

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO - ĐỒ ÁN 2

Ứng dụng hỗ trợ chăm sóc sức khỏe - WeCare

Giảng viên hướng dẫn: Thái Thụy Hàn Uyên

Sinh viên thực hiện:

LÊ ĐỨC TRUNG

19522422

HỒ HỮU THỊNH

19522275

Tp. Hồ Chí Minh, 12/2022

LỜI CẢM ƠN

Trước khi bước vào phần báo cáo đồ án, nhóm chúng em trân trọng gửi lời cảm ơn đến cô Thái Thụy Hàn Uyên đã tạo điều kiện cho chúng em có cơ hội được thực hiện đồ án và đã nhiệt hỗ trợ, cung cấp cho chúng em rất nhiều kiến thức và cả kinh nghiệm hữu ích trong và ngoài mảng lập trình di động. Đồng thời cô cũng đã hỗ trợ những thông tin cần thiết, gợi mở những trường hợp sẽ gặp phải vấn đề khi lập trình trong thực tế, đưa ra những giải pháp tối ưu và giải đáp những thắc mắc không chỉ cho nhóm chúng em trong suốt quá trình học tập và thực hiện đồ án.

Trong vòng hơn 3 tháng qua, nhờ sự chỉ dẫn nhiệt tình của cô, chúng em đã tiếp thu được những kiến thức quan trọng cùng những góp ý chân thành để có thể làm được một chương trình hoàn chỉnh.

Cũng xin cảm ơn thầy cô và bạn bè trong khoa Công nghệ phần mềm đã nhiệt tình hỗ trợ, tạo điều kiện cho nhóm em làm bài báo cáo này.

Trong quá trình học tập, cũng như là trong quá trình làm đồ án, nhóm em đã cố gắng hết sức mình, vì thời gian và kiến thức có hạn nên nhóm em không thể tránh khỏi những thiếu sót. Nhóm em mong nhận được sự thông cảm, đóng góp và xây dựng từ cô để nhóm em có thêm nhiều kiến thức có ích và ngày càng hoàn thiện hơn.

Nhóm em xin chân thành cảm ơn ạ!

Nhóm thực hiện

Trường Đại học Công nghệ Thông tin, tháng 12 năm 2022

MỤC LỤC

CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI	5
1. Giới thiệu chung	5
2. Mục tiêu đề tài	5
2.1. Lý thuyết	5
2.2. Mục tiêu	5
3. Quá trình hiện thực	5
4. Tổng quan về chức năng	6
CHƯƠNG II: KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU	7
1. Phương pháp khảo sát	7
2. Câu hỏi khảo sát	7
3. Kết quả khảo sát	7
4. Xác định yêu cầu	7
CHƯƠNG III: PHÂN TÍCH YÊU CẦU	9
1. Sơ đồ Usecase	9
2. Mô tả use case	10
3. Đặc tả usecase	11
3.1. Đăng ký	11
3.2. Đăng nhập bằng email và mật khẩu	13
3.3. Đăng nhập với Google	14
3.4. Quên mật khẩu	17
3.5. Theo dõi chỉ số khi tập thể dục	18
3.6. Xem lại quãng đường đã đi trong quá trình tập thể dục	19
3.7. Theo dõi chỉ số bước chân đi được theo ngày	21
3.8. Đặt mục tiêu cho cho bước chân cần đi mỗi ngày	22
3.9. Xem thống kê số bước chân	24
3.10. Thay đổi thông tin cá nhân	26
3.11. Thay đổi mật khẩu	27
3.12. Đăng xuất	28
CHƯƠNG IV: CƠ SỞ LÝ THUYẾT	31
1. Android	31

1.1. Tổng quan và lịch sử	31
1.2. Các thành phần cơ bản của Android App	33
2. Ngôn ngữ lập trình Kotlin	36
2.1. Lịch sử hình thành	36
3. Hệ quản trị cơ sở dữ liệu Firebase	37
3.1. Firebase là gì?	37
3.2. Firebase Realtime database	37
3.3. Firebase Authentication	38
3.4. Firebase Hosting	38
3.5. Ưu điểm của Firebase	38
3.6. Nhược điểm của Firebase	39
CHƯƠNG V: XÂY DỰNG HỆ THỐNG	39
1. Xây dựng kiến trúc hệ thống	39
1.1. Công nghệ sử dụng	39
1.2. Kiến trúc hệ thống: Mô hình Client-Server	40
1.3. Kiến trúc ứng dụng: Mô hình MVVM	41
1.4. Mô tả thành phần hệ thống	42
2. Mô tả các màn hình	43
2.1. Màn hình đăng nhập	43
2.2. Màn hình đăng ký	45
2.3. Màn hình Home	47
2.4. Màn hình Exercise Landing Page	47
2.5. Màn hình chuẩn bị chạy bộ	49
2.6. Màn hình đang đi bộ	51
2.7. Màn hình hoàn thành Exercise	53
2.8. Màn hình Replay quãng đường đã đi	55
2.9. Màn hình đếm và thống kê số bước chân	57
2.10. Màn hình quản lý tài khoản	59
CHƯƠNG VI: KẾT LUẬN	61
1. Kết quả đạt được	61
2. Ưu điểm	61
3. Nhược điểm và giải pháp	62
TÀI LIỆU THAM KHẢO	64

CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI

1. Giới thiệu chung

Sức khỏe đã và đang là vấn đề được người dùng quan tâm nhất ở thời điểm hiện tại tuy nhiên không phải ai cũng có thể dành nhiều thời gian cho việc tập luyện cũng như thường xuyên kiểm tra chỉ số sức khỏe của bản thân. Chính vì lẽ đó mà nhu cầu của người dùng dành cho một sản phẩm, giải pháp giúp họ tiết kiệm được thời gian nhưng vẫn đảm bảo được sức khỏe của bản thân là vô cùng cao. Thấu hiểu được nhu cầu đó chúng tôi đã áp dụng và cho ra mắt một sản phẩm công nghệ - ứng dụng hỗ trợ chăm sóc sức khỏe cá nhân Wecare.

Chăm sóc sức khỏe bản thân chưa bao giờ là dễ tuy nhiên nếu như có sản phẩm hỗ trợ quá trình này thì cũng đã góp phần khiến cho trải nghiệm người dùng trở nên tốt hơn.

2. Mục tiêu đề tài

2.1. Lý thuyết

- Nghiên cứu về thư viện Jetpack Compose và ngôn ngữ lập trình Kotlin
- Nắm rõ các thao tác trên môi trường phát triển tích hợp Android studio
- Nghiên cứu và sử dụng dịch vụ cơ sở dữ liệu trên nền tảng đám mây – Firebase
- Nghiên cứu cách thức phát triển ứng dụng Android

2.2. Mục tiêu

- Người dùng có thể kiểm tra số bước chân mình đi được mỗi ngày
- Ứng dụng sẽ hỗ trợ người dùng theo dõi những chỉ số cơ bản như BMI, lượng nước uống và số bước chân
- Ứng dụng có giao diện thân thiện, dễ sử dụng

3. Quá trình hiện thực

Trải qua quá trình tìm hiểu và thực hiện đề tài, nhóm đã giải quyết được các vấn đề đặt ra ban đầu, từng bước giải quyết và liên tục đặt ra những vấn đề mới để ngày càng hoàn thiện sản phẩm. Những bước thực hiện:

- Phân tích yêu cầu bài toán
- Lựa chọn nền tảng công nghệ phù hợp với yêu cầu
- Nghiên cứu cơ sở lý thuyết của công nghệ lựa chọn
- Áp dụng lý thuyết vào xây dựng ứng dụng thực tiễn
- Kiểm tra và khám phá các ứng dụng tương tự để tối ưu hóa trải nghiệm người dùng và cải thiện các tính năng cho ứng dụng

CHƯƠNG II: KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU

1. Phương pháp khảo sát

Khảo sát được thực hiện theo hình thức online với đối tượng chính là các bạn trẻ có độ tuổi từ 18-35.

2. Câu hỏi khảo sát

Câu 1: Bạn thuộc độ tuổi nào?

Câu 2: Giới tính của bạn là gì

Câu 3: Bạn có chú ý đến sức khỏe của mình không?

Câu 4: Bạn có sử dụng các ứng dụng chăm sóc sức khỏe trên điện thoại không? Tần suất sử dụng ứng dụng của bạn?

Câu 5: Theo bạn, những tính năng nào cần thiết cho một ứng dụng chăm sóc sức khỏe?

3. Kết quả khảo sát

Hơn 90% người tham gia khảo sát thuộc độ tuổi từ 18 đến 30, trong đó có 60% là nam và 40% là nữ.

Có 65% người tham gia khảo sát thường xuyên chú ý đến sức khỏe của bản thân trong khi có gần 10% là không theo dõi sức khỏe và số còn lại có tần suất chỉ là thỉnh thoảng.

Có hơn 50% người tham gia khảo sát không sử dụng bất kì ứng dụng theo dõi sức khỏe nào, trong khi đó số còn lại cũng chỉ dùng các ứng dụng theo dõi sức khỏe với tần suất là thỉnh thoảng.

90% người tham gia khảo sát yêu cầu độ chính xác cao về các giá trị đầu ra của ứng dụng, trong khi đó hơn 70% mong muốn rằng sẽ có một ứng dụng hỗ trợ nhiều tính năng về theo dõi sức khỏe. Một số các yêu cầu khác về ứng dụng bao gồm: giao diện cuốn hút, dễ dùng và thời gian truy xuất, xử lý nhanh,...

