chạy ẩn trong Android. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.
- Content Provider: kho dữ liệu chia sẻ. Content Provider được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng.
- Intent: nền tảng để truyền tải các thông báo. Intent được sử dụng để gửi các thông báo đi nhằm khởi tạo 1 Activity hay Service để thực hiện công việc bạn mong muốn. Ví dụ: khi mở 1 trang web, bạn gửi 1 intent đi để tạo 1 activity mới hiển thị trang web đó.
- Broadcast Receiver: thành phần thu nhận các Intent bên ngoài gửi tới. VD: bạn viết 1 chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn cần 1 BR để nhận biết các Intent là các cuộc gọi tới.
- Notification: đưa ra các cảnh báo mà không làm cho các Activity phải ngừng hoạt động. Activity, Service, Broadcast Receiver và Content Provider là những thành phần chính cấu thành nên ứng dụng Android, bắt buộc phải khai báo trong AndroidManifest. Android có cơ chế quản lý các process theo chế độ ưu tiên. Các process có priority thấp sẽ bị Android giải phóng mà không hề cảnh báo nhằm đảm bảo tài nguyên.
- Foreground process: là process của ứng dụng hiện thời đang được người dùng tương tác.
- Visible process: là process của ứng dụng mà activity đang hiển thị đối với người dùng (onPaused() của activity được gọi).
- Service process: là Service đang running.
- Background process: là process của ứng dụng mà các activity của nó không hiển thị với người dùng (onStoped() của activity được gọi).
- Empty process: process không có bất cứ 1 thành phần nào active. Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động kill process, trước tiên là các empty process.

- Android Activity Life Cycle:

Trong các thành phần của một ứng dụng Android Activity là thành phần quan trọng nhất và đóng vai trò chính trong xây dựng ứng dụng Android. Hệ điều

hành Android quản lý Activity theo dạng stack: khi một Activity mới được khởi tạo, nó sẽ được xếp lên đầu của stack và trở thành running activity, các Activity trước đó sẽ bị tạm dừng và chỉ hoạt động trở lại khi Activity bao gồm 4 state:

- Active (running): Activity đang hiển thị trên màn hình (foreground).
- Paused: Activity vẫn hiển thị (visible) nhưng không thể tương tác (lost focus). Ví dụ: một activity mới xuất hiện hiển thị giao diện đè lên trên activity cũ, nhưng giao diện này nhỏ hơn giao diện của activity cũ, do đó ta vẫn thấy được 1 phần giao diện của activity cũ nhưng lại không thể tương tác với nó.
- Stop: Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái stop.
- Killed: Khi hệ thống bị thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên. Các Activity ở trạng thái stop hoặc paused cũng có thể bị giải phóng và khi nó được hiển thị lại thì các Activity này phải khởi động lại hoàn toàn và phục hồi lại trạng thái trước đó.

Không giống như lập trình java thông thường, lập trình android ngoài các lớp được viết trong *.java còn sử dụng XML để thiết kế giao diện cho ứng dụng. Người lập trình hoàn toàn có thể thiết kế 1 giao diện như ý muốn mà không cần tới bất cứ một dòng XML nào, nhưng sử dụng XML sẽ đơn giản công việc đi rất nhiều. Đồng thời sử dụng XML sẽ giúp việc chỉnh sửa ứng dụng sau này trở nên dễ dàng. Về nguyên tắc, khi lập trình ứng dụng ta thiết kế giao diện bằng XML và cài đặt các xử lý khi tương tác với giao diện trong code.

2. Các thành phần trong giao diện Android

- 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.

- 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.

+ 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.

+ AbsoluteLayout: Layout này được sử dụng để bố trí các widget vào một vị trí bất kì trong layout dựa vào 2 thuộc tính tọa độ x, y. Tuy nhiên, kiểu layout này rất ít khi được dùng bởi vì tọa 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.

+ RelativeLayout: 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 RelativeLayout 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.

+ 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/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!");  
cmdButton.setOnClickListener(...);
```

+ Image Button: Cũng tương tự như Button, ImageButton chỉ có thêm một thuộc tính `android:src = "@drawable/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/cmdButton1"  
android:src="@drawable/icon"  
android:onClick="touchMe"/>
```

+ Image view: Được dùng để thể hiện một hình ảnh. Nó cũng giống như ImageButton, chỉ khác là không có hình dáng của một cái button.

Code:

```
ImageView iv = new ImageView(this);  
iv.setImageResource(R.drawable.icon);
```

+ Listview: 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.

3. Kết nối bluetooth

- Bluetooth là một đặc tả công nghiệp cho truyền thông không dây tầm gần giữa các thiết bị điện tử. Công nghệ này hỗ trợ việc truyền dữ liệu qua các khoảng cách ngắn giữa các thiết bị di động và cố định, tạo nên các mạng cá nhân không dây (Wireless Personal Area Network-PANs).
- Bluetooth có thể đạt được tốc độ truyền dữ liệu 1Mb/s. Bluetooth hỗ trợ tốc độ truyền tải dữ liệu lên tới 720 Kbps trong phạm vi 10 m–100 m. Khác với kết nối hồng ngoại (IrDA), kết nối Bluetooth là vô hướng và sử dụng giải tần 2,4 GHz.
- Bluetooth cho phép kết nối và trao đổi thông tin giữa các thiết bị như điện thoại di động, điện thoại cố định, máy tính xách tay, PC, máy in, thiết bị định vị dùng GPS, máy ảnh số, và video game console.

Các ứng dụng nổi bật của Bluetooth gồm:

- Điều khiển và giao tiếp không dây giữa một điện thoại di động và tai nghe không dây.
- Mạng không dây giữa các máy tính cá nhân trong một không gian hẹp đòi hỏi ít băng thông.
- Giao tiếp không dây với các thiết bị vào ra của máy tính, chẳng hạn như chuột, bàn phím và máy in.
- Truyền dữ liệu giữa các thiết bị dùng giao thức OBEX.
- Thay thế các giao tiếp nối tiếp dùng dây truyền thống giữa các thiết bị đo, thiết bị định vị dùng GPS, thiết bị y tế, máy quét mã vạch, và các thiết bị điều khiển giao thông.
- Thay thế các điều khiển dùng tia hồng ngoại.
- Gửi các mẫu quảng cáo nhỏ từ các pa-nô quảng cáo tới các thiết bị dùng Bluetooth khác.
- Điều khiển từ xa cho các thiết bị trò chơi điện tử như Wii - Máy chơi trò chơi điện tử thế hệ 7 của Nintendo[1] và PlayStation 3 của Sony.

4. Mô hình đồng bộ dữ liệu

- Là thư viện của công ty TNHH công nghệ Ngôi nhà thông minh BKAV SmartHome được viết bởi Phạm Quang Hòa Giám đốc công nghệ của BKAV SmartHome.
- Đầu tiên phải nói đến việc tổ chức dữ liệu đó là các dữ liệu được lưu ra file và tổ chức theo cây thư mục. Lý do dùng lưu ra file đó là đảm bảo dữ liệu không bị mất, có thể khôi phục và chỉnh sửa dữ liệu. - Minh họa cách tổ chức dữ liệu đồng bộ:
 - o Khi lần tiên chạy thì sẽ sinh ra các đối tượng chính như sms, email hay còn gọi là thư mục cha các sms.0, sms.1, email.1 là các

đối tượng thông tin đã lấy được và lưu lại. Cơ chế đồng bộ chính là dựa vào sự thay đổi của các đối tượng chính.

- Sau khi tổ chức dữ liệu thành các đối tượng đồng bộ thì sẽ có một luồng kiểm tra liên tục các đối tượng, để tìm ra các đối tượng đã bị thay đổi thì sẽ đóng gói thành gói tin theo dạng chuỗi sang cho thiết bị đang kết nối. Và ở phía bên kia nhận gói tin lưu lại thành đối tượng chuẩn giống với bên gửi sang.
 - Ví dụ về gói tin đồng bộ: ví dụ có thêm một tin nhắn nữa và sinh ra đối tượng sms.2 gói tin gửi đi sẽ có dạng {sms:{sms2:{contactName:195,number:195,content:Viettel chương trình khuyến mãi tặng 50% cho mỗi thẻ nạp,time:06/01/2015 1\;36\;36}} } –
- Vấn đề đặt ra là liệu dữ liệu có bị mất khi đang đồng bộ và mất kết nối không? Câu trả lời là không vì khi đang đồng bộ dữ liệu được gửi sang từng byte một, khi bị mất kết nối dữ liệu nhận được ở bên kia sẽ không phải là một gói tin đầy đủ do vậy dữ liệu sẽ không được đồng bộ sang khi đang đồng bộ mà bị mất kết nối.

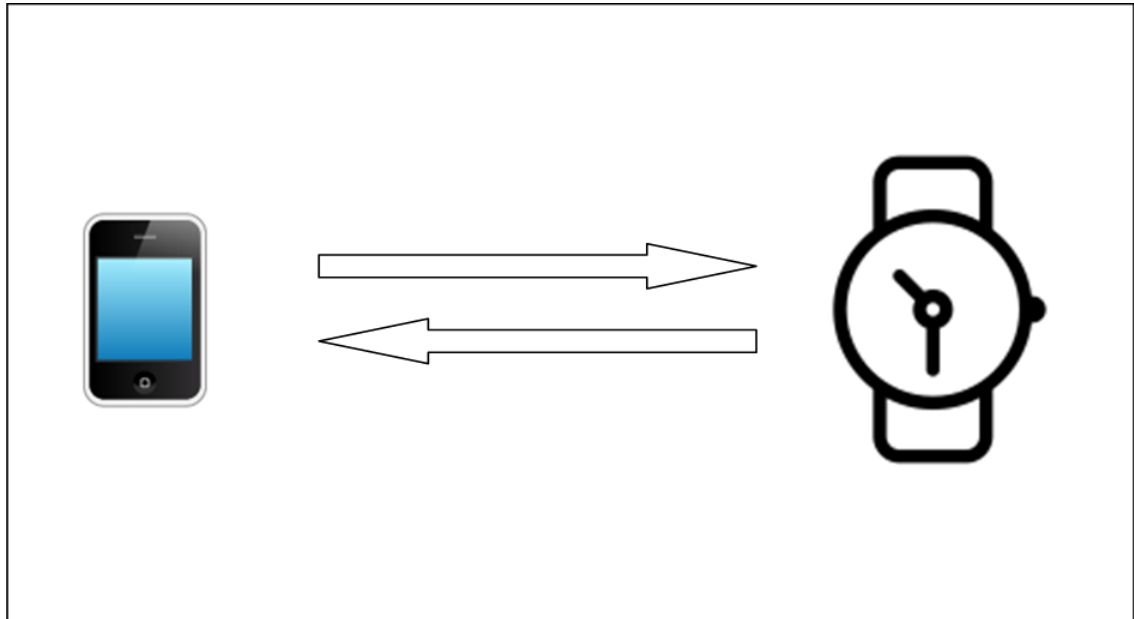
PHẦN II

CHƯƠNG II - PHÂN TÍCH VÀ XÂY DỰNG ỨNG DỤNG

I. Phân tích ứng dụng

1. Tổng quan về đồng hồ thông minh

Đồng hồ thông minh là một thiết bị độc lập sử dụng nền tảng hệ điều hành android. Trên đồng hồ có cài đặt phần mềm được build trực tiếp vào hệ điều hành giúp nó có thể giao tiếp với các thiết bị smartphone chạy android khác để thực hiện các chức năng phục vụ cho người sử dụng.



Hình 3. Mô hình đồng hồ thông minh tổng quan

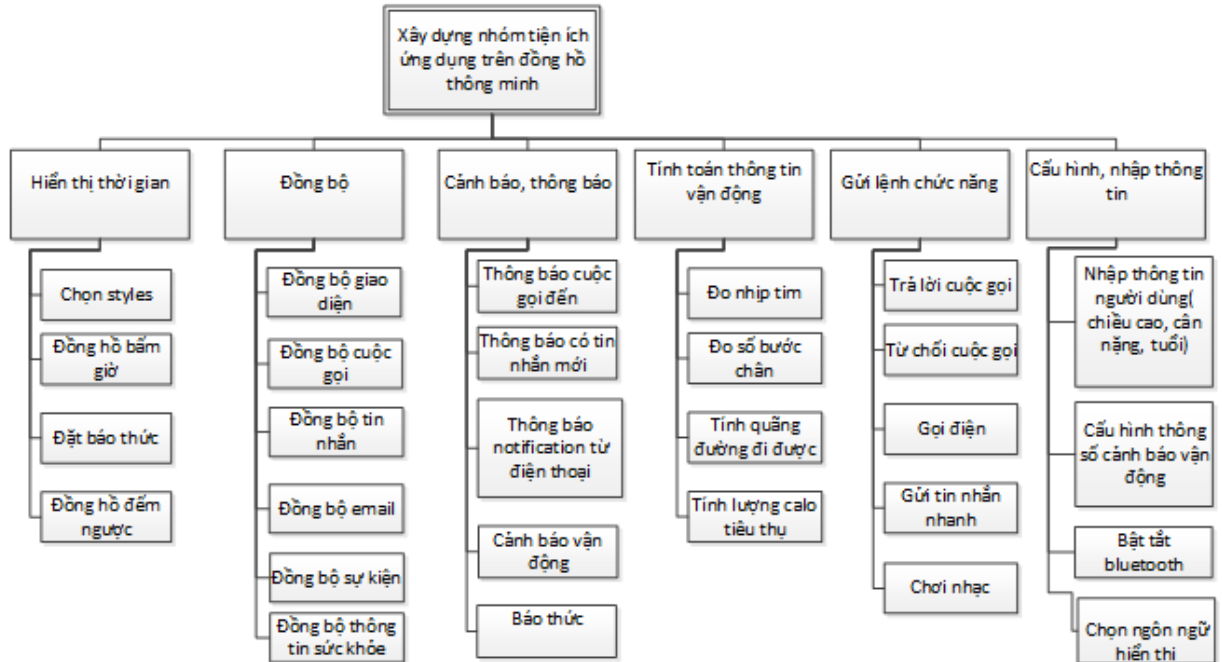
Đồng hồ và điện thoại sử dụng kết nối bluetooth để giao tiếp với nhau, trong đó chủ yếu nhất là vấn đề đồng bộ dữ liệu.

Ứng dụng có các nhóm chức năng chính đó là :

- + Đặt thời gian: hiển thị thời gian ở các định dạng khác nhau, các chức năng hẹn giờ, bấm giờ, đặt báo thức.
- + Cài đặt giao diện: Cho phép người dùng chọn các giao diện đồng hồ khác nhau.
- + Đồng bộ: nhận các dữ liệu về tin nhắn, cuộc gọi, event, email, danh bạ, giao diện đồng hồ, chỉ số vận động...
- + Hiển thị cảnh báo, thông báo, rung: khi có cuộc gọi đến, tin nhắn mới, các notify từ điện thoại, báo thức, các cảnh báo về vận động, cảnh báo về sức khỏe.