4. Xác định yêu cầu

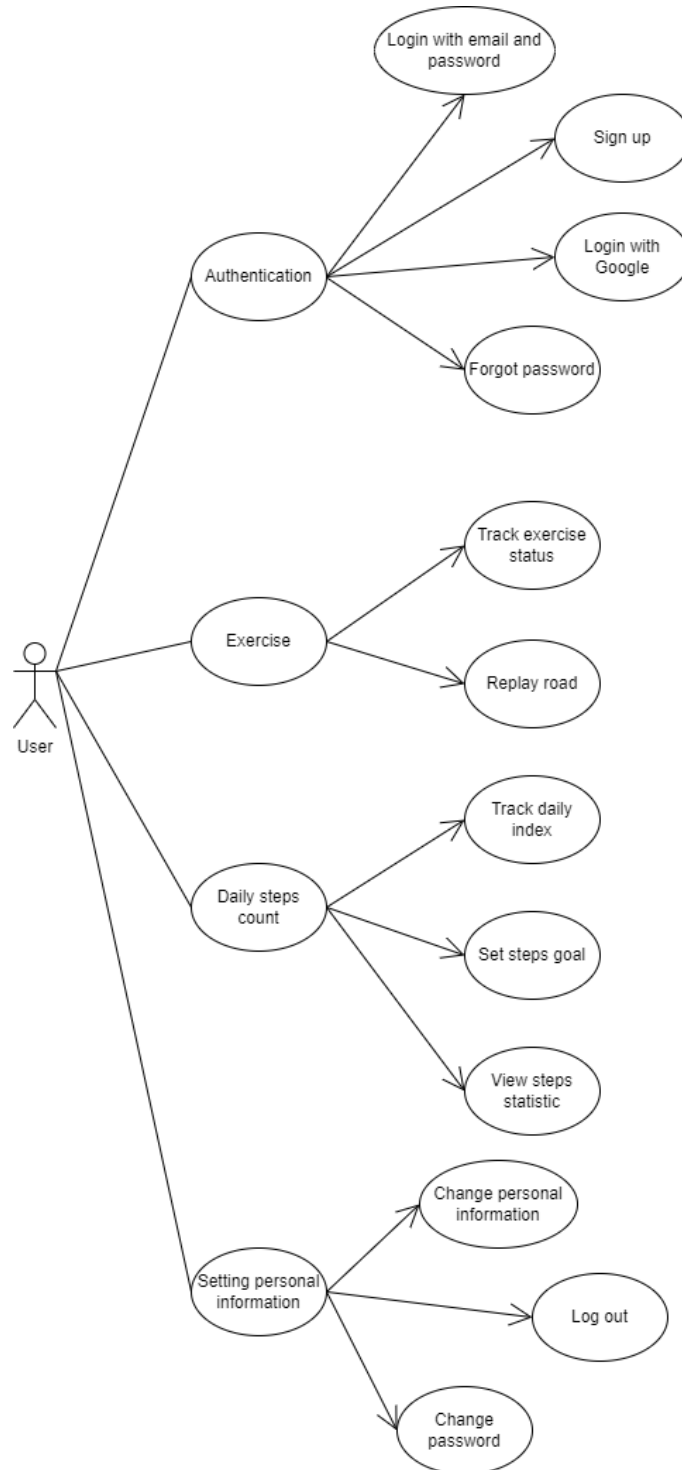
Xây dựng một ứng dụng với độ chính xác cao và hỗ trợ nhiều tính năng giúp người dùng theo dõi sức khỏe mỗi ngày.

Hướng đến xây dựng ứng dụng với tone màu sáng chủ đạo, giao diện thân thiện, tạo sự động lực giúp người dùng truy cập ứng dụng thường xuyên hơn.

Thông báo, nhắc nhở người dùng theo định kỳ hàng giờ để tạo cho người dùng thói quen truy cập ứng dụng những cũng đồng thời không nhắc nhở quá nhiều để gây ra sự khó chịu cho người dùng.

CHƯƠNG III: PHÂN TÍCH YÊU CẦU

1. Sơ đồ Usecase



2. Mô tả use case

STT	Chức năng	Mô tả
1	Đăng ký	Người dùng có thể tạo tài khoản mới bằng email và mật khẩu
2	Đăng nhập bằng email và mật khẩu	Người dùng tiến hành đăng nhập bằng email và mật khẩu đã được đăng ký trước đó
3	Đăng nhập bằng google	Người dùng tiến hành đăng nhập bằng tài khoản google
4	Quên mật khẩu	Người dùng có thể đổi lại mật khẩu bằng email được dùng để đăng ký tài khoản được trước
5	Theo dõi chỉ số khi tập thể dục	Người dùng có thể kiểm tra các chỉ số như quãng đường, khoảng thời gian, số calories đã tiêu hao sau khi hoàn thành bài thể dục
6	Xem lại quãng đường đã đi	Người dùng có thể xem lại video mô phỏng quãng đường đã đi được trong quá trình tập thể dục

7	Theo dõi chỉ số bước chân đi được theo ngày	Cho phép người dùng xem những chỉ số như số bước chân, lượng calo tiêu hao, khoảng thời gian di chuyển
8	Đặt mục tiêu cho bước chân cần đi mỗi ngày	Người dùng có thể đặt mục tiêu số bước chân cần đi hàng ngày
9	Xem thống kê số bước chân	Người dùng có thể xem số bước chân theo mỗi ngày và mỗi ngày theo từng giờ
10	Thay đổi thông tin cá nhân	Người dùng có thể thay đổi những thông tin như cân nặng, chiều cao
11	Thay đổi mật khẩu	Người dùng có thể thay đổi mật khẩu
12	Đăng xuất	Người dùng có thể đăng xuất khỏi phiên đăng nhập hiện tại đồng thời xóa dữ liệu liên quan

3. Đặc tả usecase

3.1. Đăng ký

Use case ID	UC1
--------------------	------------

Tên	Đăng ký
Mô tả tóm tắt	Cho phép người dùng tạo tài khoản để sử dụng nhiều chức năng hơn của ứng dụng
Người thực hiện (Actors)	Khách (Guest)
Sự kiện kích hoạt (Trigger)	Bấm chọn nút Đăng ký ngay
Điều kiện tiên quyết (Pre-conditions)	Email được nhập chưa được đăng ký trước đó
Điều kiện kết quả (Post-conditions)	Tài khoản mới và thông tin người dùng được lưu vào hệ thống
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng bấm chọn nút Đăng ký ngay. 2. Hệ thống hiện bản nhập liệu thông tin đăng ký tài khoản. 3. Nhập thông tin và bấm Đăng ký. 4. Hệ thống kiểm tra tính hợp lệ và tạo tài khoản.

	5. Hệ thống lưu lại thông tin người dùng và điều hướng đến màn hình chính của ứng dụng.
Kịch bản thay thế (Alternative flow)	3a.1. Người dùng nhập thiếu thông tin và bấm Đăng ký 3a.2. Hệ thống hiện thông báo điền thiếu thông tin 4.1. Hệ thống hiện tài khoản đã tồn tại, người dùng phải thay đổi tài khoản đăng ký để tiếp tục

3.2. Đăng nhập bằng email và mật khẩu

Use case ID	UC2
Tên	Đăng nhập bằng email và mật khẩu
Mô tả tóm tắt	Cho phép người dùng đăng nhập vào ứng dụng với tài khoản đã được tạo trước đó.
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Bấm chọn nút Đăng nhập

Điều kiện tiên quyết (Pre-conditions)	Tài khoản và mật khẩu được nhập vào phải hợp lệ.
Điều kiện kết quả (Post-conditions)	Ứng dụng thông báo đăng nhập thành công và điều hướng đến màn hình chính của ứng dụng.
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng nhập thông tin email, mật khẩu và bấm nút Đăng nhập. 2. Hệ thống sẽ kiểm tra tính hợp lệ của tài khoản. 3. Ứng dụng sẽ điều hướng người dùng đến màn hình chính của ứng dụng.
Kịch bản thay thế (Alternative flow)	<p>2a.1. Người dùng nhập sai thông tin đăng nhập.</p> <p>2a.2. Hệ thống thông báo tài khoản không hợp lệ, đăng nhập thất bại.</p>

3.3. Đăng nhập với Google

Use case ID	UC3
Tên	Đăng nhập bằng Google

Mô tả tóm tắt	Cho phép người dùng đăng nhập vào ứng dụng với tài khoản Google.
Người thực hiện (Actors)	Người dùng (User) hoặc Khách (Guest)
Sự kiện kích hoạt (Trigger)	Bấm chọn nút Đăng nhập với Google
Điều kiện tiên quyết (Pre-conditions)	Tài khoản Google phải hợp lệ
Điều kiện kết quả (Post-conditions)	Ứng dụng thông báo đăng nhập thành công và điều hướng đến màn hình chính của ứng dụng.
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng bấm nút Đăng nhập với Google và nhập thông tin tài khoản Google 2. Hệ thống sẽ kiểm tra tính hợp lệ của tài khoản. 3. Ứng dụng sẽ điều hướng người dùng đến màn hình chính của ứng dụng.

Kịch bản thay thế (Alternative flow)	<p>2a.1. Người dùng nhập sai thông tin tài khoản Google.</p> <p>2a.2. Hệ thống thông báo tài khoản không hợp lệ, đăng nhập thất bại.</p>
---	--

3.4. Quên mật khẩu

Use case ID	UC4
Tên	Quên mật khẩu
Mô tả tóm tắt	Cho phép người dùng đổi lại mật khẩu khi quên.
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Bấm chọn nút Quên mật khẩu tại màn hình đăng nhập
Điều kiện tiên quyết (Pre-conditions)	Tài khoản ứng với email đó phải được tạo từ trước.
Điều kiện kết quả (Post-conditions)	Người dùng thay đổi mật khẩu thành công
Kịch bản chính (Basic flow)	<ol style="list-style-type: none">1. Người dùng bấm vào nút Quên mật khẩu.2. Người dùng nhập vào email ứng với tài khoản đã được tạo bởi email.

	3. Người dùng kiểm tra mail và tạo password mới dựa trên đường link được cung cấp.
Kịch bản thay thế (Alternative flow)	

3.5. Theo dõi chỉ số khi tập thể dục

Use case ID	UC5
Tên	Theo dõi chỉ số khi tập thể dục
Mô tả tóm tắt	Cho phép người dùng theo dõi những chỉ số thiết yếu khi đang tập thể dục
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Người dùng tiến hành các exercise khi bấm vào Button Go
Điều kiện tiên quyết (Pre-conditions)	Người dùng phải có tài khoản và phải đăng nhập thành công trước đó

Điều kiện kết quả (Post-conditions)	Người dùng có thể thấy được màn hình tính toán các chỉ số khi đang thực hiện exercise
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng bấm vào nút Go sau khi đã thiết lập mục tiêu. 2. Người dùng có thể thấy các thông tin chỉ số hiển thị trên màn hình. 3. Người dùng bấm icon Nhạc để bật tắt Nhạc. 4. Người dùng bấm Stop để hoàn thành tính toán exercise.
Kịch bản thay thế (Alternative flow)	4.1. Người dùng bấm Pause để tạm dừng tính toán các chỉ số.

3.6. Xem lại quãng đường đã đi trong quá trình tập thể dục

Use case ID	UC6
Tên	Xem lại quãng đường đã đi trong quá trình tập thể dục

Mô tả tóm tắt	Cho phép người dùng xem lại quãng đường đã đi qua trong lần tập gần nhất
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Người dùng hoàn thành một lần thực hiện exercise
Điều kiện tiên quyết (Pre-conditions)	Người dùng phải có tài khoản và phải đăng nhập thành công trước đó
Điều kiện kết quả (Post-conditions)	Người dùng có thể xem được tuyến đường đã đi trong lần thực hiện exercise gần nhất
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng thực hiện exercise và bấm vào Stop để hoàn thành bài tập. 2. Người dùng bấm vào View route. 3. Màn hình View Route hiện ra. Người dùng bấm Play History để bắt đầu Replay
Kịch bản thay thế (Alternative flow)	

3.7. Theo dõi chỉ số bước chân đi được theo ngày

Use case ID	UC7
Tên	Theo dõi chỉ số bước chân đi được theo ngày
Mô tả tóm tắt	Cho phép người dùng theo dõi chỉ số bước chân đi được theo ngày
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Người dùng bước đi
Điều kiện tiên quyết (Pre-conditions)	Điện thoại của người dùng có cảm biến vận động Người dùng cho phép truy cập vào cảm biến
Điều kiện kết quả (Post-conditions)	Số bước chân thay đổi khi người dùng di chuyển

Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng truy cập vào app 2. Người dùng cho phép truy cập vào nhận diện hoạt động 3. Người dùng di chuyển
Kịch bản thay thế (Alternative flow)	

3.8. Đặt mục tiêu cho cho bước chân cần đi mỗi ngày

Use case ID	UC8
Tên	Đặt mục tiêu cho số bước chân cần đi mỗi ngày
Mô tả tóm tắt	Cho phép người dùng đặt mục tiêu cho số bước chân muốn đi hàng ngày
Người thực hiện (Actors)	Người dùng (User)

Sự kiện kích hoạt (Trigger)	Người dùng vào màn hình đếm số bước chân và click chọn đặt mục tiêu
Điều kiện tiên quyết (Pre-conditions)	Không
Điều kiện kết quả (Post-conditions)	Mục tiêu số bước chân của người dùng thay đổi theo mong muốn
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 4. Người dùng truy cập vào màn hình đếm số bước chân 5. Người dùng chọn đặt mục tiêu 6. Người dùng chọn số bước chân mong muốn
Kịch bản thay thế (Alternative flow)	

3.9. Xem thống kê số bước chân

Use case ID	UC9
Tên	Xem thống kê số bước chân
Mô tả tóm tắt	Cho phép người dùng xem thống kê số bước chân theo ngày và giờ
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Người dùng truy cập màn hình đếm số bước chân
Điều kiện tiên quyết (Pre-conditions)	Người dùng đã đăng nhập và có di chuyển trước đó
Điều kiện kết quả (Post-conditions)	Người dùng xem được số bước chân số bước chân theo ngày và giờ

Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng truy cập vào app 2. Người dùng vào màn hình đếm số bước chân
Kịch bản thay thế (Alternative flow)	

3.10. Thay đổi thông tin cá nhân

Use case ID	UC10
Tên	Thay đổi thông tin cá nhân
Mô tả tóm tắt	Cho phép người dùng thay đổi những thông tin cá nhân như chiều cao và cân nặng
Người thực hiện (Actors)	Người dùng (User)

Sự kiện kích hoạt (Trigger)	Người dùng truy cập vào màn hình tài khoản và chọn “thay đổi thông tin”
Điều kiện tiên quyết (Pre-conditions)	Người dùng đăng nhập
Điều kiện kết quả (Post-conditions)	Thông tin được thay đổi như người dùng mong muốn
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng truy cập vào app 2. Người dùng vào màn hình account 3. Người dùng chọn “Thay đổi thông tin”
Kịch bản thay thế (Alternative flow)	

3.11. Thay đổi mật khẩu

Use case ID	UC11
--------------------	-------------

Tên	Thay đổi mật khẩu
Mô tả tóm tắt	Cho phép người dùng thay đổi mật khẩu
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Người dùng chọn thay đổi mật khẩu ở màn hình account
Điều kiện tiên quyết (Pre-conditions)	Người dùng đã đăng nhập
Điều kiện kết quả (Post-conditions)	Người dùng đổi mật khẩu thành công
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 1. Người dùng truy cập vào app 2. Người dùng truy cập vào màn hình account 3. Người dùng chọn “Thay đổi mật khẩu”

Kịch bản thay thế (Alternative flow)	
---	--

3.12. Đăng xuất

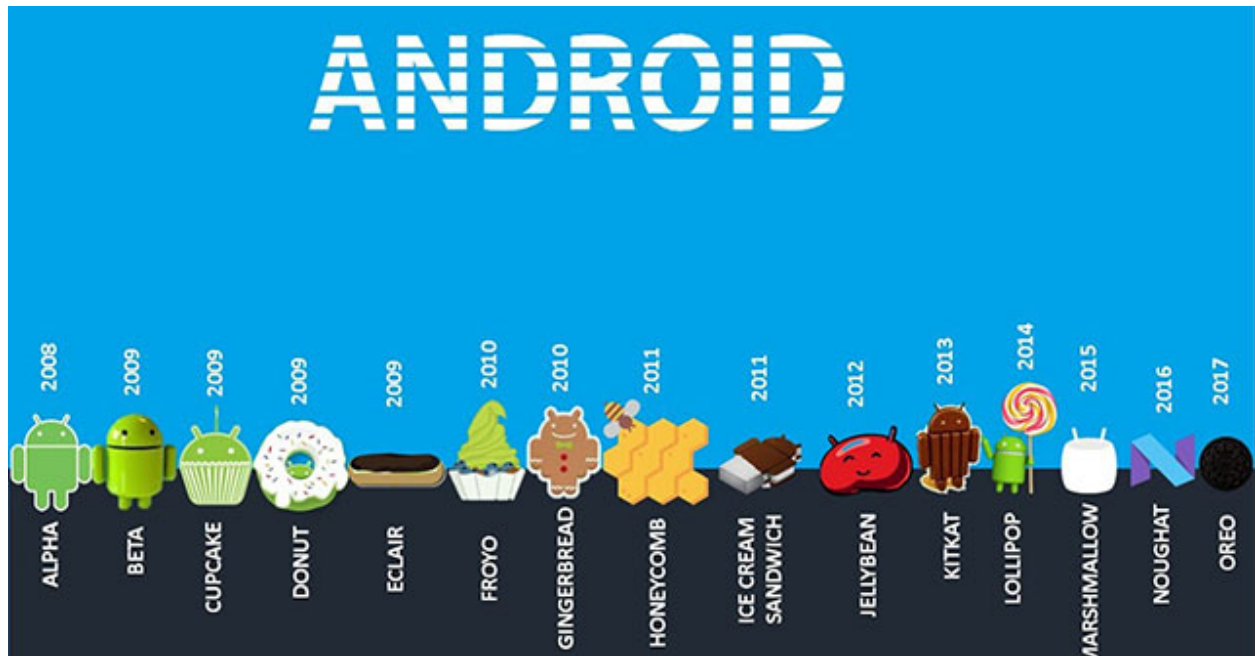
Use case ID	UC12
Tên	Đăng xuất
Mô tả tóm tắt	Cho phép người dùng đăng xuất khỏi phiên đăng nhập
Người thực hiện (Actors)	Người dùng (User)
Sự kiện kích hoạt (Trigger)	Người dùng chọn đăng xuất ở màn hình account
Điều kiện tiên quyết (Pre-conditions)	Người dùng đã đăng nhập

Điều kiện kết quả (Post-conditions)	Người dùng đăng xuất khỏi phiên đăng nhập
Kịch bản chính (Basic flow)	<ol style="list-style-type: none"> 4. Người dùng truy cập vào app 5. Người dùng truy cập vào màn hình account 6. Người dùng chọn “Đăng xuất”
Kịch bản thay thế (Alternative flow)	

CHƯƠNG IV: CƠ SỞ LÝ THUYẾT

1. Android

1.1. Tổng quan và lịch sử



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 Android Inc. với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

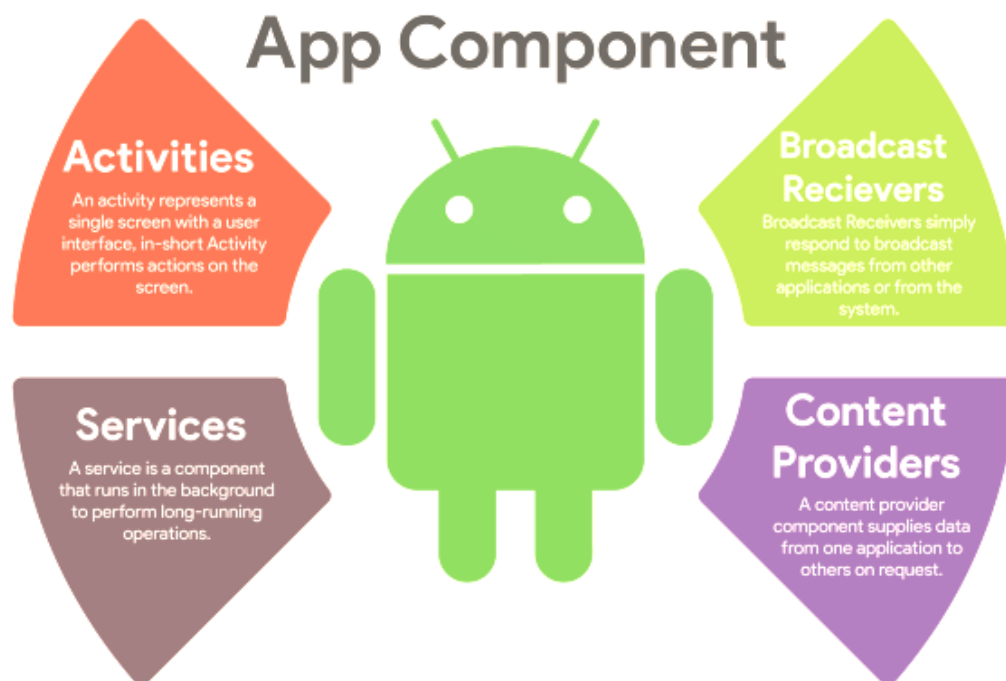
Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cảm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào năm 2008.