- + Thống kê sức khỏe: số bước trong ngày, quãng đường đi được, lượng calo tiêu thụ, nhịp tim.
- + Điều khiển điện thoại: gửi thông tin vận động, trả lời, từ chối hoặc đặt chế độ im lặng khi có cuộc gọi đến, gọi lại, nhắn tin nhanh.
- + Chức năng đặt cấu hình thông số cho hệ thống, nhập thông tin người dùng.

2. Biểu đồ phân rã chức năng

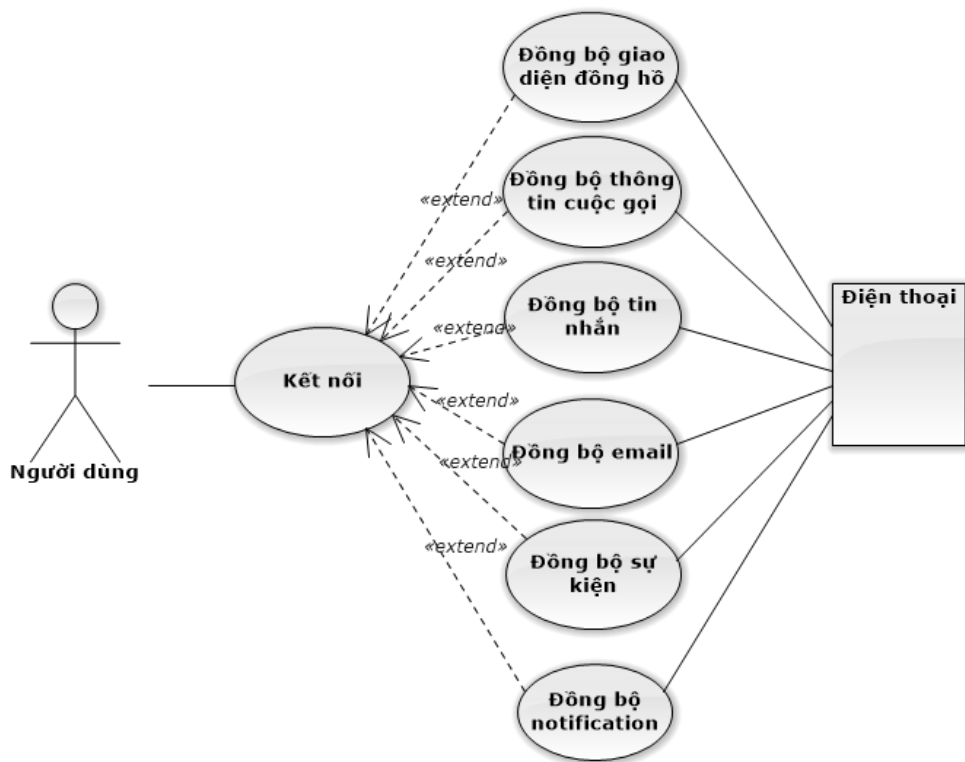


Hình 4. Biểu đồ phân rã chức năng

3. Biểu đồ usecase và đặc tả

1.1. Biểu đồ use case đồng bộ giao diện đồng hồ, thông tin cuộc gọi, tin nhắn, email, sự kiện, notification.

- + Đây là các chức năng đồng bộ dữ liệu mà được lưu trữ trên điện thoại về đồng hồ.
- + Cấu trúc dữ liệu các thông tin giao diện đồng hồ, thông tin cuộc gọi, tin nhắn, email, sự kiện, notification đã được định nghĩa thống nhất giữa 2 bên điện thoại và đồng hồ. Khi đồng hồ đồng bộ từ điện thoại về, nó sẽ lưu vào bộ nhớ trong để sử dụng.



Hình 5. Biểu đồ usecase đồng bộ giao diện đồng hồ, thông tin cuộc gọi, tin nhắn, email, sự kiện, notification.

Đặc tả use case:

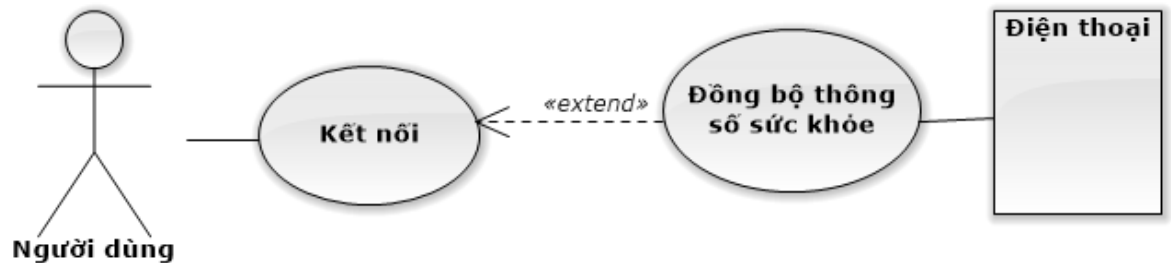
- Use case đồng bộ giao diện đồng hồ, thông tin cuộc gọi, tin nhắn, email, sự kiện, notification

Tên	Đồng bộ giao diện, cuộc gọi, tin nhắn, email, sự kiện
Mô tả chung	Cho phép đồng hồ lấy danh sách các styles được người dùng lựa chọn trên điện thoại, thông tin lịch sử cuộc gọi, tin nhắn, email, các sự kiện trong calendar.
Tác nhân	Điện thoại
Xảy ra khi	Khi các styles được chọn trên app điện thoại thay đổi hoặc khi thông tin về lịch sử cuộc gọi, tin nhắn, email, các sự kiện trong calendar có sự thay đổi, điện thoại sẽ gửi về thông báo cho đồng hồ.
Điều kiện	Khi kết nối đã được thiết lập.
Dòng sự kiện	1. Danh sách styles đồng hồ, thông tin về lịch sử cuộc gọi, tin nhắn, email, các sự kiện trong

	calendar có sự thay đổi. 2. Điện thoại gửi thông báo về cho đồng hồ. 3. Đồng hồ sẽ đồng bộ dữ liệu và cập nhật lên giao diện.
Ngoại lệ	Không có.

Bảng 1. Đặc tả use case đồng bộ giao diện đồng hồ, thông tin cuộc gọi, tin nhắn, email, sự kiện, notification

1.2. Biểu đồ use case đồng bộ thông số sức khỏe



Hình 6. Biểu đồ usecase đồng bộ thông số sức khỏe

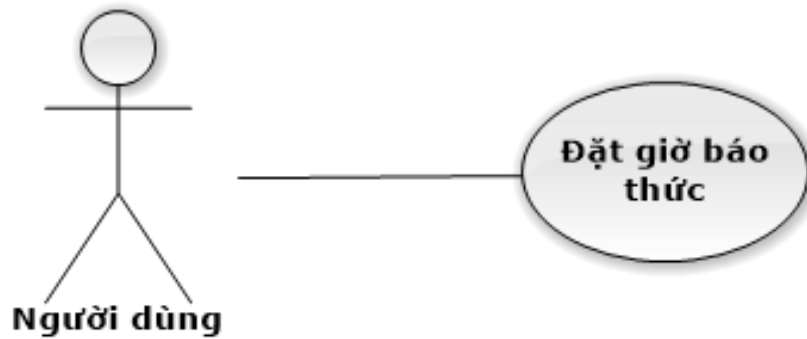
Đặc tả use case:

➤ Use case: Đồng bộ thông số sức khỏe

Tên	Đồng bộ thông số sức khỏe
Mô tả chung	Trường hợp này cho phép đồng hồ đồng bộ các thông số sức khỏe tính toán được về cho điện thoại: số bước chân, quãng đường, lượng calo tiêu thụ, nhịp tim đo được.
Tác nhân	Người dùng
Xảy ra khi	Khi có sự thay đổi về thông số vận động, sức khỏe trên đồng hồ.
Điều kiện	Chương trình phải đang kết nối với đồng hồ.
Dòng sự kiện	1. Người dùng đo nhịp tim, số bước chân, quãng đường, calo tiêu thụ 2. Thông tin được gửi về cho điện thoại.
Ngoại lệ	Không có.

Bảng 2. Đặc tả use case đồng bộ thông số sức khỏe

1.3. Biểu đồ use case báo thức



Hình 7. Biểu đồ usecase báo thức

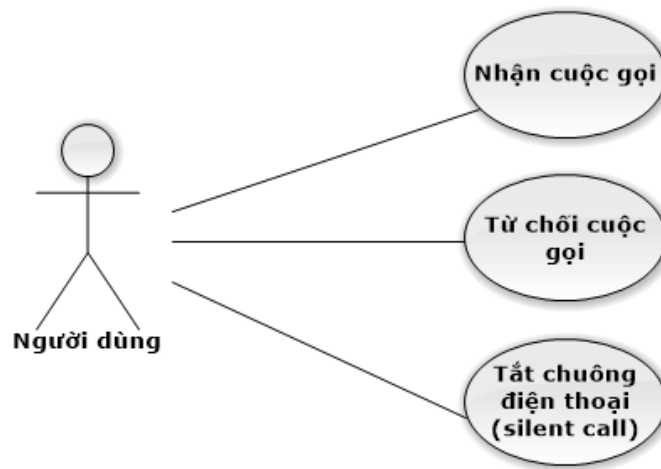
Đặc tả use case

➤ Use case: *Báo thức*

Tên	Báo thức
Mô tả chung	Trường hợp này cho phép đồng hồ rung và hiển thị báo thức.
Tác nhân	Người dùng
Xảy ra khi	Khi đến giờ đặt báo thức.
Điều kiện	Đồng hồ đang hoạt động.
Dòng sự kiện	1. Khi đến giờ báo thức, đồng hồ rung và hiển thị thông báo.
Ngoại lệ	Không có.

Bảng 3. Đặc tả use case báo thức

1.4. Biểu đồ use case trả lời, từ chối cuộc gọi, tắt chuông



Hình 8. Biểu đồ usecase trả lời, từ chối cuộc gọi, tắt chuông

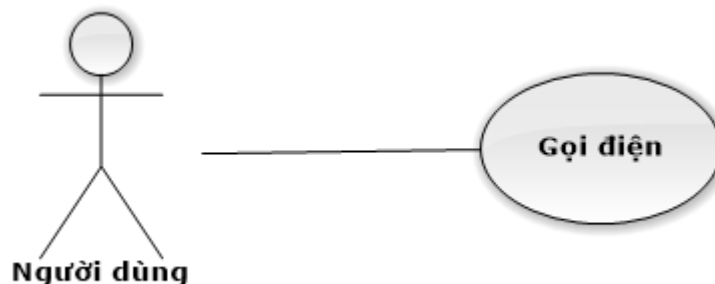
Đặc tả use case:

➤ Use case: *Trả lời, từ chối cuộc gọi, tắt chuông*

Tên	Nghe cuộc gọi
Mô tả chung	Trường hợp này cho phép người dùng có thể nghe cuộc gọi, từ chối cuộc gọi, tắt chuông bằng cách thao tác trên đồng hồ.
Tác nhân	Người dùng
Xảy ra khi	Người dùng muốn nghe cuộc gọi bằng đồng hồ.
Điều kiện	Chương trình phải đang kết nối với đồng hồ.
Dòng sự kiện	1. Người dùng thao tác trên màn hình nghe gọi. 2. Đồng hồ gửi lệnh tương ứng cho điện thoại để thực hiện.
Ngoại lệ	Không có.

Bảng 4. Đặc tả use case trả lời, từ chối cuộc gọi, tắt chuông

1.5. Biểu đồ use case thực hiện cuộc gọi



Hình 9. Use case thực hiện cuộc gọi

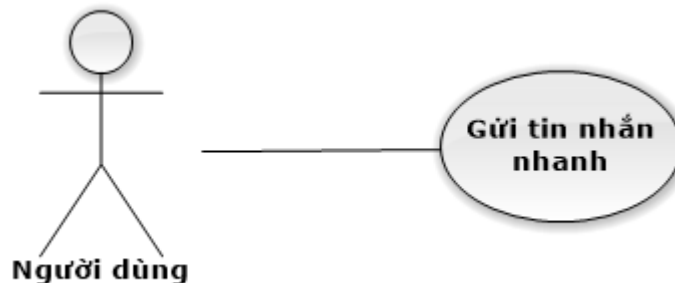
Đặc tả use case:

- Use case: *Thực hiện cuộc gọi*

Tên	Thực hiện cuộc gọi
Mô tả chung	Trường hợp này cho phép người dùng có thể thực hiện cuộc gọi bằng cách thao tác trên đồng hồ.
Tác nhân	Đồng hồ, người dùng
Xảy ra khi	Người dùng muốn thực hiện cuộc gọi bằng đồng hồ.
Điều kiện	Chương trình phải đang kết nối với đồng hồ.
Dòng sự kiện	1. Người dùng chọn vào thực hiện cuộc gọi trên đồng hồ. 2. Đồng hồ gửi lệnh cho hệ thống của điện thoại để thực hiện cuộc gọi.
Ngoại lệ	Không có.

Bảng 5. Đặc tả use case thực hiện cuộc gọi

1.6. Biểu đồ use case gửi tin nhắn nhanh



Hình 10. Biểu đồ use case gửi tin nhắn nhanh

Đặc tả use case:

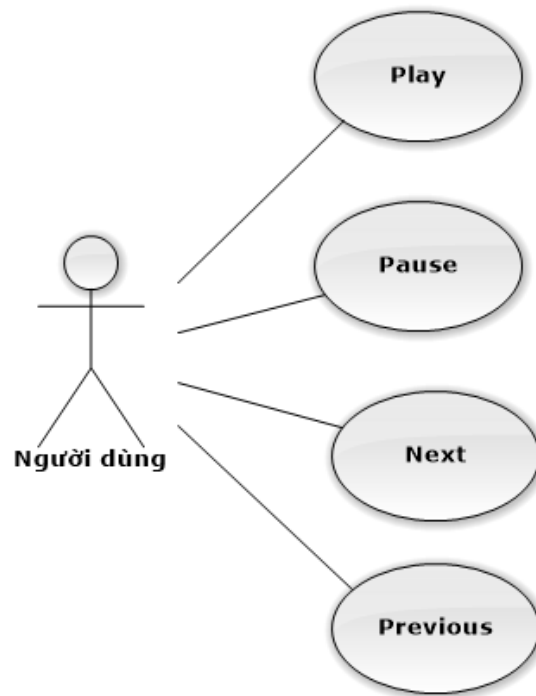
- Use case: *Gửi tin nhắn nhanh*

Tên	Gửi tin nhắn nhanh
Mô tả chung	Trường hợp này cho phép người dùng có thể gửi tin nhắn nhanh bằng cách thao tác trên đồng hồ.
Tác nhân	Đồng hồ, người dùng
Xảy ra khi	Người dùng muốn gửi tin nhắn nhanh bằng đồng hồ.

Điều kiện	Chương trình phải đang kết nối với đồng hồ.
Dòng sự kiện	<ol style="list-style-type: none"> 1. Người dùng chọn vào phím gửi tin nhắn nhanh trên đồng hồ. 2. Người dùng chọn một tin nhắn trong mẫu danh sách tin nhắn nhanh hiển thị lên. 3. Đồng hồ gửi lệnh với tin nhắn nhanh người dùng đã chọn cho điện thoại để thực hiện gửi tin nhắn.
Ngoại lệ	Không có.

Bảng 6. Đặc tả use case gửi tin nhắn nhanh

1.7. Biểu đồ usecase play, pause, next, previous nhạc



Hình 11. Biểu đồ use case play, pause, next, previous nhạc

Đặc tả use case:

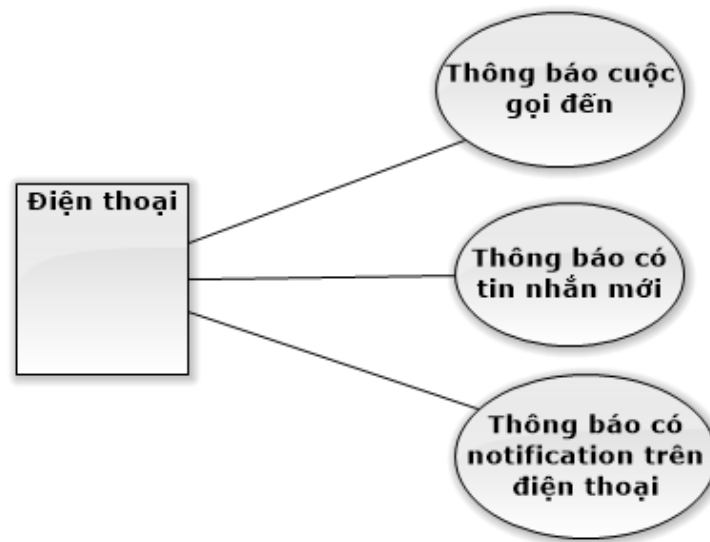
- Use case: play, pause, next, previous nhạc

Tên	Chơi nhạc
Mô tả chung	Trường hợp này cho phép người dùng có thể play, pause, next, previous nhạc trên điện thoại bằng cách thao tác trên đồng hồ.
Tác nhân	Đồng hồ, người dùng
Xảy ra khi	Người dùng muốn điều khiển chơi nhạc trên điện thoại từ đồng hồ.

Điều kiện	Chương trình phải đang kết nối với đồng hồ.
Dòng sự kiện	1. Người dùng chọn chức năng điều khiển nhạc trên màn hình điều khiển trên đồng hồ. 2. Đồng hồ gửi lệnh chức năng yêu cầu đến điện thoại.
Ngoại lệ	Không có.

Bảng 7. Đặc tả use case: play, pause, next, previous nhạc

1.8. Biểu đồ use case thông báo cuộc gọi đến, tin nhắn mới, notification



Hình 12. Biểu đồ usecase thông báo cuộc gọi đến, tin nhắn mới, notification

Đặc tả use case:

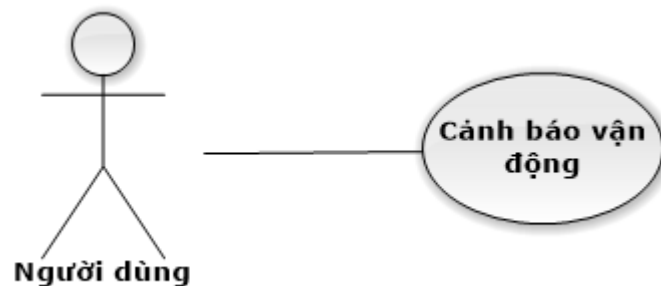
➤ Use case: *Thông báo cuộc gọi đến, tin nhắn mới, notification*

Tên	Thông báo cuộc gọi đến, tin nhắn mới, notification
Mô tả chung	Đồng hồ sẽ rung và hiển thị dialog thông báo có cuộc gọi đến, tin nhắn mới hoặc các notification từ điện thoại.
Tác nhân	Điện thoại
Xảy ra khi	Khi có cuộc gọi đến, tin nhắn mới, notification

Điều kiện	Kết nối đã được thiết lập, điện thoại có sim để nghe gọi.
Dòng sự kiện	<ol style="list-style-type: none"> 1. Khi có cuộc gọi tới, tin nhắn mới, notifications. 2. Điện thoại sẽ gửi thông báo sang đồng hồ. 3. Đồng hồ đồng bộ lấy dữ liệu. 4. Đồng hồ báo rung và hiển thị dialog thông báo.
Ngoại lệ	Không có.

Bảng 8. Đặc tả use caase thông báo cuộc gọi đến, tin nhắn mới, notification

1.9. Biểu đồ use case cảnh báo vận động



Hình 13. Biểu đồ use case cảnh báo vận động

Đặc tả use case:

➤ Use case: *Cảnh báo vận động*

Tên	Cảnh báo vận động
Mô tả chung	Trường hợp này cho phép đồng hồ đưa ra cảnh báo cho người dùng về sức khỏe như quá lâu không vận động, đạt chỉ tiêu đặt vận động đặt ra trong ngày, cảnh báo khi nhịp tim quá cao hoặc quá thấp.
Tác nhân	Người dùng
Xảy ra khi	Khi các thông số sức khỏe đo được nằm ở mức cần cảnh báo cho người dùng.
Điều kiện	Đồng hồ đang hoạt động.
Dòng sự kiện	<ol style="list-style-type: none"> 1. Người dùng thực hiện đo thông số sức khỏe 2. Khi các thông số sức khỏe đo được nằm vào mức cảnh báo.

	3. Đồng hồ sẽ rung và hiển thị dialog cảnh báo.
Ngoại lệ	Không có.

Bảng 9. Đặc tả use case cảnh báo vận động

1.10. Biểu đồ use case đo nhịp tim



Hình 14. Biểu đồ usecase đo nhịp tim

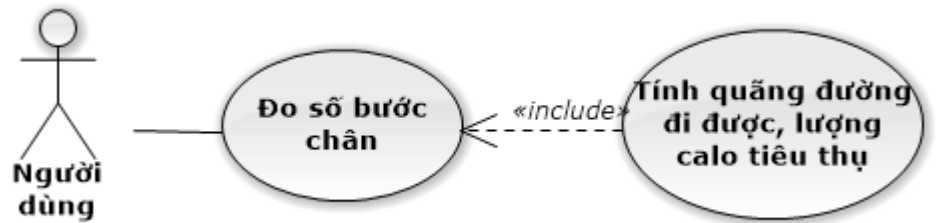
Đặc tả use case:

➤ Use case: *Đo nhịp tim*

Tên	Đo nhịp tim
Mô tả chung	Người dùng có thể yêu cầu đo nhịp tim.
Tác nhân	Đồng hồ, người dùng
Xảy ra khi	Người dùng muốn đo nhịp tim.
Điều kiện	Đồng hồ đang chạy
Dòng sự kiện	1. Người dùng nhấn nút đo nhịp tim trên màn hình HeartRate. 2. Đồng hồ thực hiện đo nhịp tim và hiển thị kết quả trên màn hình.
Ngoại lệ	Không có.

Bảng 10. Đặc tả use case đo nhịp tim

1.11. Biểu đồ use case đo số bước chân, tính quãng đường, lượng calo tiêu thụ



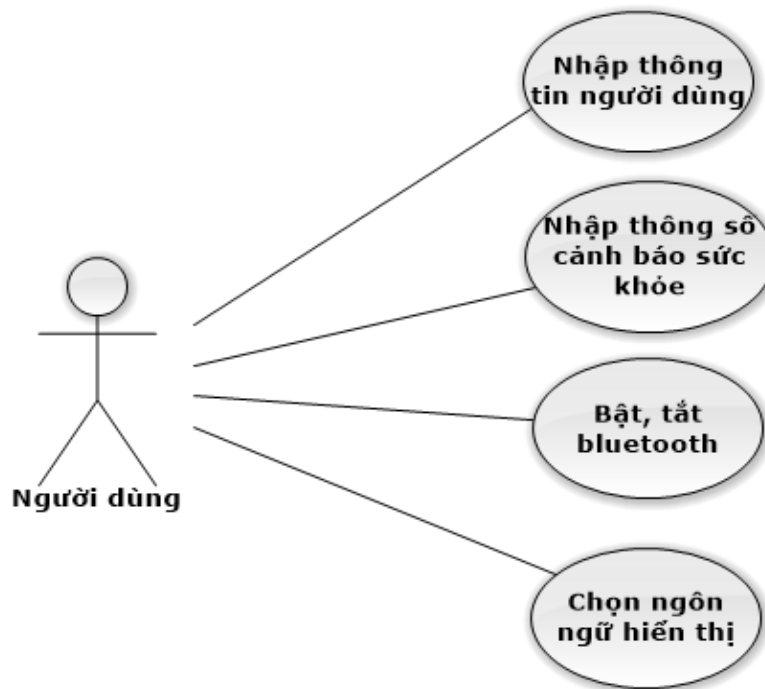
Hình 15. Biểu đồ use case đo bước chân, tính quãng đường, lượng calo tiêu thụ
 Đặc tả use case:

➤ Use case: *Đo số bước chân, tính quãng đường, lượng calo tiêu thụ*

Tên	Đo số bước chân
Mô tả chung	Đồng hồ liên tục đo số bước chân vận động của người dùng, dựa vào số bước chân và các thông số người dùng nhập vào như chiều cao, cân nặng để tính ra quãng đường đi được và lượng calo tiêu thụ.
Tác nhân	Người dùng.
Xảy ra khi	Khi người dùng di chuyển.
Điều kiện	Đồng hồ đang chạy
Dòng sự kiện	1. Đồng hồ đo số bước chân chuyển động của người dùng. 2. Dùng thuật toán để tính quãng đường và lượng calo tiêu thụ. 3. Hiển thị lên màn hình đồng hồ.
Ngoại lệ	Không có.

Bảng 11. Đặc tả use case đo số bước chân, tính quãng đường, lượng calo tiêu thụ

1.12. Biểu đồ use case bật tắt bluetooth



Hình 16. Biểu đồ usecase bật tắt bluetooth

Đặc tả use case

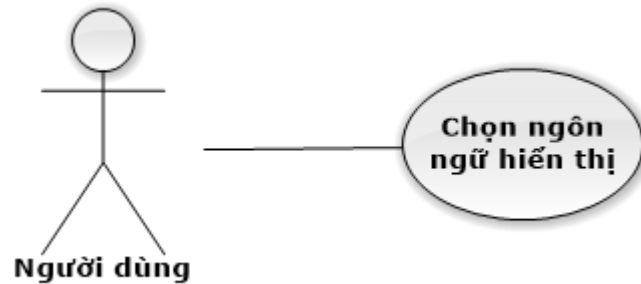
➤ Use case bật tắt bluetooth

Tên	Bật tắt bluetooth
Mô tả chung	Cho phép người dùng bật tắt bluetooth, tùy chọn cho các thiết bị xung quanh có thể thấy hoặc không thấy được.
Tác nhân	Người dùng
Xảy ra khi	Khi người dùng muốn kết nối với điện thoại
Điều kiện	- Người dùng phải khởi động và đang chạy chương trình.
Dòng sự kiện	<ol style="list-style-type: none"> 1. Chương trình yêu cầu người dùng bật bluetooth. 2. Chương trình yêu cầu người dùng cho phép các thiết bị xung quanh có thể nhìn thấy. 3. Khi có một yêu cầu ghép nối đến từ thiết bị khác, chương trình yêu cầu người dùng xác nhận.

Ngoại lệ	Chương trình sẽ báo lỗi liên quan đến kết nối khi thất bại.
-------------	---

Bảng 12. Đặc tả use case bật tắt bluetooth

1.13. Biểu đồ use case thay đổi ngôn ngữ

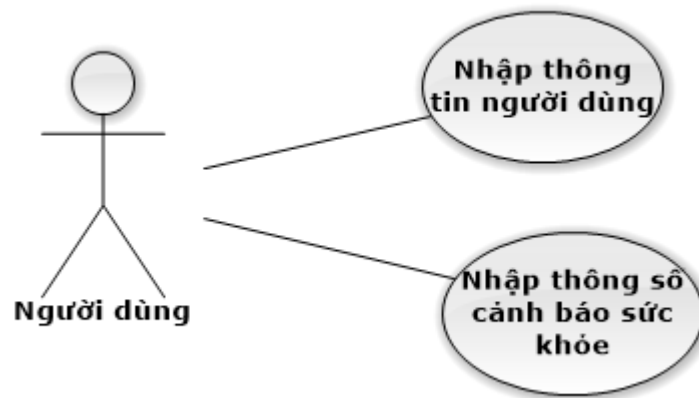


➤ Use case: Thay đổi ngôn ngữ

Tên	Thay đổi ngôn ngữ
Mô tả chung	Trường hợp này cho phép người dùng thay đổi ngôn ngữ hiển thị của chương trình.
Tác nhân	Người dùng
Xảy ra khi	Khi người dùng muốn thay đổi ngôn ngữ hiển thị.
Điều kiện	- Người dùng phải đang chạy chương trình.
Dòng sự kiện	1. Chương trình yêu cầu chọn một trong 2 gói ngôn ngữ có sẵn: tiếng việt, tiếng anh. 2. Chương trình sẽ tự động cập nhật ngôn ngữ vào toàn bộ các màn hình trong chương trình.
Ngoại lệ	Không có