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. Tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

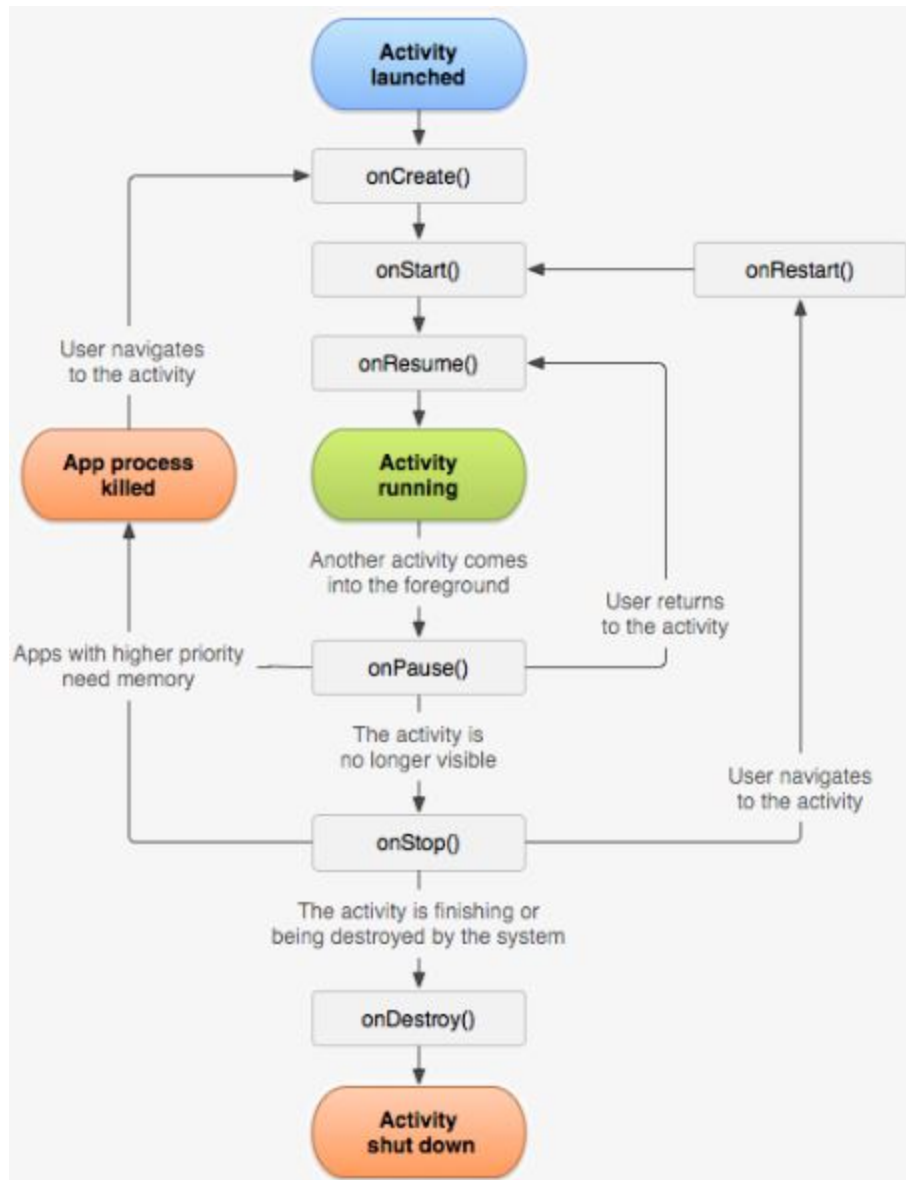
Android chiếm 87,7% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 2 năm 2017, với tổng cộng 2 tỷ thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong cái gọi là "cuộc chiến điện thoại thông minh" giữa các công ty công nghệ.

1.2. Các thành phần cơ bản của Android App



- Activity

- Giao diện người dùng của ứng dụng được xây dựng dựa trên một hoặc nhiều Activity. Activity sử dụng Fragment và View để bố trí và hiển thị thông tin và tương tác với hoạt động của người dùng.
- Activity là một lớp được có sẵn trong Android và tất cả các ứng dụng trong đó có giao diện người dùng phải kế thừa nó để tạo ra cửa sổ.
- Chẳng hạn, lớp MainActivity là một Activity nó mở rộng từ một Activity có sẵn trong thư viện Android.
- Một vòng đời của Activity mô tả các quá trình của một Activity kể từ khi nó bắt đầu chạy (Launched), cho tới khi ứng dụng bị tắt. Bao gồm cả quá trình Activity bị dừng tạm thời, chạy lại (resume)



- Service

- Service có chức năng giúp ứng dụng vẫn chạy được, nhưng không cần hiển thị trên giao diện. Ví dụ các bạn dùng các ứng dụng nghe nhạc, mặc dù các bạn tắt ứng dụng rồi nhưng vẫn nghe được nhạc (đó là vì nó đang chạy dưới nền/background).
- Chúng ta có thể liên kết/ kết nối giữa một Activity với một service. Ví dụ: khi download một file từ trên mạng, việc download thực hiện ở service. Sau

đó sẽ trả kết quả phần trăm download lên activity để hiển thị cho người dùng biết.

- Chú ý: Mặc dù service chạy ở chế độ background nhưng cần phân biệt giữa service và thread. Service không phải thread, do đó tùy trường hợp mà chúng ta sử dụng và xử lý cho phù hợp.
- Khi sử dụng service chúng ta sẽ kế thừa từ lớp cha là: Service.
- Broadcast Receivers
 - Broadcast receiver được sử dụng trong nhiều trường hợp. Ví dụ: chúng ta có thể chuyển dữ liệu từ service lên activity (ngoài sử dụng binding) chúng ta có thể sử dụng broadcast để gửi dữ liệu. Hoặc trong các ứng dụng như hẹn giờ, khi đến giờ hẹn, ứng dụng sẽ sử dụng broadcast báo thức, tạo ra notification trên màn hình để báo cho người dùng biết.
 - Khi sử dụng broadcast receiver chúng ta kế thừa từ BroadcastReceiver.
- Content Providers
 - Một thành phần Content Provider cung cấp dữ liệu từ một ứng dụng tới các ứng dụng khác theo yêu cầu.
 - Những yêu cầu này được xử lý bởi các phương thức của lớp ContentResolver.
 - Một Content Provider có thể sử dụng các cách khác nhau để lưu giữ dữ liệu và dữ liệu có thể được lưu giữ trong một cơ sở dữ liệu, một file, hoặc có thể thông qua một mạng...

2. Ngôn ngữ lập trình Kotlin



Kotlin là một ngôn ngữ lập trình kiểu tĩnh chạy trên máy ảo Java (JVM) và có thể được biên dịch sang mã nguồn Java hay sử dụng cơ sở hạ tầng trình biên dịch LLVM. Nó được tài trợ và phát triển bởi JetBrains. Mặc dù cú pháp không tương thích với Java, những bản thực hiện JVM của thư viện chuẩn Kotlin được thiết kế để tương tác với mã Java và dựa vào mã Java từ Java Class Library có sẵn, ví dụ như collections framework. Kotlin sử dụng suy luận kiểu một cách tích cực để xác định kiểu của giá trị và biểu thức vốn không được nêu rõ. Điều này giúp giảm tính dài dòng của ngôn ngữ so với Java, vốn thường đòi hỏi toàn bộ đặc kiểu một cách dư thừa mãi đến phiên bản 10. Mã Kotlin có thể chạy trên JVM đến phiên bản Java 11 mới nhất.

Kể từ Android Studio 3.0 (phát hành vào tháng 10 năm 2017), Kotlin được Google hỗ trợ đầy đủ để sử dụng cho việc lập trình ứng dụng cho hệ điều hành Android của họ, và được nhúng trực tiếp vào trong gói cài đặt của IDE đó để thay thế cho trình biên dịch Java tiêu chuẩn. Trình biên dịch Android Kotlin cho phép người dùng chọn lựa giữa hướng đến mã bytecode tương thích với Java 6, hay Java 8.

2.1. Lịch sử hình thành

- Vào tháng 7 năm 2011, JetBrains đã giới thiệu Project Kotlin, là một ngôn ngữ mới cho JVM đã được phát triển trong một năm. Một trong những mục tiêu được nhắc đến của Kotlin là biên dịch nhanh như Java. Vào tháng 2 năm 2012, JetBrains mở nguồn dự án theo giấy phép Apache 2
- Kotlin v1.0 được phát hành vào ngày 15 tháng 2 năm 2016. Đây được coi là phiên bản chính thức ổn định đầu tiên và JetBrains đã cam kết tương thích ngược dài hạn kể từ phiên bản này.
- Tại hội nghị Google I/O 2017, Google công bố hỗ trợ hạng nhất (first-class support) cho Kotlin trên Android.
- Kotlin v1.2 được phát hành vào ngày 28 tháng 11 năm 2017. Tính năng chia sẻ mã nguồn giữa nền tảng JVM và Javascript mới được thêm vào bản phát hành này.

- Kotlin v1.3 được phát hành vào ngày 29 tháng 10 năm 2018, với các coroutine API cho lập trình bất đồng bộ.