Bảng 13. Đặc tả use case thay đổi ngôn ngữ

1.14. Biểu đồ use case nhập thông tin người dùng và cài đặt thông số cảnh báo sức khỏe

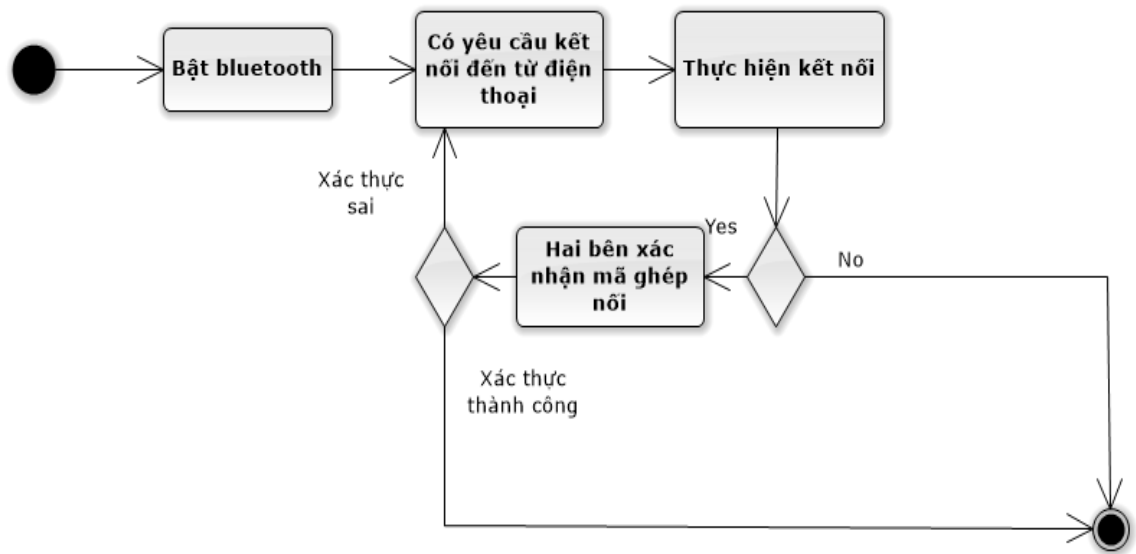


➤ Use case: Nhập thông tin người dùng và cài đặt thông số cảnh báo sức khỏe

Tên	Nhập thông tin người dùng và cài đặt thông số cảnh báo thông tin vận động
Mô tả chung	Trường hợp này cho phép người dùng nhập vào thông tin: chiều cao, cân nặng, tuổi. Đặt các thông số cảnh báo: target steps, thời gian cảnh báo không vận động.
Tác nhân	Người dùng
Xảy ra khi	Khi người dùng muốn nhập thông tin, thông số cài đặt.
Điều kiện	Chương trình đang chạy.
Dòng sự kiện	1. Người dùng chọn thông tin muốn nhập. 2. Chương trình hiện dialog để người dùng nhập liệu. 3. Chương trình lưu lại thông số người dùng đã nhập và áp dụng vào các thuật toán tính toán.
Ngoại lệ	Không có

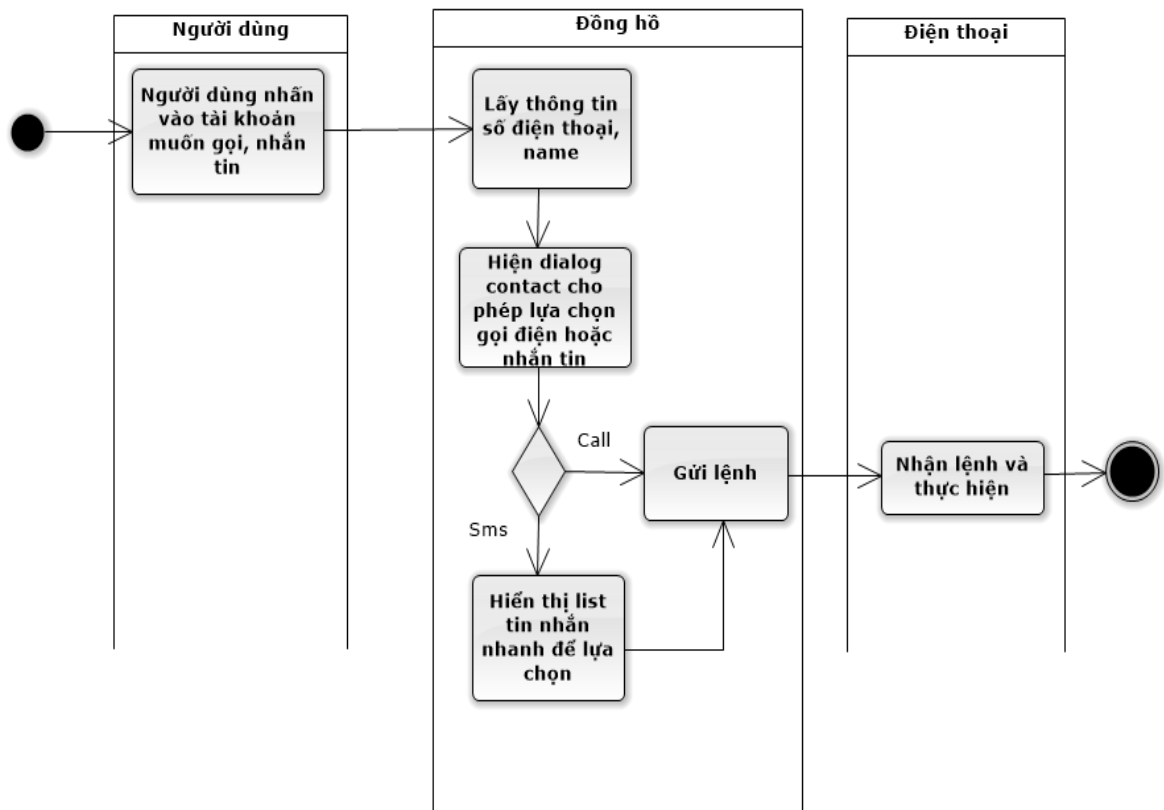
Bảng 14. Đặc tả use case nhập thông tin người dùng và cài đặt thông số cảnh báo sức khỏe

4. Biểu đồ hoạt động



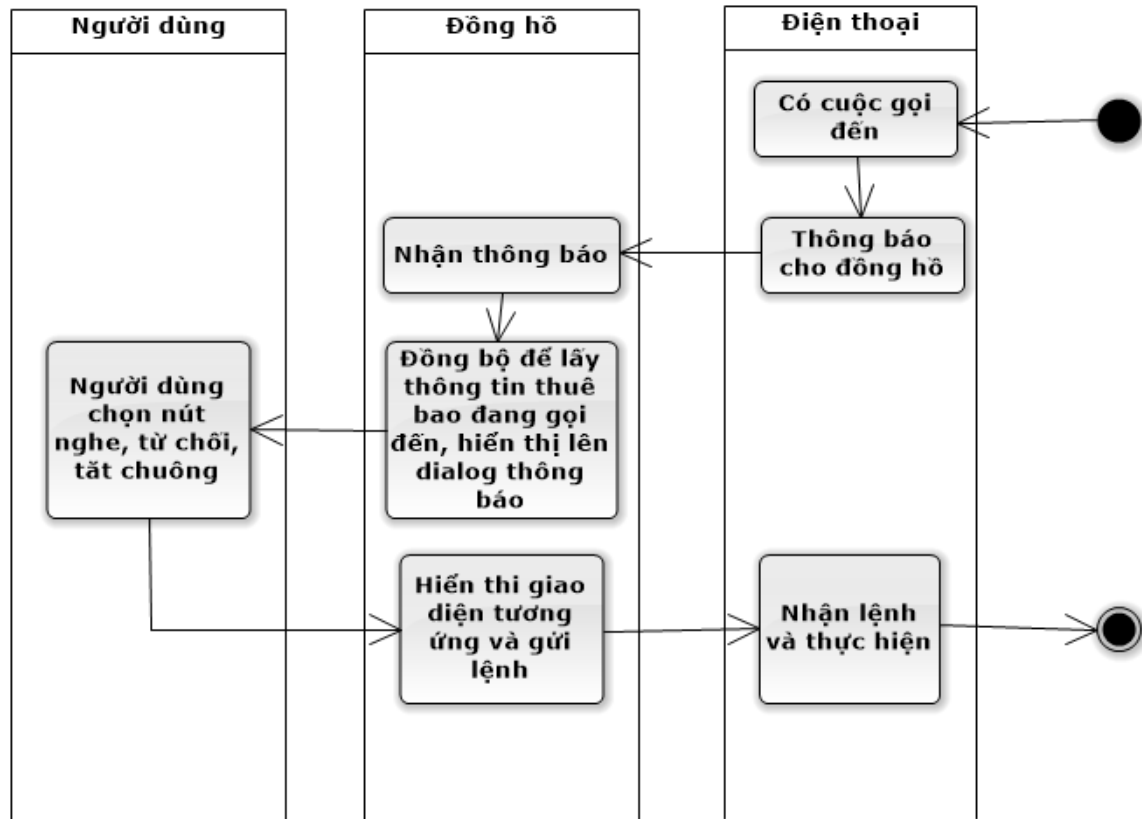
Hình 17. Biểu đồ hoạt động kết nối

- Biểu đồ hoạt động kết nối:
 - + Người dùng bật bluetooth trên đồng hồ.
 - + Đồng hồ sẽ luôn nằm ở trạng thái chờ kết nối, điện thoại cần tìm thiết bị và gửi yêu cầu kết nối đến đồng hồ.
 - + Khi có yêu cầu kết nối từ điện thoại, người dùng sẽ chọn thực hiện kết nối hoặc không.
 - + Khi chọn thực hiện kết nối, từ đồng hồ và điện thoại sẽ hiện lên mã xác thực ghép nối. Nếu 2 bên xác thực đúng, kết nối sẽ được khởi tạo.
 - + Nếu một trong hai bên không xác thực đúng, việc kết nối sẽ bị hủy bỏ, đồng hồ sẽ trở về trạng thái tiếp tục chờ kết nối.



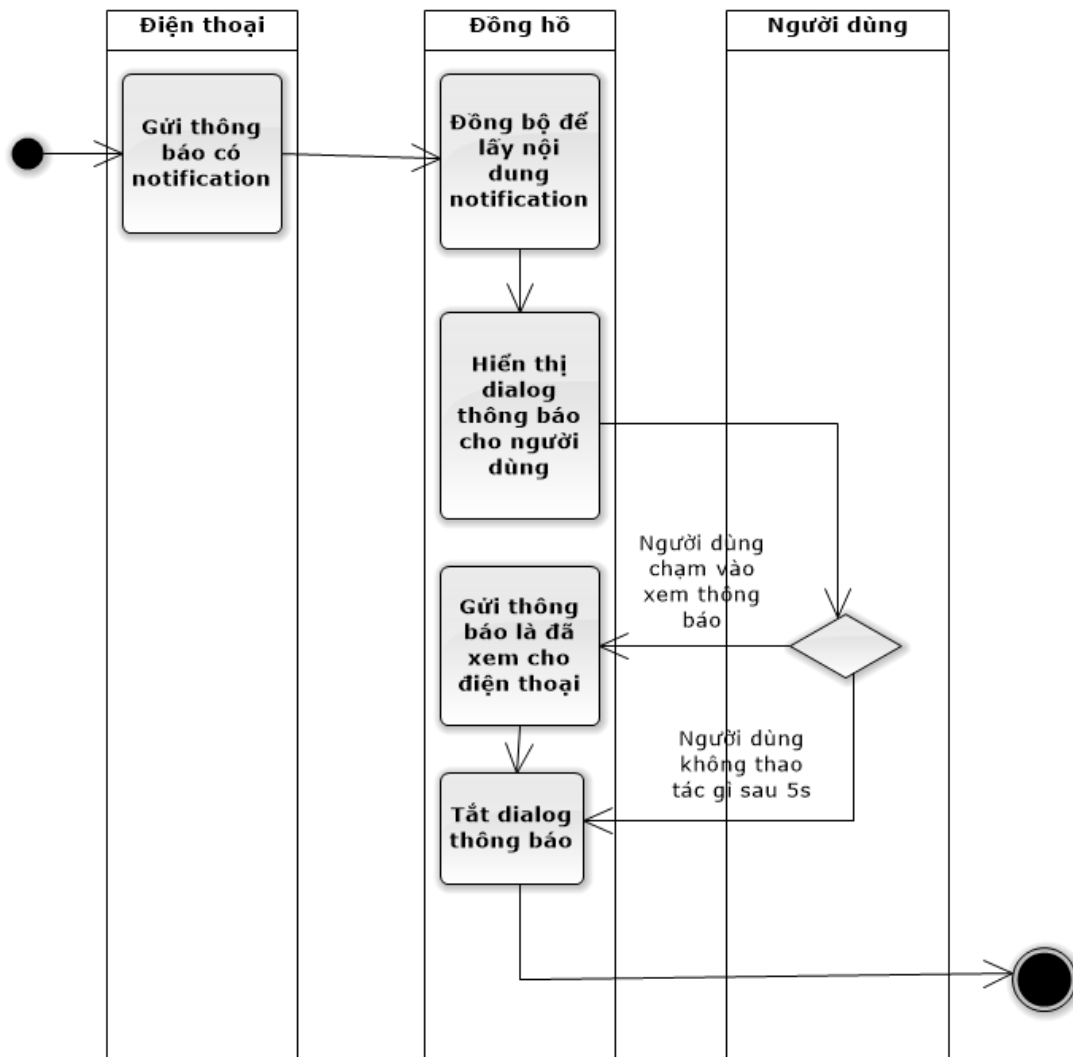
Hình 18. Biểu đồ hoạt động gọi điện, nhắn tin nhanh

- + Người dùng khi nhấn vào cuộc gọi trong màn hình cuộc gọi hoặc tin nhắn trong màn hình tin nhắn, một dialog sẽ hiển thị bao gồm thông tin tên thuê bao, số điện thoại của thuê bao tương ứng.
- + Trên dialog sẽ có 2 button để lựa chọn việc gọi điện hoặc nhắn tin nhanh.
- + Nếu người dùng chọn gọi điện, đồng hồ sẽ thực hiện gửi lệnh yêu cầu điện thoại gọi điện đến thuê bao đã chọn.
- + Nếu người dùng chọn gửi tin nhắn nhanh, một dialog nữa tiếp tục hiện lên và hiển thị danh sách các tin nhắn nhanh mà được đồng bộ từ điện thoại. Người dùng chọn một tin nhắn và đồng hồ sẽ gửi yêu cầu tương ứng tới điện thoại để điện thoại thực hiện nhắn tin.



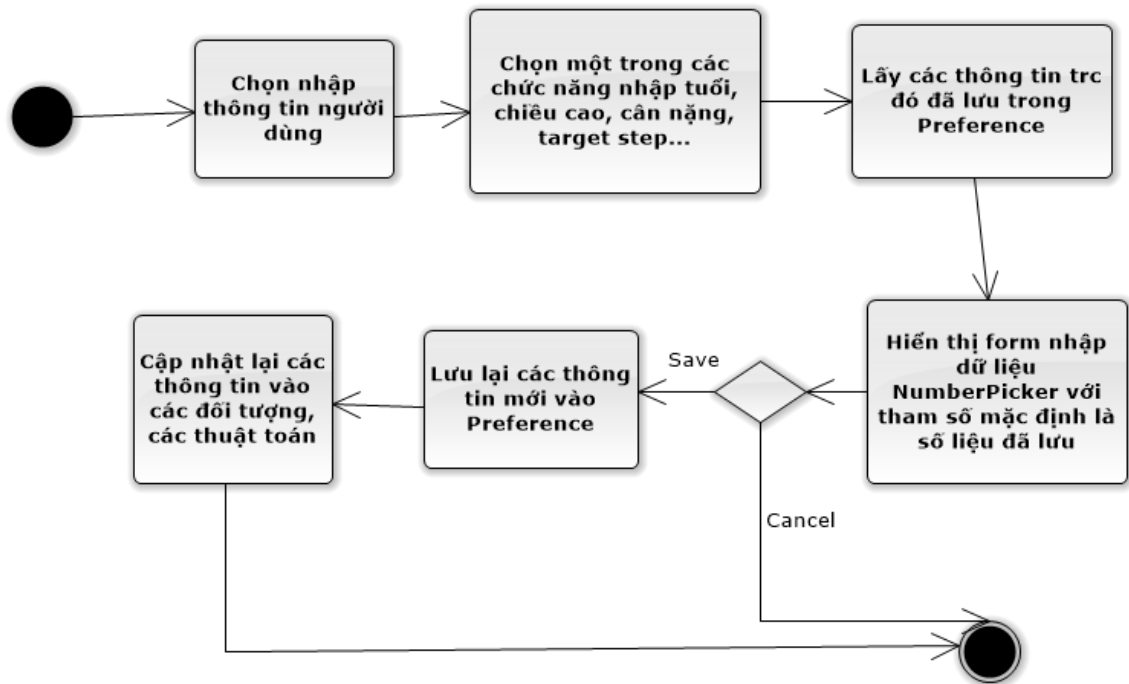
Hình 19. Biểu đồ hoạt động thông báo, trả lời cuộc gọi

- + Khi có cuộc gọi đến trên điện thoại, điện thoại sẽ gửi thông báo đến cho đồng hồ.
- + Đồng hồ nhận được thông báo sẽ thực hiện đồng bộ dữ liệu cho đối tượng incomingSms
- + Đồng hồ rung và hiển thị dialog thông báo cho người dùng biết. Trên dialog hiển thị tên và số điện thoại thuê bao gọi đến, đồng thời cho phép người dùng chọn nghe, từ chối hoặc tắt chuông điện thoại.
- + Khi người dùng chọn, đồng hồ sẽ gửi yêu cầu tương ứng đến điện thoại để thực hiện.



Hình 20. Biểu đồ hoạt động thông báo notification có trên điện thoại

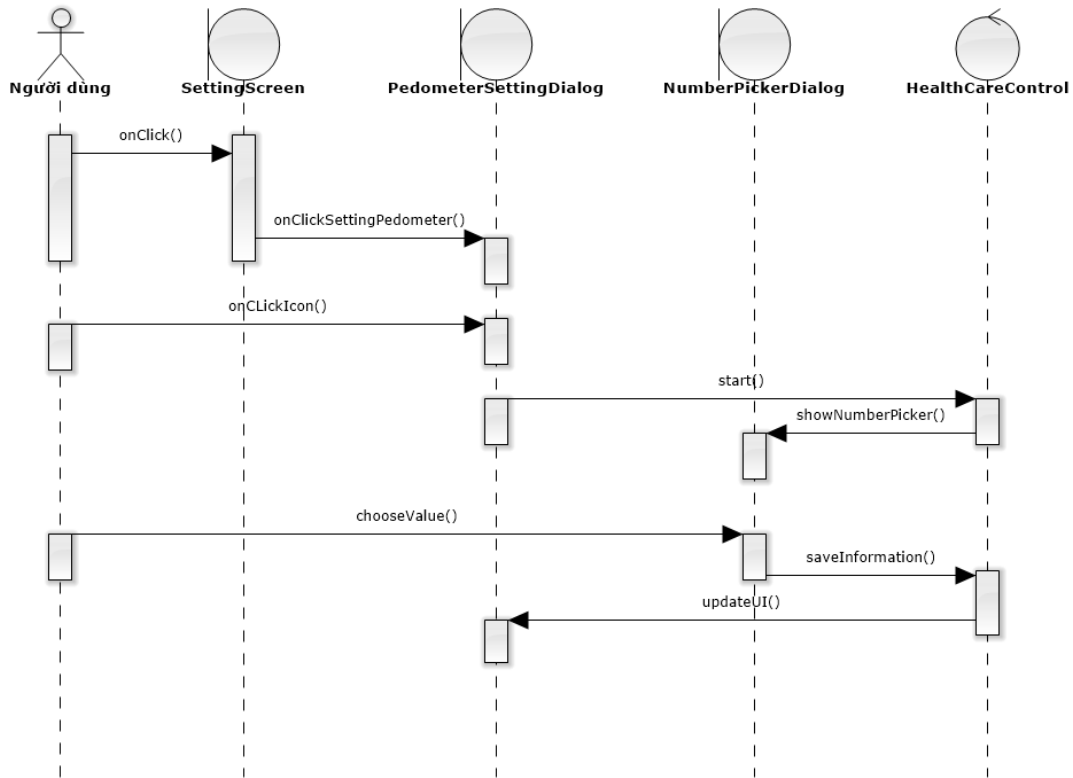
- + Khi trên điện thoại có 1 notification xuất hiện, có thể là từ cảnh báo của hệ thống hoặc từ bất kỳ một ứng dụng nào, app trên điện thoại sẽ lấy thông tin và gửi thông báo đến cho đồng hồ.
- + Đồng hồ trước tiên thực hiện đồng bộ dữ liệu để lấy nội dung thông tin cảnh báo và icon của ứng dụng cảnh báo.
- + Đồng hồ rung và hiển thị dialog thông báo cho người dùng biết. Khi người dùng chạm vào dialog notification đó, đồng hồ sẽ gửi lại cho điện thoại biết là đã xem để điện thoại xóa bỏ khỏi danh sách notification.
- + Nếu sau 5 s mà người dùng không thao tác gì, dialog sẽ tự tắt.



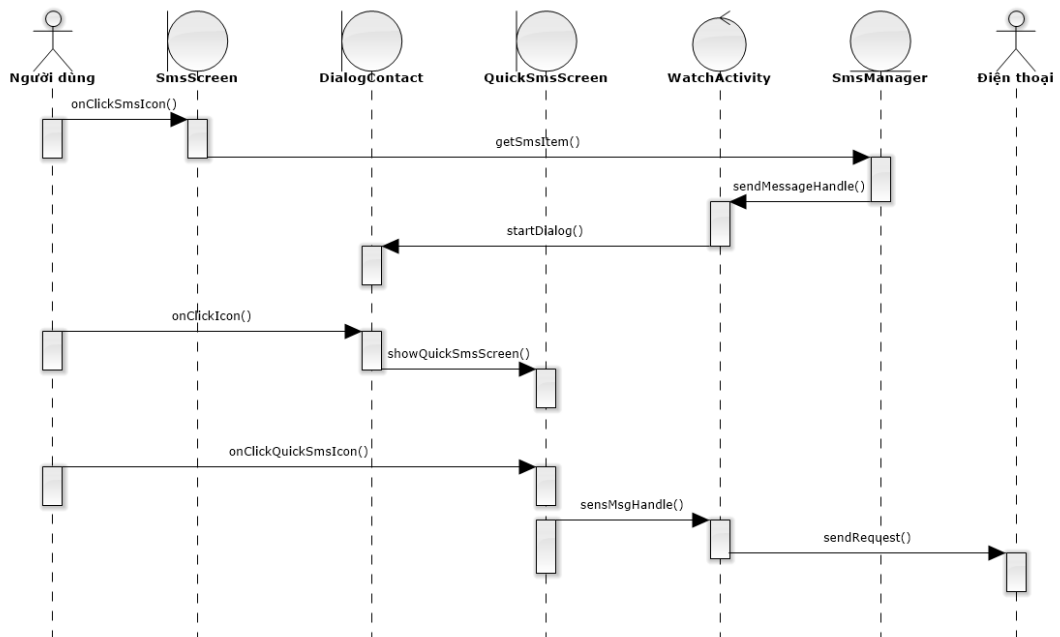
Hình 21. Biểu đồ hoạt động nhập thông tin

- + Khi người dùng chọn nhập thông tin, màn hình chứa form nhập thông tin sẽ hiện lên, các giá trị mặc định của form sẽ được lấy từ lần nhập trước đó, nếu là lần đầu tiên thì sẽ là các giá trị mặc định.
- + Người dùng sẽ chọn nhập các thuộc tính tuổi, chiều cao, cân nặng, số bước chân cần đạt được trong ngày...
- + Hiển thị form nhập dữ liệu là các numberPicker cho phép người dùng vuốt để chọn giá trị.
- + Nếu người dùng chọn lưu lại, các thông tin sẽ được lưu vào Preference, đồng thời sẽ cập nhật vào các đối tượng trong các luồng tính toán.
- + Nếu người dùng cancel, các giá trị vừa chọn sẽ không được lưu lại.

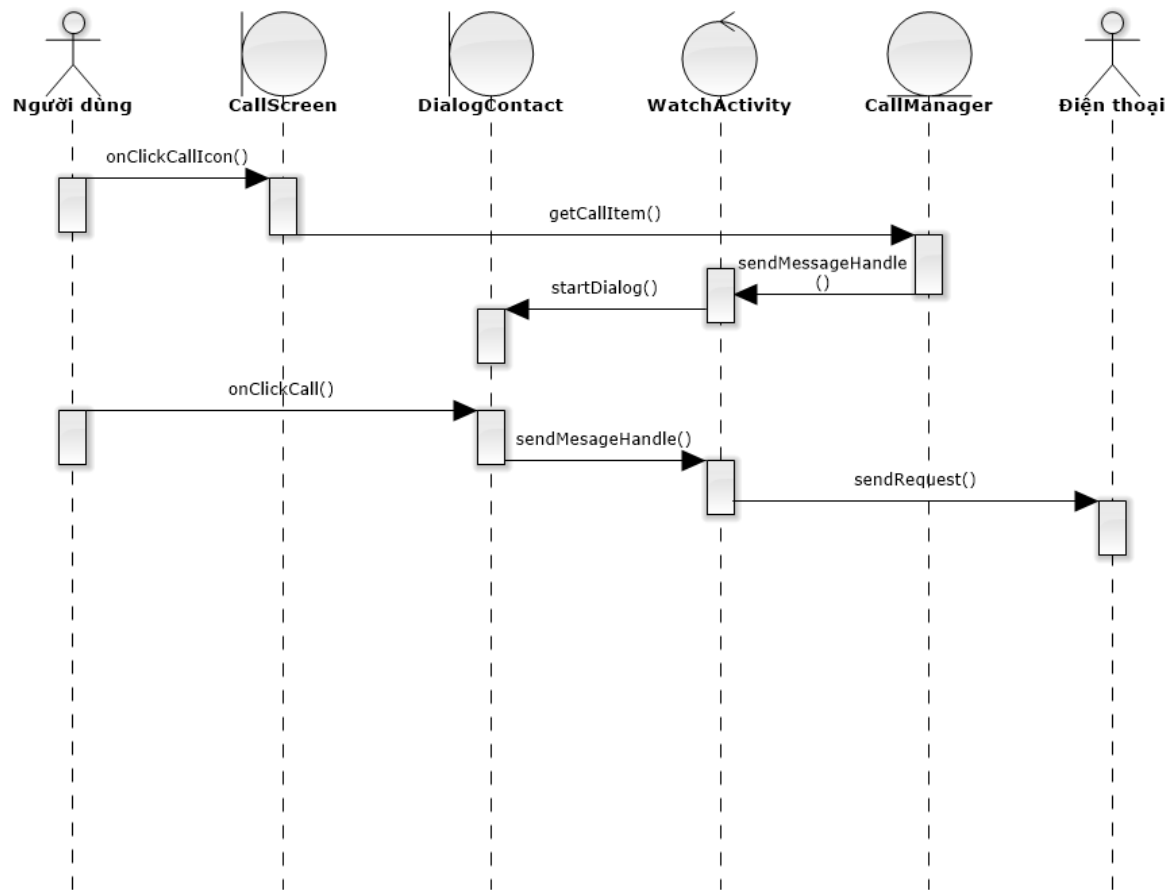
5. Biểu đồ trình tự



Hình 22. Biểu đồ trình tự nhập thông tin



Hình 23. Biểu đồ trình tự gửi tin nhắn nhanh

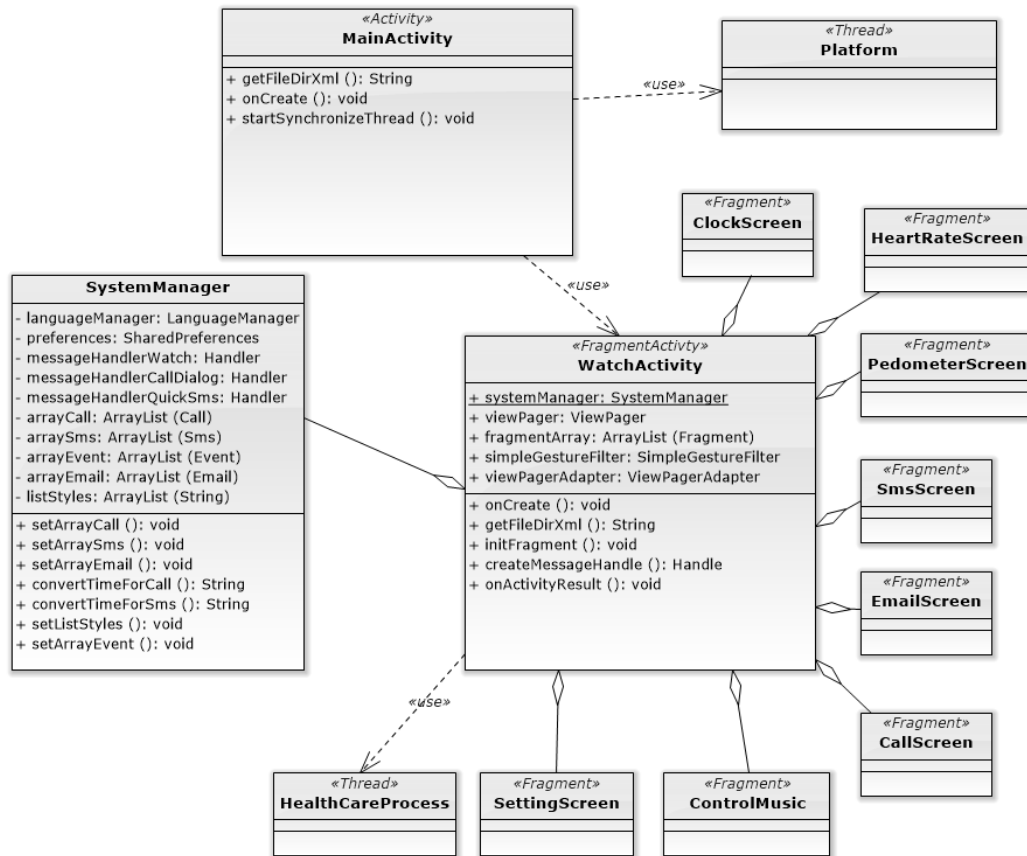


Hình 24. Biểu đồ trình tự gọi điện

- + Người dùng khi nhấn vào icon cuộc gọi trong màn hình cuộc gọi sẽ kích hoạt sự kiện hàm `onClickCallIcon`, một dialog sẽ hiển thị bao gồm thông tin tên thuê bao, số điện thoại của thuê bao tương ứng được lấy từ lớp quản lý dữ liệu cuộc gọi `CallManager`.
- + Trên dialog sẽ có button để lựa chọn việc gọi điện.
- + Người dùng chọn gọi điện, một message sẽ được gửi về handle ở lớp activity, ở đó sẽ thực hiện việc gửi một request về điện thoại yêu cầu gọi điện đến thuê bao đã được chọn.