3. Hệ quản trị cơ sở dữ liệu Firebase



3.1. Firebase là gì?

Firebase là một nền tảng mà Google cung cấp để phát triển các ứng dụng di động và trang web, hỗ trợ Developer trong việc đơn giản hóa các thao tác với dữ liệu và không cần tác động tới backend hay server

3.2. Firebase Realtime database

Đầu tiên, để sử dụng nền tảng này tạo ứng dụng, lập trình viên cần đăng ký tài khoản Firebase. Đăng ký xong, bạn sẽ sở hữu một Realtime Database có dạng JSON được đồng bộ thời gian đến tất cả kết nối client Dữ liệu trong các database được tự động cập nhật liên tục khi lập trình viên tiến hành phát triển ứng dụng. Sau khi được cập nhật, những dữ liệu này sẽ được truyền tải thông qua kết nối SSL có chứng nhận 2048 bit Tại những ứng dụng đa nền tảng toàn bộ máy khách đều dùng chung một database. Trong trường hợp kết nối Internet bị gián đoạn, các dữ liệu sẽ được lưu trữ trên local và cập nhật khi đường truyền ổn định

3.3. Firebase Authentication

Hoạt động kế tiếp của Firebase là tạo lập quy trình xác thực người dùng thông qua các phương tiện như Google, Email, GitHub, Facebook, Twitter và xác thực nặc danh đối với các ứng dụng. Việc xác thực giúp bảo vệ an toàn dữ liệu về thông tin cá nhân của người dùng mạng

3.4. Firebase Hosting

Các thức hoạt động tiêu biểu cuối cùng, không thể không nhắc đến của Firebase là cung cấp hosting. Firebase hosting thông qua sự hỗ trợ của SSL đến từ mạng CDN. Nhờ vậy mà lập trình viên có thể tiết kiệm rất nhiều thời gian thiết kế cũng như phát triển ứng dụng.

3.5. Ưu điểm của Firebase

- **Tốc độ cao:** Firebase hỗ trợ phát triển ứng dụng với tốc độ nhanh chóng, giảm bớt thời gian phát triển và tiếp thị ứng dụng
- **Đơn giản, dễ dàng trong sử dụng:** Người dùng có thể tạo tài khoản Firebase bằng tài khoản Google và cũng như sử dụng nền tảng này trong phát triển ứng dụng một cách dễ dàng
- **Một nền tảng, đa dịch vụ:** Firebase cung cấp đầy đủ các dịch vụ hỗ trợ phát triển web, bạn có thể chọn database Firestore hoặc Realtime theo ý muốn
- **Là một phần của Google:** Firebase đã được mua lại và hiện tại đang là một phần của Google. Nó khai thác triệt để sức mạnh và các dịch vụ sẵn có của Google
- **Tập trung phát triển chủ yếu về giao diện người dùng:** Firebase cho phép lập trình viên tập trung chủ yếu vào phát triển phần giao diện người dùng nhờ kho Backend mẫu phong phú đa dạng
- **Không có máy chủ:** Điều này giúp Firebase có khả năng tối ưu hóa về hiệu suất làm việc thông qua mở rộng cụm database

- Học máy: Firebase cung cấp học máy cho lập trình viên, hỗ trợ tối đa cho việc phát triển ứng dụng
- Tạo ra lưu lượng truy cập: Với khả năng hỗ trợ tạo lập các chỉ mục, Firebase giúp nâng cao thứ hạng của ứng dụng trên bảng xếp hạng Google, từ đó tăng lượt traffic
- Theo dõi lỗi: Firebase là công cụ phát triển, khắc phục lỗi tuyệt vời
- Sao lưu: Firebase sao lưu thường xuyên, giúp đảm bảo tính sẵn có và bảo mật thông tin cho dữ liệu

3.6. Nhược điểm của Firebase

- Không phải mã nguồn mở - Chỉ hoạt động trên CSDL NoSQL
- Lập trình viên không được truy cập vào mã nguồn
- Truy vấn chậm
- Phạm vi hoạt động bị hạn chế ở một vài quốc gia
- Một vài dịch vụ cần trả phí, chi phí dành cho Firebase tương đối lớn và không ổn định
- Thiếu Dedicated Servers và cả hợp đồng doanh nghiệp

CHƯƠNG V: XÂY DỰNG HỆ THỐNG

1. Xây dựng kiến trúc hệ thống

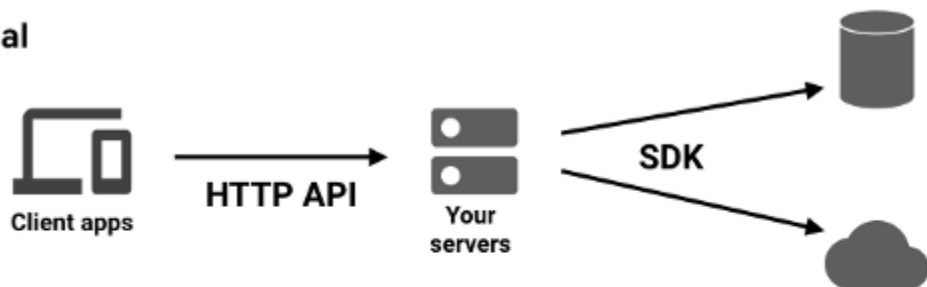
1.1. Công nghệ sử dụng

- Ngôn ngữ: Kotlin
- UI Framework: Jetpack Compose, XAML
- UI Design tool: Figma
- Database: Firebase firestore, Room
- IDE: Android Studio
- Kiến trúc ứng dụng: MVVM

1.2. Kiến trúc hệ thống: Mô hình Client-Server

- **Lớp Client:** Hiển thị giao diện người dùng, thông qua kết nối Internet để kết nối đến cơ sở dữ liệu quản lý thư viện. Là nơi diễn ra các tương tác của người dùng với chương trình như nhập, các thao tác truy xuất dữ liệu.
- **Lớp Server:** Quản lý và chứa toàn bộ dữ liệu của phần mềm. Đồng thời xử lý các yêu cầu nhập/xuất được gửi xuống từ lớp Client. Ứng dụng được dùng làm lớp Server là Firebase.

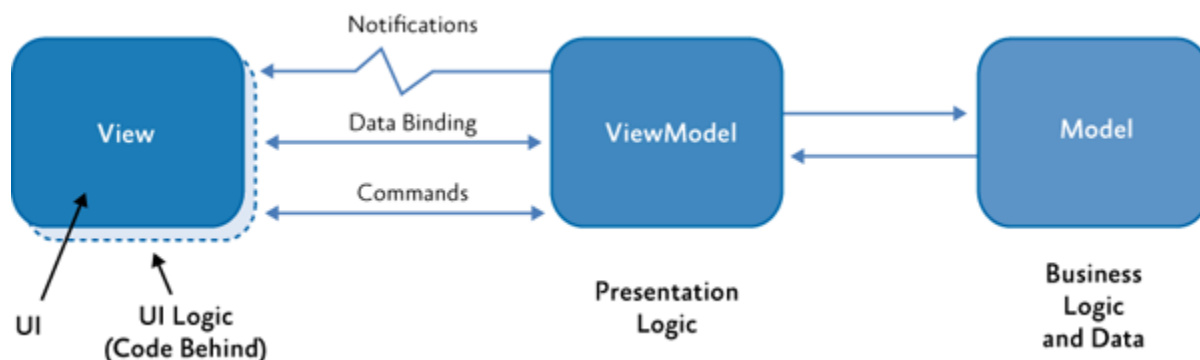
Traditional



Firebase



1.3. Kiến trúc ứng dụng: Mô hình MVVM



MVVM (viết tắt của Model-View-ViewModel) là một mẫu kiến trúc thiết kế phần mềm dựa trên việc tách quá trình phát triển GUI (view) ra khỏi quá trình phát triển các xử lý logic và backend, sao cho phần view không bị phụ thuộc vào bất kì nền tảng model cụ thể nào. Trong MVVM, các tầng bên dưới sẽ không biết được thông tin của các tầng trên. MVVM bao gồm 3 phần:

- View: phần giao diện của ứng dụng dùng hiển thị dữ liệu và nhận tương tác người dùng.
- Model: là đối tượng giúp truy xuất và thao tác trên database.
- View Model: là lớp trung gian giữa View và Model.

Mục đích lựa chọn:

- Tận dụng cơ chế data binding: Tối ưu việc phân tách lớp view khỏi phần còn lại của hệ thống, bằng việc loại bỏ hầu như toàn bộ các đoạn code-behind ở lớp view. Điều này giúp cho các lập trình viên UI, UX tập trung vào việc thiết kế giao diện mà không phải bận tâm đến việc xử lý logic. Các lớp của phần mềm sẽ được phát triển nhanh hơn vì sự phân tách luồng công việc rõ ràng, tăng hiệu suất. Ngay cả khi một lập trình viên phải làm việc với toàn bộ code thì họ vẫn sẽ dễ đọc và chỉnh sửa code, hơn nữa UI thường sẽ thay đổi xuyên suốt quá trình phát triển dựa vào

feedback từ khách hàng nên đây sẽ là một lựa chọn ổn định. Ngoài ra kiểm tra, bảo trì và mở rộng khá dễ dàng.