II. Thiết kế ứng dụng

1.1. Thiết kế lớp



Hình 25. Biểu đồ lớp tổng quan

- MainActivity: Activity mà chương trình khi khởi động sẽ chạy vào đầu tiên, nó có vai trò khởi tạo các luồng của thư viện đồng bộ để tạo nên cấu trúc dữ liệu. Khi luồng khởi tạo dữ liệu chạy xong sẽ gọi đến WatchActivity.

- WatchActivity:

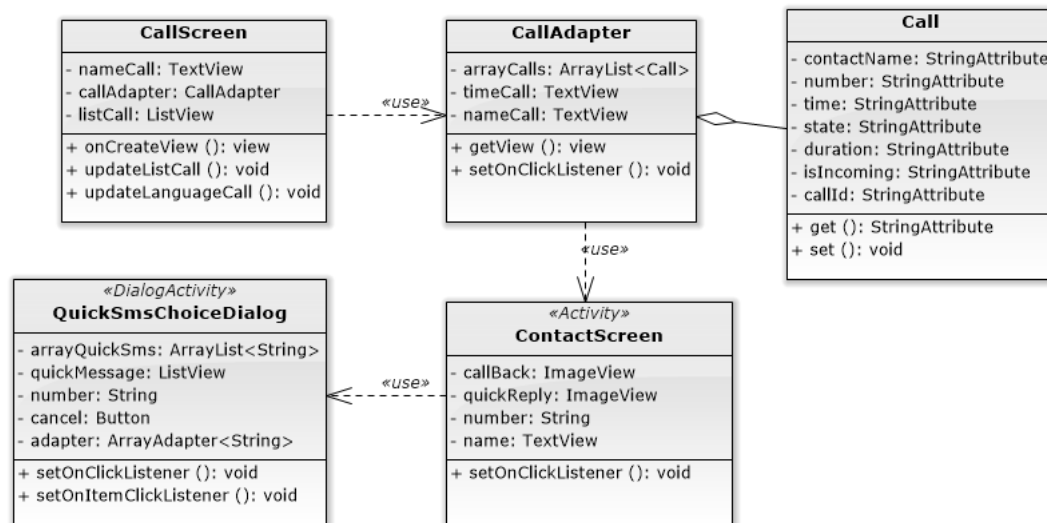
+ Lớp chính có vai trò xử lý hầu hết các công việc trong chương trình, định nghĩa việc update giao diện, gọi các sự kiện hiển thị dialog trên handle.

+ fragmentArray: chứa các màn hình được add vào view pager như SmsScreen, CallScreen...

+ getFileDirXml: Trả về đường dẫn vị trí lưu trữ dữ liệu trên bộ nhớ.

+ onActivityResult: Xử lý các kết quả trả về từ các activity được gọi start.

- **SystemManager**: Lớp định nghĩa và khởi tạo các đối tượng dữ liệu dùng trên toàn chương trình.



Hình 26. Biểu đồ các lớp màn hình cuộc gọi

- **CallScreen**: màn hình giao diện hiển thị danh sách cuộc gọi. Phương thức ghi đè `onCreateView` để khởi tạo và hiển thị các đối tượng giao diện. `ListCall` là listview hiển thị danh sách cuộc gọi. `updateListCall` hàm thực hiện việc update giao diện khi có dữ liệu thay đổi. `updateLanguageCall` hàm thực hiện việc chuyển ngôn ngữ khi người dùng chọn thay đổi ngôn ngữ hiển thị.

- **CallAdapter**: lớp khởi tạo các view cho `listCall`, khởi tạo truyền vào đối tượng dữ liệu **Call**, hàm `getView` khởi tạo các đối tượng view ứng với mỗi phần tử trong mảng dữ liệu.

- **Call**: Lớp dữ liệu định nghĩa một đối tượng call. Các thuộc tính bao gồm:

- + Tên thuê bao: `contactName`.
- + Số điện thoại: `number`.
- + Thời gian cuộc gọi đến: `time`.
- + Thời gian diễn ra cuộc gọi: `duration`.

+ Trạng thái là cuộc gọi đến (INCOMING) – cuộc gọi đi (OUTGOING) – cuộc gọi nhớ (MISSED): `state`.

- ContactScreen: Màn hình dialog hiện lên khi người dùng nhấn vào một cuộc gọi trong danh sách cuộc gọi.

+ number: số thuê bao tương ứng.

+ name: tên thuê bao

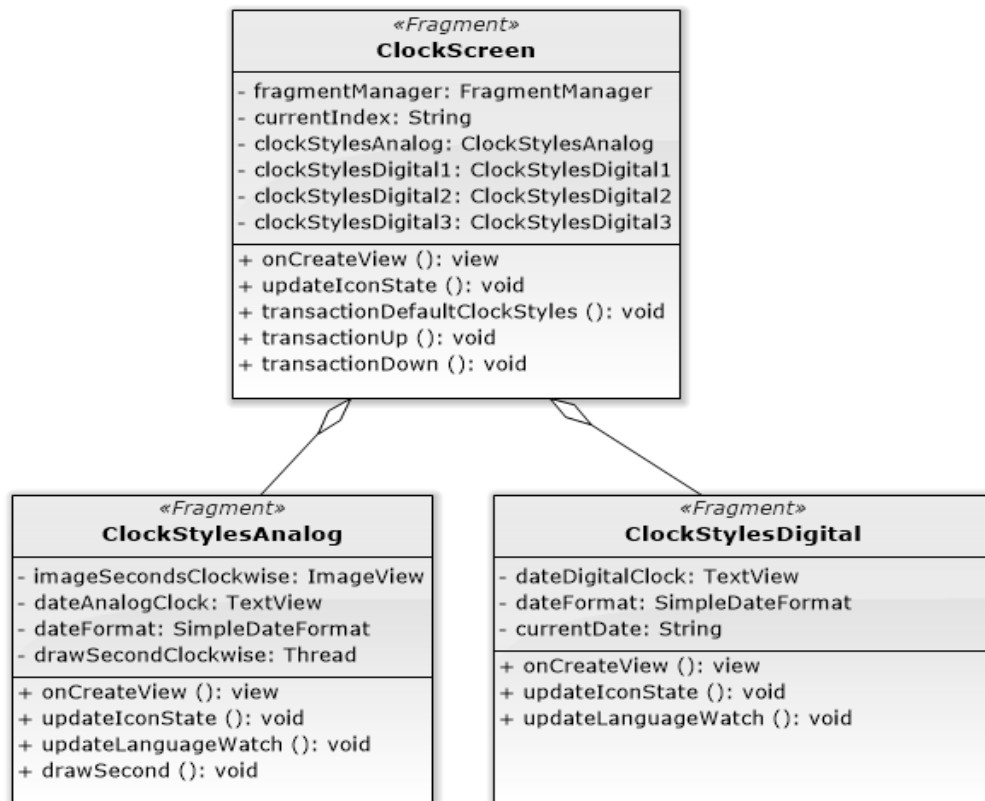
+ callBack: nhấn vào sẽ thực hiện việc gọi điện.

+ quickReply: nhấn vào sẽ thực hiện việc gửi tin nhắn nhanh.

- QuickSmsChoiceDialog: Màn hình dialog hiển thị danh sách các mẫu tin nhắn nhanh để người dùng lựa chọn.

+ arrayQuickSms : mảng chứa các tin nhắn nhanh đồng bộ từ điện thoại về.

+ adapter: khởi tạo các view



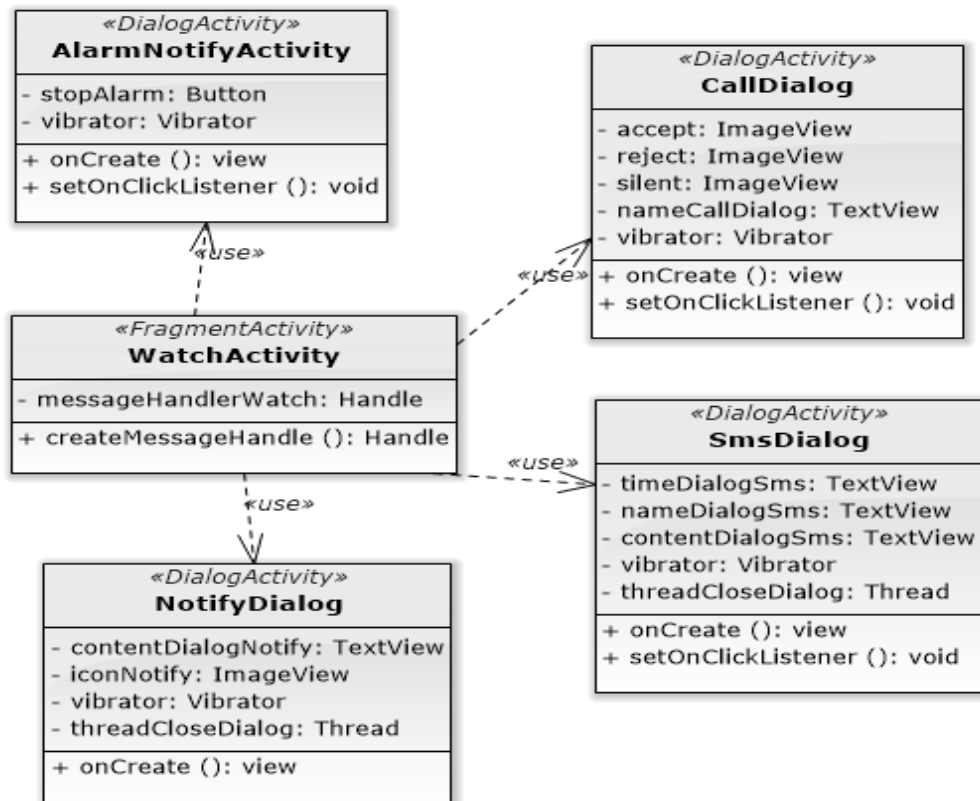
Hình 27. Biểu đồ các lớp màn hình đồng hồ

- ClockScreen: màn hình hiển thị đồng hồ, số cuộc gọi nhớ, số tin nhắn, email chưa đọc. Nó cho phép người dùng chọn các styles khác nhau bằng việc vuốt lên

hoặc xuống màn hình (`transactionUp()`, `transactionDown()`), hàm `updateIconState` thực hiện việc update các icon hiển thị số cuộc gọi nhỡ, số tin nhắn email chưa đọc.

- `ClockStylesAnalog`: view hiển thị đồng hồ ở dạng kim, hàm `drawSecond` : thực hiện việc vẽ kim giây.

- `ClockStylesDigital`: view hiển thị đồng hồ ở dạng số.



Hình 28. Biểu đồ các lớp dialog thông báo

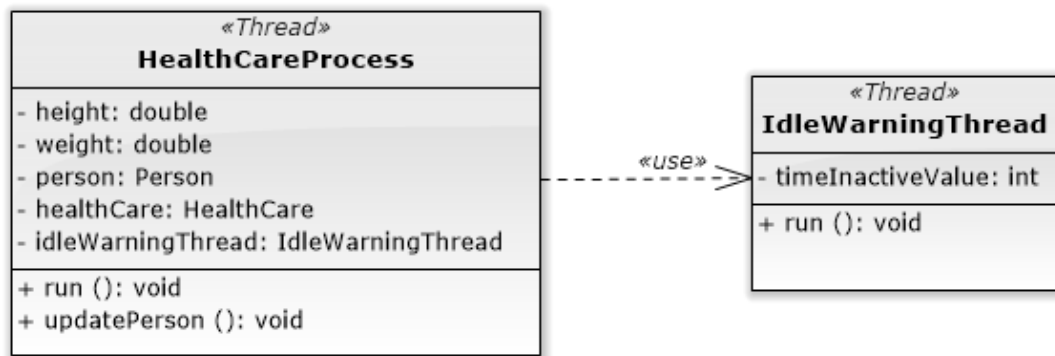
- `CallDialog`: dialog hiện lên khi có cuộc gọi đến.

- + `accept`: chấp nhận cuộc gọi
- + `reject`: từ chối cuộc gọi
- + `silent`: tắt chuông, tắt rung
- + `vibrator`: gọi chế độ rung

- `SmsDialog`: dialog hiện lên khi có tin nhắn mới

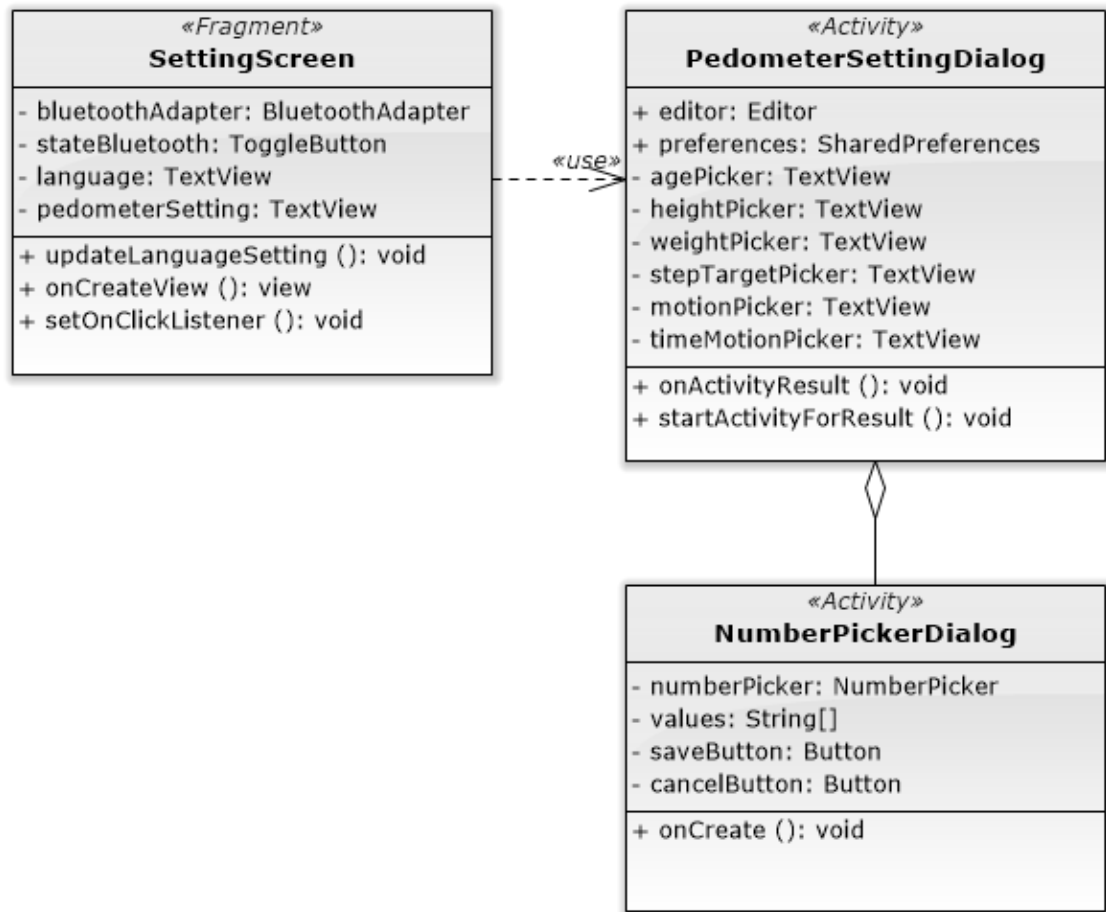
- + `timeDialogSms`: thời gian tin nhắn đến
- + `nameDialogSms`: tên người gửi
- + `contentDialogSms`: nội dung tin nhắn

- + threadCloseDialog: luồng thực hiện chức năng đóng dialog nếu sau 5s người dùng không tác động gì
- + vibrator: gọi chế độ rung báo cho người dùng biết.
- AlarmNotifyDialog: hiển thị khi đến giờ báo thức.
- NotifyDialog: hiển thị khi có một cảnh báo từ một ứng dụng bất kỳ trên điện thoại:
 - + contentDialogNotify: nội dung notification
 - + iconNotify: hiển thị icon của notification mà điện thoại gửi sang
 - + threadCloseDialog: luồng thực hiện chức năng đóng dialog nếu sau 5s người dùng không tác động gì.
 - + vibrator: gọi chế độ rung báo cho người dùng biết.



Hình 29. Biểu đồ lớp luồng đo, cảnh báo sức khỏe

- HealthCareProcess: Luồng thực hiện việc khởi tạo các đối tượng để đo thông số vận động
 - + Person: đối tượng thực hiện đo số bước chân, định nghĩa trong thư viện HeartCare.
 - + height: chiều cao người dùng, tham số truyền vào cho đối tượng person
 - + weight: cân nặng người dùng, tham số truyền vào cho đối tượng person
 - + idleWarningThread: luồng thực hiện việc cảnh báo khi các thông số ở mức không bình thường, hoặc là đạt chỉ tiêu.
 - + updatePerson: cập nhật thông tin đối tượng person khi người dùng thay đổi thông tin chiều cao, cân nặng.



Hình 30. Biểu đồ các lớp setting

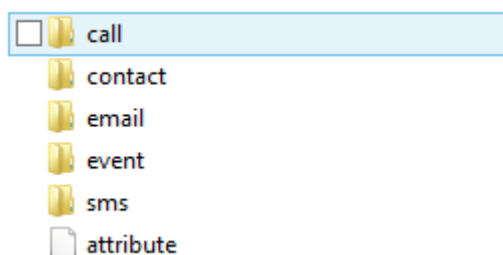
1.2. Thiết kế cơ sở dữ liệu

Các dữ liệu chuẩn để đồng bộ.

Các dữ liệu này sẽ được lưu trữ vào bộ nhớ chương trình, điều này vẫn đảm bảo tốc độ truy cập nhanh, tốc độ đồng bộ giữa hai thiết bị và nó cũng sẽ đảm bảo được bảo mật vì nó cần phải có quyền root thì người sử dụng mới có thể đọc được.

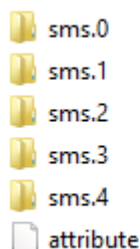
Các dữ liệu chuẩn để lưu trữ và đồng bộ ngầm gồm có: tin nhắn, email, nhật ký cuộc gọi, danh bạ, sự kiện.

Các file được tổ chức lưu trữ theo cây thư mục, cây thư mục được sinh ra khi khởi tạo các object để đồng bộ của hệ thống chương trình. Cụ thể như sau:



Hình 31. Cấu trúc cây thư mục watch

Ta có thư mục cha là *watch* bên trong nó chứa các folder đại diện cho các con của nó như *call*, *contact*, *email*, *event*, *sms* và một file *attribute* để lưu thông tin của đối tượng watch.



Hình 32. Cấu trúc cây thư mục sms

Tương tự như vậy bên trong thư mục cha sms sẽ là một tập các sms.id mà chương trình lấy được và lưu lại, mỗi sms.id là một folder và bên trong mỗi sms.id không còn con bé hơn nên nó sẽ chỉ có file attribute chứa giá trị các thuộc tính của một sms.

Ví dụ về định dạng thông tin một sms.id được lưu trong một file attribute:

{contactName:195,number:195,content:Viettel chương trình khuyến mãi tặng 50% cho mỗi thẻ nạp,time:06/01/2015 1\:36\:36, state:-1,type:1,smsId:1}.

File attribute được lưu theo cấu trúc “*tên thuộc tính:@giá trị*” . Tại đó *tên thuộc tính* là các thuộc tính đã đặt theo object và *@giá trị* là giá trị của thuộc tính đó.

1.3. Thiết kế giao diện

III. Thiết kế kiểm thử
a. Tính đúng đắn.

Tên chức năng	Tên test case	Điều kiện tiên quyết	Các bước thực hiện	Kết quả mong đợi	Người test	Ngày test
Kết nối	Kết nối khi có yêu cầu kết nối	Khi bật bluetooth	1. Khởi động chương trình	Thông báo kết quả kết nối	Dong NV	25/06/2015
Cập nhật dữ liệu	Cập nhật lại dữ liệu	Khi đang kết nối với điện thoại	1. Chạy chương trình	Thông tin mới về cuộc gọi, tin nhắn... được cập nhật liên tục mỗi khi dữ liệu có thay đổi	Dong NV	25/06/2015
Điều khiển điện thoại	Điều khiển chơi nhạc	Khi đang kết nối với điện thoại	1. Chọn play nhạc	Điện thoại bắt đầu phát nhạc	Dong NV	25/06/2015
	Điều khiển gọi điện		1. Chọn gọi điện	Điện thoại thực hiện cuộc gọi tới thuê bao được chọn		
	Điều khiển nhắn tin nhanh		2. Chọn gửi tin nhắn nhanh	Điện thoại gửi tin nhắn nhanh đã chọn		
Nhập thông tin người dùng	Nhập chiều cao, cân nặng	Khi chương trình đang chạy	1. Vào màn hình nhập thông tin 2. Nhấn vào từng thông tin để hiện lên dialog nhập.	Cập nhật trên giao diện hiển thị, cập nhật vào đối tượng heart care để từ số bước chân tính	Dong NV	25/06/2015

				quãng đường được phù hợp.		
	Nhập targetStep			Cập nhật giá trị trên màn hình hiển thị, cập nhật vào luồng cảnh báo vận động.	Dong NV	25/06/2015
Cài đặt ứng dụng	Cài đặt ngôn ngữ	Khi chương trình đang chạy	1. Chọn màn hình Setting 2. Chọn ngôn ngữ mong muốn	Ứng dụng thay đổi ngôn ngữ đúng theo tùy chọn	Dong NV	25/06/2015
	Cài đặt bluetooth	Khi chương trình đang chạy	1. Chọn màn hình Setting 2. Chọn tùy chọn bật tắt bluetooth	Ứng dụng thông báo cho người dùng.	Dong NV	25/06/2015
Nhận thông báo	Nhận thông báo từ điện thoại	Khi có kết nối	1. Nhận thông báo 2. Hiện thị dialog tương ứng	Thông báo cho người dùng khi trên điện thoại phát sinh sự kiện	Dong NV	25/06/2015

Bảng 15. Thiết kế kiểm thử tính đúng đắn

b. Độ tin cậy.

STT	Hành động	Kết quả	Thực tế	Người thực hiện	Ngày thực hiện
1	Mất kết nối bluetooth	Tự động connect lại địa chỉ cũ	Có / Không.	DongNV	25/05/2015
2	Đồng bộ dữ liệu	Đồng bộ được dữ liệu khi có kết nối	Có / Không.	DongNV	25/05/2015
3	Gửi lệnh điều khiển	Điện thoại thực hiện đúng yêu cầu	Có / Không.	DongNV	25/05/2015

		gửi sang			
4	Đo thông số	Đo được thông số vận động và thông số sức khỏe	Có / Không.	DongNV	25/05/2015

Bảng 16. Thiết kế kiểm thử tính tin cậy

c. Tính dùng được

STT	Hành động	Thực tế	Người thực hiện	Ngày thực hiện
1	Có thông báo tình trạng kết nối không?	Có / Không.	DongNV	25/05/2015
2	Hình ảnh, font chữ, giao diện có đẹp không?	Có / Không.	DongNV	25/05/2015
3	Ứng dụng có đơn giản dễ dùng không?	Có / Không.	DongNV	25/05/2015
4	Có thông báo sự kiện bằng rung không?	Có / Không.	DongNV	25/05/2015
5	Phím chức năng có dễ bấm không?	Có / Không.	DongNV	25/05/2015

Bảng 17. Thiết kế kiểm thử tính dùng được

IV. Các kịch bản thử nghiệm

Đưa ra các kết quả cần đạt được khi thử nghiệm ứng dụng:

Màn hình đồng hồ

- Gồm các màn hình đồng hồ với nhiều styles khác nhau (đồng hồ kim, đồng hồ điện tử..), người dùng có thể lựa chọn bằng cách vuốt lên xuống.
- Trên màn hình đồng hồ có 3 icon hiển thị số cuộc gọi nhỡ, số tin nhắn mới chưa đọc, số email mới chưa đọc.
- Khi số cuộc gọi nhỡ, số tin nhắn mới chưa đọc, số email mới mà bằng 0 thì icon sẽ tự động bị ẩn đi.
- Nhấn vào mỗi icon sẽ đến được màn hình chi tiết tương ứng.
- Vuốt ngang từ phải sang trái để đến màn hình chức năng tiếp theo.

Màn hình cuộc gọi

- Hiện thị đầy đủ các cuộc gọi liên lạc trên điện thoại với các thông tin: thời gian gọi, người gọi, thời lượng cuộc gọi diễn ra, trạng thái cuộc gọi (cuộc gọi đến, gọi đi, gọi nhỡ).
- Khi nhấn vào mỗi cuộc gọi sẽ có màn hình thông báo (dialog) hiển thị lên cho phép người dùng gọi lại hoặc gửi tin nhắn đến thuê bao đó.
- Vuốt ngang từ phải sang trái để đến màn hình chức năng tiếp theo hoặc ngược lại để quay về màn hình trước đó.

Màn hình tin nhắn

- Hiện thị đầy đủ các tin nhắn trên điện thoại với các thông tin: thời gian, người gửi, nội dung tin nhắn.
- Khi nhấn vào mỗi tin nhắn sẽ có dialog hiển thị cho phép người dùng gọi lại hoặc gửi tin nhắn đến thuê bao đó.
- Vuốt ngang từ phải sang trái để đến màn hình chức năng tiếp theo hoặc ngược lại để quay về màn hình trước đó.

Màn hình email

- Hiện thị các email trong hộp thư đến của tài khoản mail mà người dùng điền trên app.
- Vuốt ngang từ phải sang trái để đến màn hình chức năng tiếp theo hoặc ngược lại để quay về màn hình trước đó.

Màn hình sức khỏe

- Thống kê số bước đi của người dùng.
- Dựa trên các thông số về hình thể như chiều cao, cân nặng để tính ra quãng đường đi được, lượng calo tiêu thụ.
- So sánh với các mục tiêu mà người dùng đặt ra, qua đó hỗ trợ người dùng tập luyện một cách hiệu quả.

Màn hình setting

- Bật tắt kết nối bluetooth.
- Điều chỉnh độ sáng màn hình.
- Đặt thời gian màn hình sleep để tiết kiệm pin.
- Cài đặt ngôn ngữ hiển thị (tiếng Anh – tiếng Việt).

Các màn hình thông báo

- Coming call: hiển thị lên khi có cuộc gọi đến, trên màn hình này cho phép người dùng các tùy chọn chấp nhận cuộc gọi, từ chối cuộc gọi hoặc để về trạng thái im lặng. Khi người dùng chọn 1 trong 3 chức năng trên hoặc khi cuộc gọi hết thời gian chờ, dialog sẽ tự động tắt và chương trình trở về màn hình đồng hồ.

- New sms: Hiển thị lên khi có tin nhắn mới đến. Khi người dùng nhấn vào, chương trình sẽ nhảy đến màn hình tin nhắn. Nếu người dùng không thao tác gì, dialog cũng sẽ tự tắt sau 10s và trở về màn hình đồng hồ.
- New email: Hiển thị lên khi có email mới đến. Khi người dùng nhấn vào, chương trình sẽ nhảy đến màn hình email. Nếu người dùng không tác động gì thì dialog cũng sẽ tự tắt sau 10s và trở về màn hình đồng hồ.

Đo nhịp tim:

- Người dùng nhấn vào icon để bắt đầu đo nhịp tim
- Giữ nguyên tay không chuyển động để kết quả đo được chính xác hơn, kết quả sẽ có được sau khoảng 15-30s

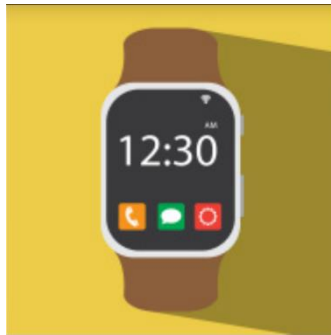
Điều khiển nhạc:

- Người dùng có thể chọn chức năng chơi nhạc trên điện thoại từ màn hình này.