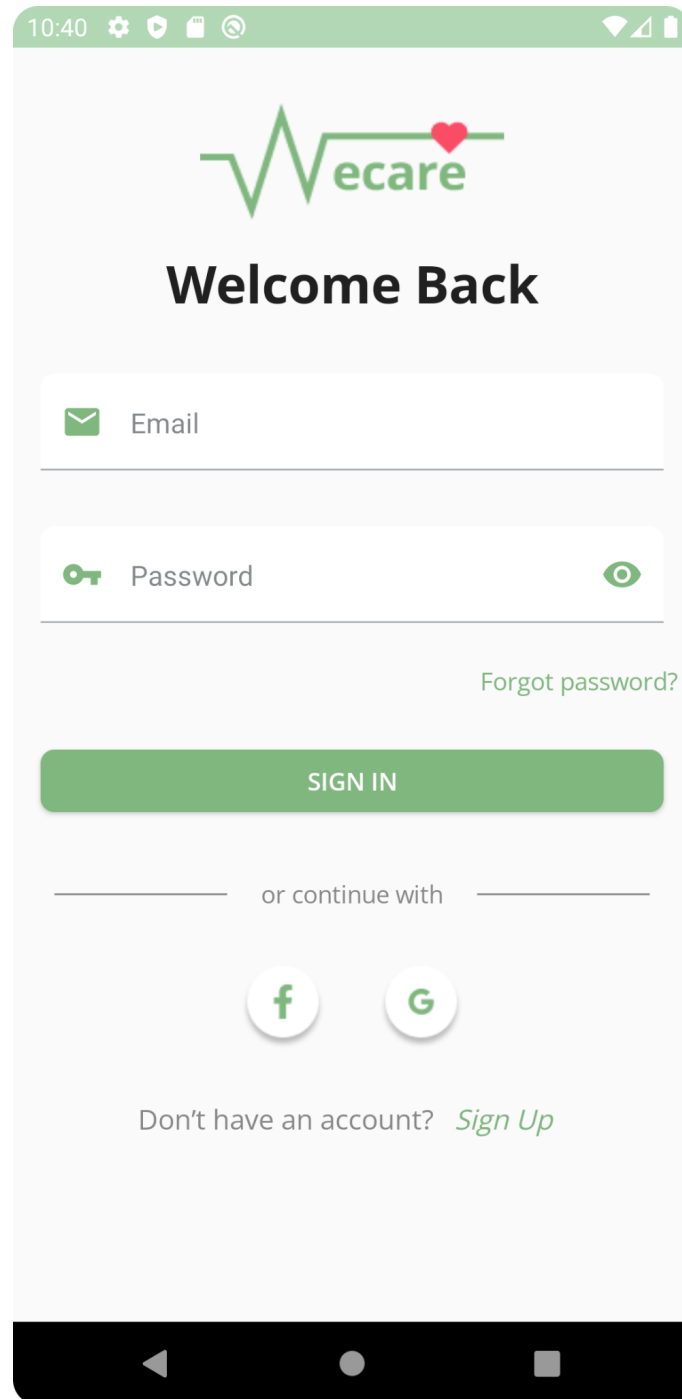
- Thực hiện Unit testing dễ dàng, vì nhà phát triển thực sự không phụ thuộc vào view
- MVVM sẽ tạo sự tương tác hiệu quả giữa designer và developer
- Tăng khả năng sử dụng lại các thành phần hay việc thay đổi giao diện chương trình mà không cần phải viết lại code quá nhiều
- Phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì...

1.4. Mô tả thành phần hệ thống

STT	Thành phần	Mô tả
1	Repository	Chứa các service kết nối với BE được cung cấp Firebase SDK
2	Models	Các class đối tượng thuộc về ứng dụng
3	Views	Chứa giao diện các màn hình
4	Widgets	Chứa các custom element
5	Components	Chứa các group các element khác nhau
6	ViewModels	Chứa các file ViewModel của từng màn hình khác nhau
7	Constants	Chứa các file định dạng các hằng số được liên tục tái sử dụng trong ứng dụng ví dụ như chỉ số dimensions, colors, strings, ...
8	Drawable	Chứa các file hình ảnh, icon, gif của ứng dụng

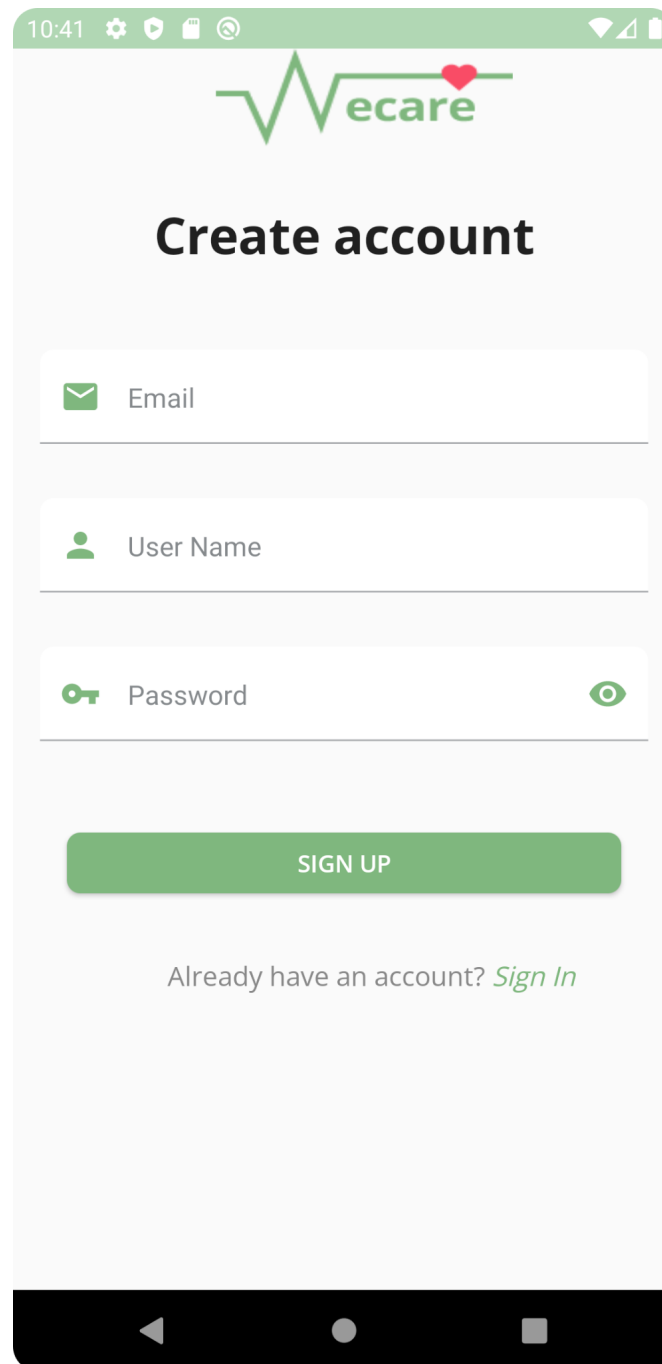
2. Mô tả các màn hình

2.1. Màn hình đăng nhập




STT	Sự kiện	Xử lý
1	Chọn Icon Eyes tại text field Password	Ẩn/hiện password
2	Chọn Forgot password?	Di chuyển qua màn hình Forgot password?
3	Chọn Sign In	Xác thực tài khoản và di chuyển đến màn hình Home
4	Chọn icon Facebook	Tiến hành đăng nhập vào ứng dụng với tài khoản Facebook
5	Chọn icon Google	Tiến hành đăng nhập vào ứng dụng với tài khoản Google
6	Chọn vào chữ Sign up	Di chuyển qua màn hình đăng ký tài khoản

2.2. Màn hình đăng ký





The screenshot shows a mobile application interface for creating an account. At the top, there is a status bar with the time 10:41 and various icons. Below the status bar is the 'Vecare' logo, which consists of a green heart icon and the word 'Vecare' in green. The main heading is 'Create account' in bold black text. There are three input fields: 'Email' with a green envelope icon, 'User Name' with a green person icon, and 'Password' with a green key icon and a green eye icon for toggling visibility. A green 'SIGN UP' button is located below the input fields. At the bottom, there is a link that says 'Already have an account? *Sign In*'. The bottom of the screen shows the Android navigation bar with back, home, and recent apps buttons.



10:41



Create account

 Email

 User Name

 Password 

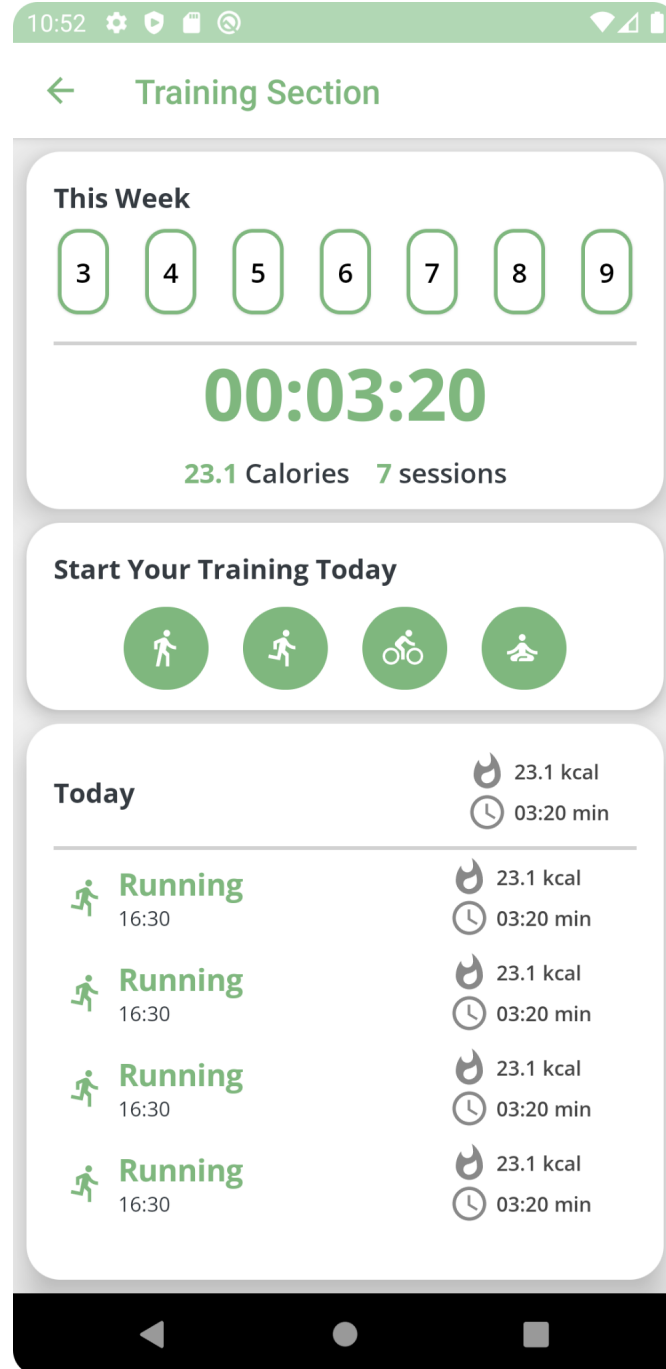
SIGN UP

Already have an account? *Sign In*

STT	Sự kiện	Xử lý
1	Chọn Icon Eyes tại text field Password	Ẩn/hiện password
2	Chọn Sign up	Xác thực tài khoản và tiến hành đăng ký tài khoản
3	Chọn Sign In	Di chuyển đến màn hình Sign In