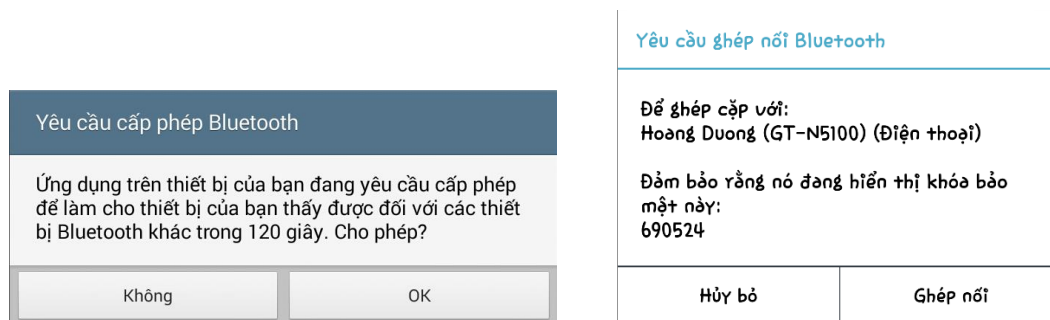
V. Cài đặt chương trình và kết quả đạt được

1. Giao diện khi sử dụng

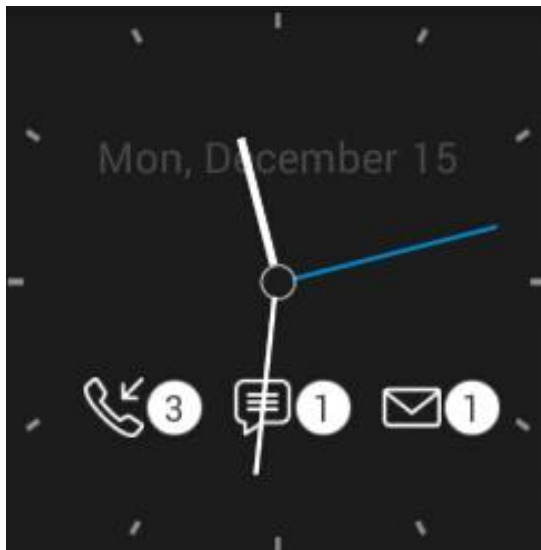
Một số hình ảnh khi chạy chương trình:



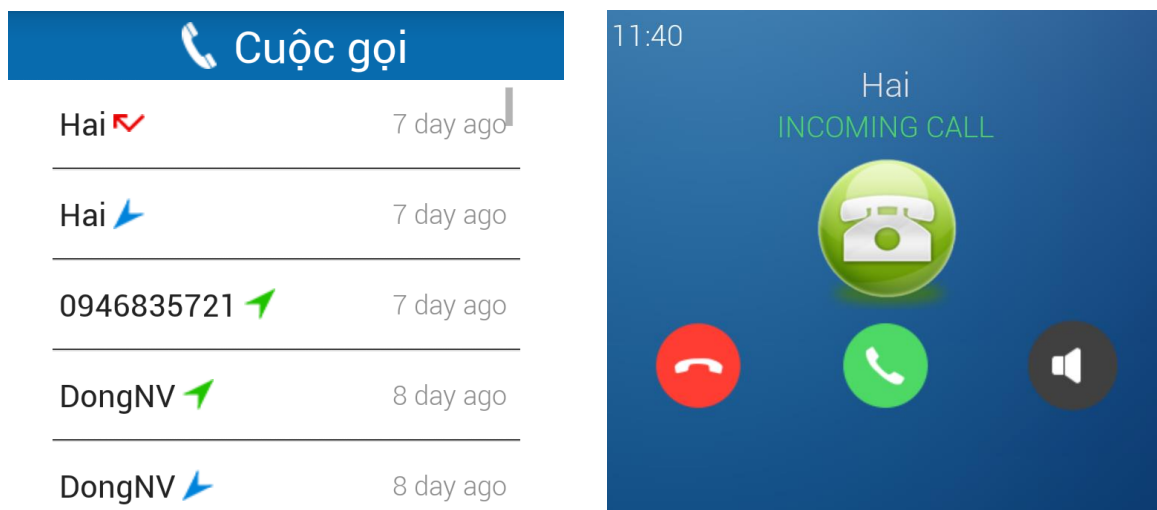
Hình 33. Màn hình khởi động



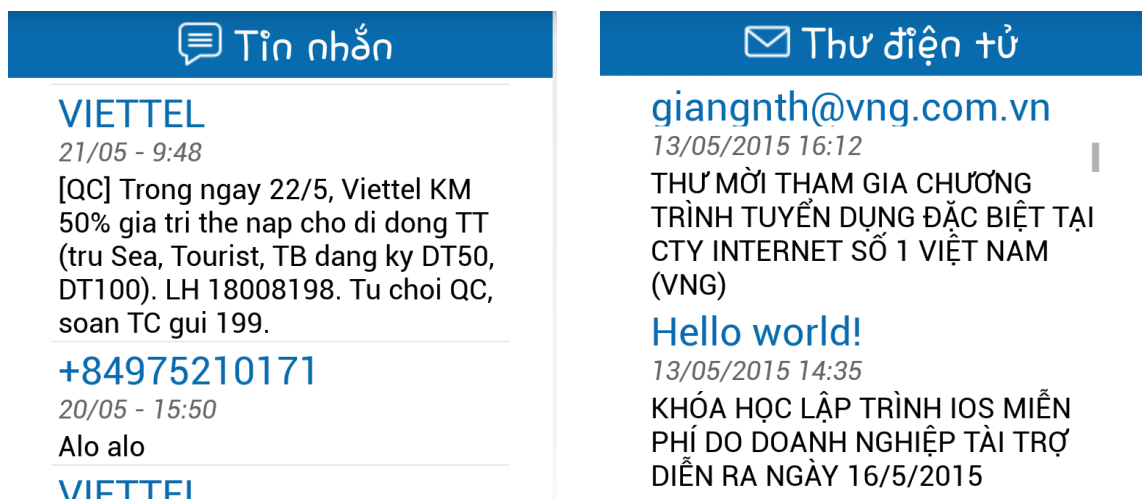
Hình 34. Màn hình yêu cầu và xác thực kết nối



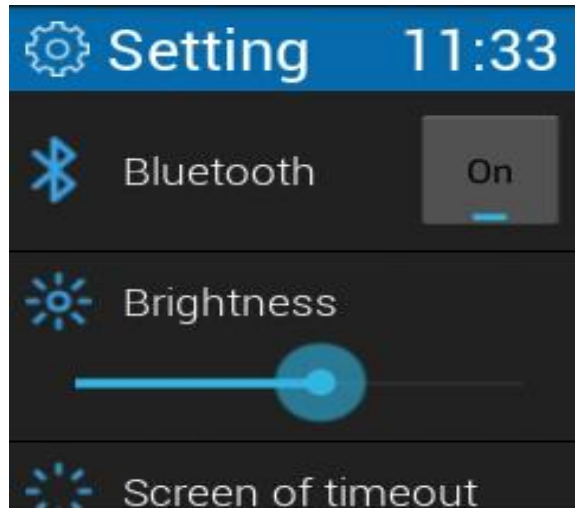
Hình 35. Màn hình đồng hồ



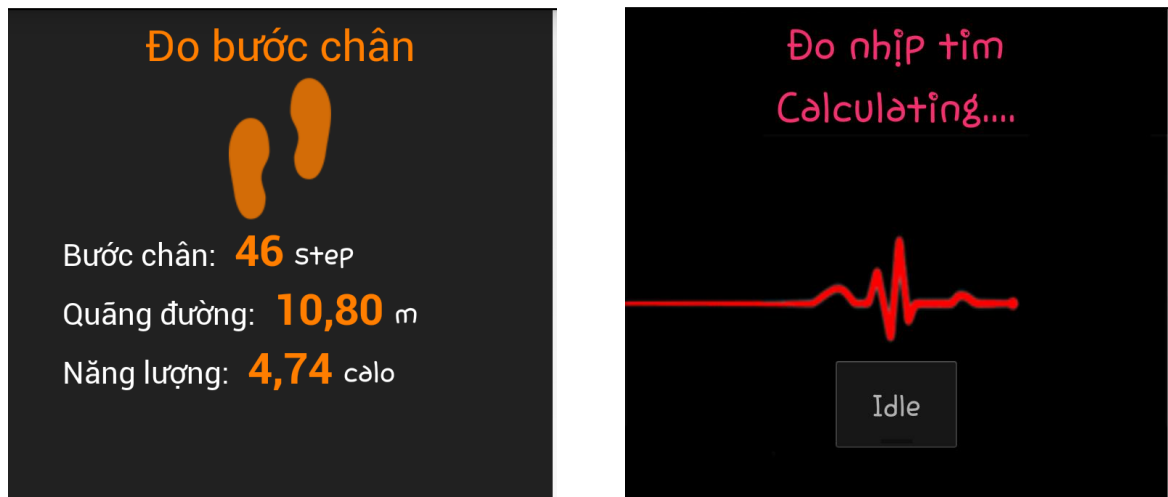
Hình 36. Màn hình cuộc gọi



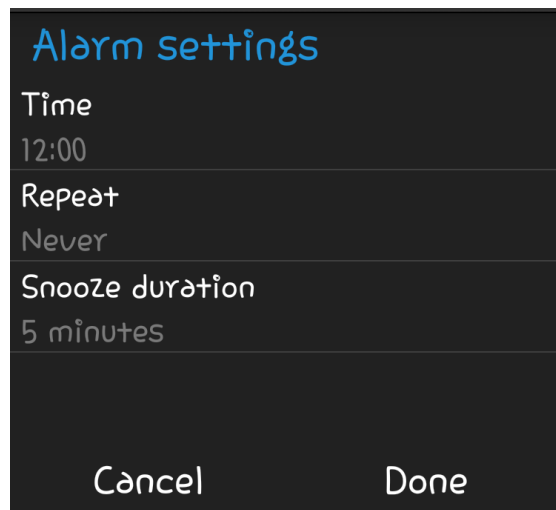
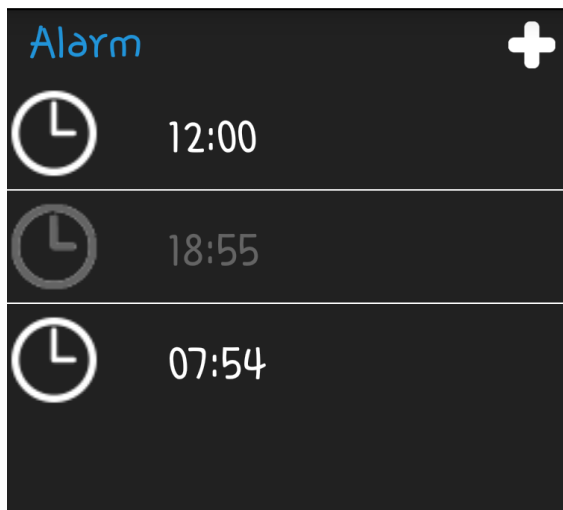
Hình 37. Màn hình tin nhắn, email



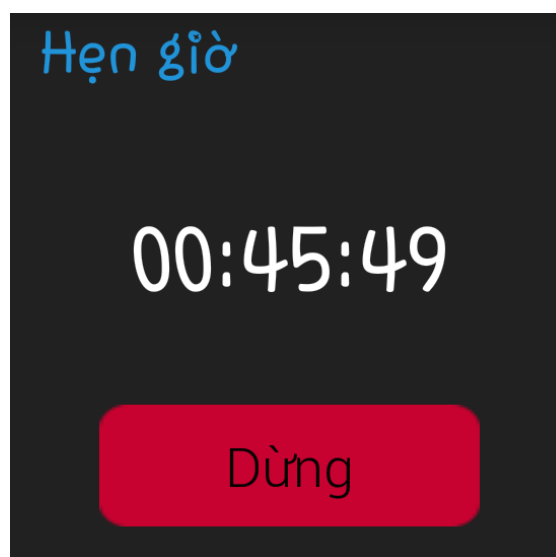
Hình 38. Màn hình setting



Hình 39. Màn hình vận động và nhịp tim



Hình 40. Màn hình đặt báo thức



Hình 41. Màn hình đồng hồ bấm giờ, hẹn giờ

2. Đánh giá chung về ứng dụng

Ưu điểm

- Ứng dụng đã giải quyết gần như đầy đủ các vấn đề đã đặt ra.
- Giao diện đơn giản, trực quan dễ dàng sử dụng, khả năng đa ngôn ngữ dễ dàng được mở rộng.
- Hỗ trợ tự động kết nối và kết nối lại với điện thoại đã từng kết nối.
- Dữ liệu luôn được đồng bộ một cách tự động.
- Giao diện tự động cập nhật mỗi khi có dữ liệu thay đổi.

Nhược điểm

- Do khả năng thiết kế của em còn nhiều hạn chế nên các hiệu ứng giao diện chưa đẹp mắt.
- Do hạn chế về thuật toán, thiết bị phần cứng nên việc đo các thông số vận động, chăm sóc sức khỏe như số bước chân, nhịp tim còn chưa đo được sự chính xác.

Một số hướng khắc phục

- Trau chuốt lại các thiết kế icon, hỗ trợ tốt hơn các hiệu ứng chuyển màn hình cho người dùng.
- Tiếp tục tối ưu thuật toán để các thông số đo được chuẩn xác hơn.

CHƯƠNG III – KẾT LUẬN

1. Kết luận chung

Sau một thời gian nghiên cứu và phát triển dưới sự chỉ bảo của thầy hướng dẫn, đồ án tốt nghiệp đã hoàn thành đầy đủ chức năng cơ bản đã đặt ra.

Phần mềm nhóm các tiện ích trên đồng hồ thông minh có thể thực hiện các chức năng:

- Thực hiện các chức năng về thời gian như một đồng hồ thông thường.
- Kết nối với smartPhone qua kết nối bluetooth.
- Đồng bộ dữ liệu về thông tin cuộc gọi, tin nhắn, email, event, notification...
- Nhận được thông báo cuộc gọi đến, tin nhắn mới, email mới, các sự kiện và cảnh báo xuất hiện trên điện thoại.
- Điều khiển điện thoại thực hiện một số chức năng như chơi nhạc, gọi điện, gửi tin nhắn nhanh.
- Đo thông số vận động bước chân, tính quãng đường, calo đi được, đưa ra cảnh báo vận động.
- Đo thông số nhịp tim, đưa ra cảnh báo

Các chức năng phần mềm hoạt động tốt, ổn định, đảm bảo tính logic và hiệu năng sử dụng, có tính áp dụng thực tiễn cao trong đời sống.

2. Định hướng phát triển của ứng dụng

Ứng dụng còn cần bổ sung, nâng cấp các tính năng:

- Tiếp tục phát triển thuật toán tính toán để có thể có được các thông số vận động, thông số sức khỏe chính xác hơn.
- Tối ưu việc đồng bộ dữ liệu để cập nhật nhanh chóng hơn.
- Thiết kế giao diện thuận tiện nhất cho người sử dụng, thêm các hiệu ứng để đồng hồ không chỉ là sản phẩm ứng dụng mà còn có tính thời trang.

TÀI LIỆU THAM KHẢO

- Bài giảng “PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN” của Trần Đình Quế - Nguyễn Mạnh Sơn – Học viện công nghệ bưu chính viễn thông – 2007
- Dương Anh Đức, “Bài giảng Phân tích thiết kế hướng đối tượng sử dụng UML”, Đại học KHTN – Đại học Quốc gia TP HCM, 9-2000