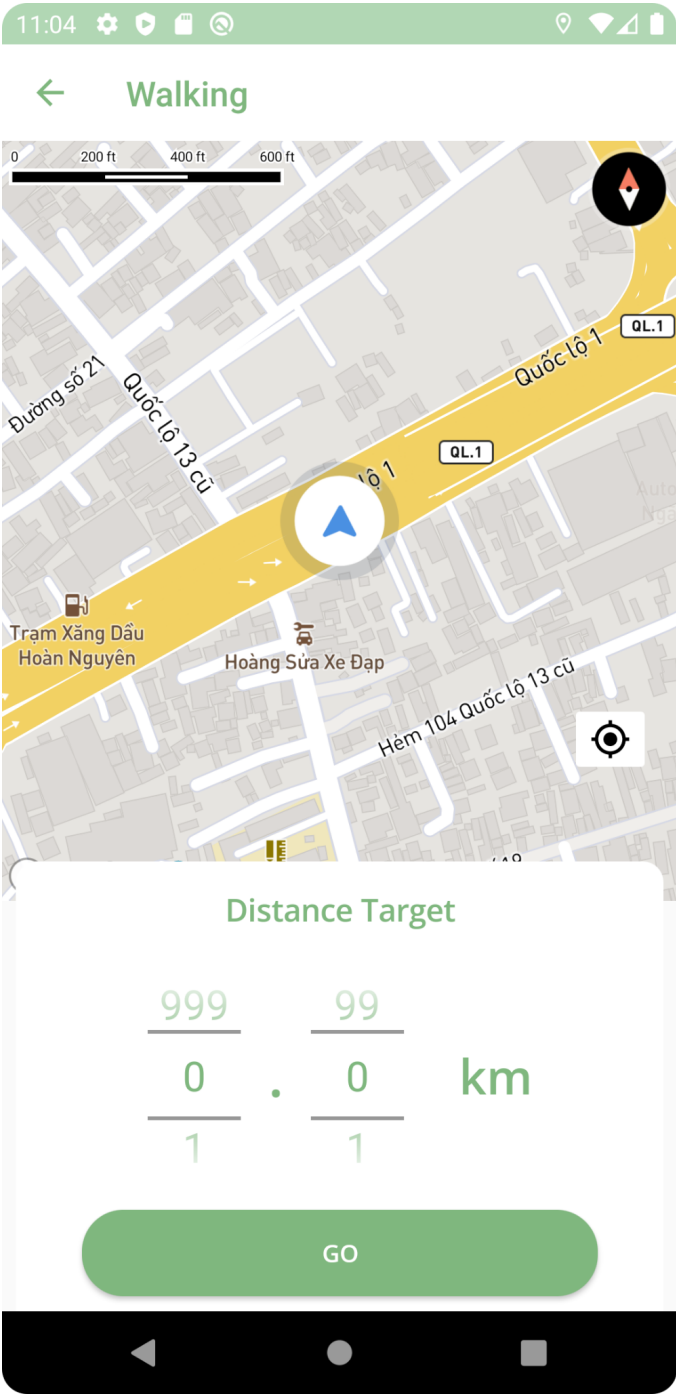
2.3. Màn hình Home

2.4. Màn hình Exercise Landing Page



STT	Sự kiện	Xử lý
1	Chọn Icon quay lại	Quay lại màn hình Home
2	Chọn Icon đi bộ	Di chuyển đến màn hình chuẩn bị đi bộ
3	Chọn Icon chạy bộ	Di chuyển đến màn hình chuẩn bị chạy bộ
4	Chọn Icon đạp xe	Di chuyển đến màn hình chuẩn bị đạp xe
5	Chọn Icon ngồi thiền	Di chuyển đến màn hình đang ngồi thiền

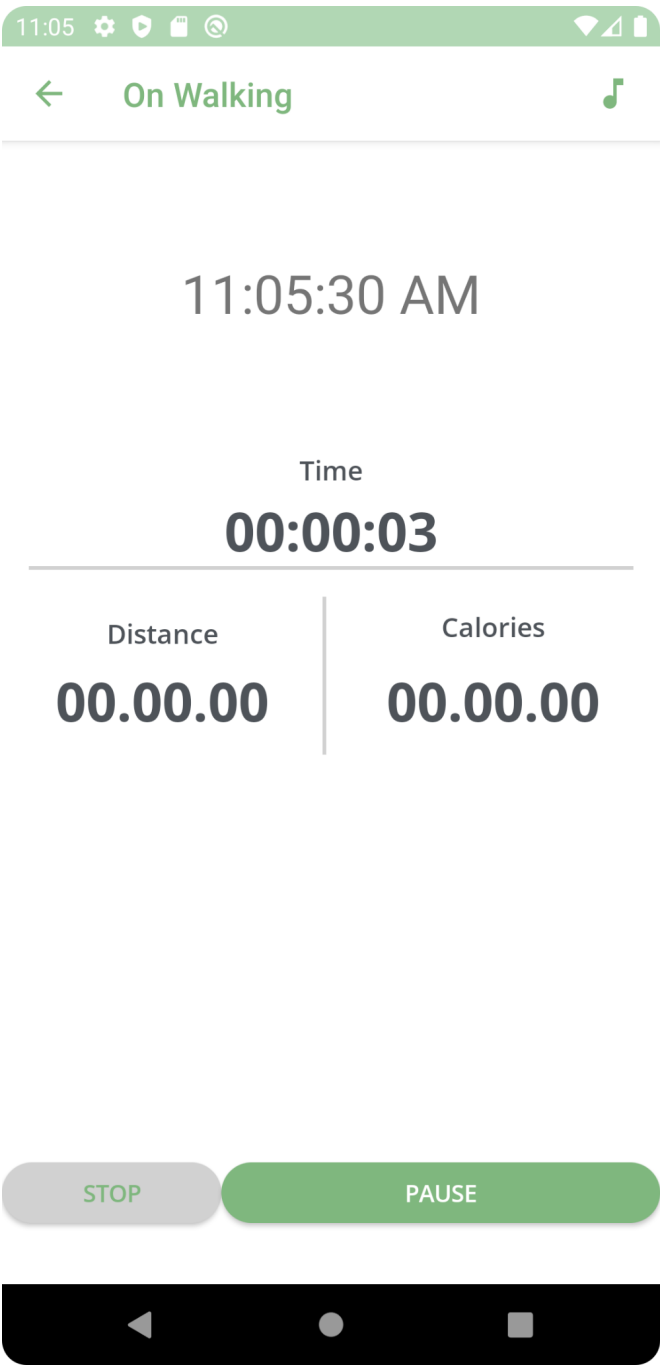
2.5. Màn hình chuẩn bị chạy bộ



STT	Sự kiện	Xử lý
-----	---------	-------

1	Chọn Icon quay lại	Quay lại màn hình Exercise Landing Page
2	Vuốt chọn Spinner	Chọn số km target cho lần đi bộ
3	Chọn Button Go	Di chuyển đến màn hình đang đi bộ

2.6. Màn hình đang đi bộ



STT	Sự kiện	Xử lý
-----	---------	-------

1	Chọn Icon quay lại	Quay lại màn hình Exercise Landing Page
2	Chọn Icon Nhạc	Bật tắt nhạc lúc đang đi bộ
3	Chọn Button Stop	Mở hộp thoại Confirm và di chuyển đến màn hình hoàn thành đi bộ
4	Chọn Button Pause	Tạm thời dừng tính các chỉ số thống kê

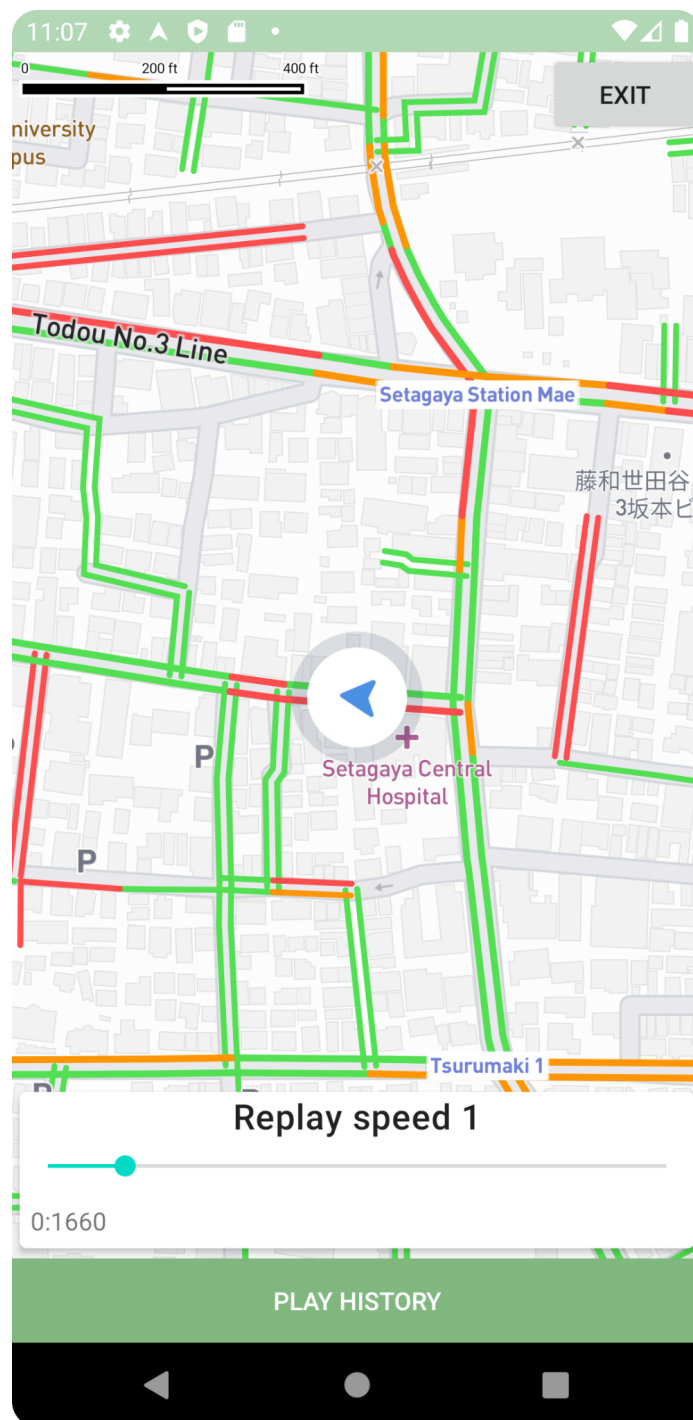
2.7. Màn hình hoàn thành Exercise



STT	Sự kiện	Xử lý
-----	---------	-------

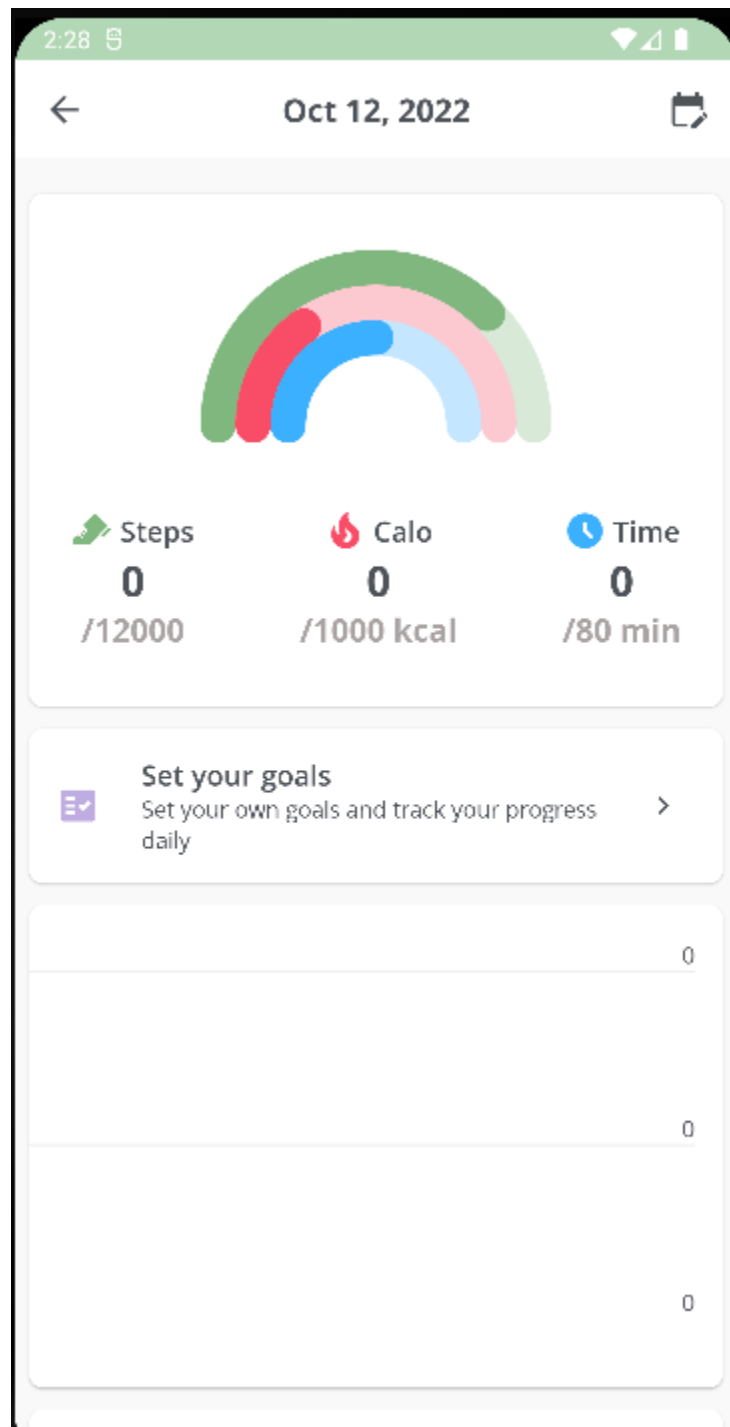
1	Chọn Button Exit	Quay lại màn hình Exercise Landing Page
2	Chọn Button View Route	Di chuyển đến màn hình Replay quãng đường đã đi

2.8. Màn hình Replay quãng đường đã đi



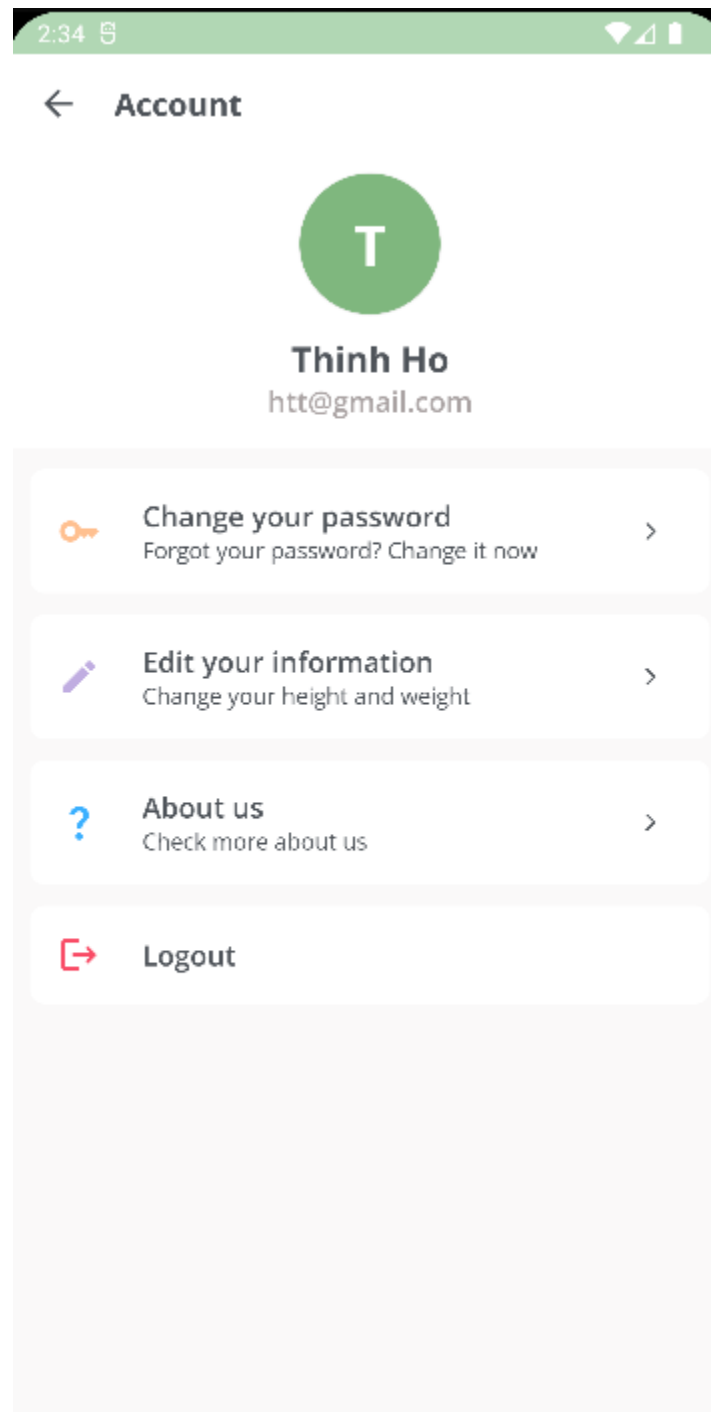
STT	Sự kiện	Xử lý
1	Chọn Button Exit	Quay lại màn hình Exercise Landing Page
2	Chọn Button Play History	Tiến hành Replay quãng đường đã đi
3	Kéo chọn Speed	Tăng giảm tốc độ Replay

2.9. Màn hình đếm và thống kê số bước chân



STT	Sự kiện	Xử lý
1	Chọn Button back	Quay lại màn hình Home
2	Chọn Button edit calendar	Mở dialog cho phép người dùng chọn ngày
3	Chọn button “Set your goals”	Mở bottom sheet cho phép người dùng đặt mục tiêu cho số bước chân

2.10. Màn hình quản lý tài khoản



STT	Sự kiện	Xử lý
1	Chọn Button back	Quay lại màn hình Home
2	Chọn “Change your password”	Mở dialog cho phép người dùng thay đổi mật khẩu
3	Chọn “Edit your information”	Mở dialog cho phép người dùng thay đổi chỉ số chiều cao và cân nặng
4	Chọn “About us”	Show dialog hiển thị thông tin liên hệ và thông tin về ứng dụng
5	Chọn “Logout”	Đăng xuất người dùng khỏi phiên đăng nhập

CHƯƠNG VI: KẾT LUẬN

1. Kết quả đạt được

- Tạo được ứng dụng có tính thực tiễn, áp dụng cao.
- Áp dụng được các kiến thức về quy trình phát triển một phần mềm hướng đối tượng một cách chuyên nghiệp.
- Tìm hiểu thêm về các công nghệ kiến trúc kỹ thuật mới. Đặc biệt là biết cách làm việc với UI Framework Jetpack Compose và MapBox API.
- Hiểu hơn về IDE Android Studio. Biết cách fix bug và detect root cause bằng Logcat.
- Hiểu và nắm được các nguyên lý hoạt động, cách xây dựng 1 ứng dụng native Android.
- Có thêm các kiến thức chuyên sâu hơn về Android. Ví dụ: Android Permission, Motion Sensor, ...
- Thuần thục hơn trong việc sử dụng ngôn ngữ Kotlin.
- Có thêm nhiều góc nhìn về 1 project Android như phân chia module, Android Architecture, micro front-end, tại sao một ứng dụng Android Kotlin Native lại có performance cao hơn so với các ứng dụng cross-platform, ...
- Phát triển nhiều hơn trong việc phân chia công việc, quản lý công việc, quản lý tiến độ, ...

2. Ưu điểm

- Giao diện thân thiện, đẹp mắt.
- Dễ hiểu, dễ dùng, dễ sử dụng.
- Có nhiều chức năng khác nhau đáp ứng được nhiều mục đích của người dùng. Các chức năng đều có tính ứng dụng cao, thân thuộc và cần thiết trong đời sống hàng ngày của mỗi người.
- Có áp dụng những công nghệ tiên về native Android, đáp ứng được yêu cầu của người dùng.

- Hoàn toàn miễn phí

3. Nhược điểm và giải pháp

- Màn hình Home đang giữ entry point cho các màn hình khác → tăng áp lực giao diện lên màn hình Home, giảm khả năng scalability cho ứng dụng.
 - Giải pháp: Migrate qua một cách Navigation khác linh hoạt hơn như sử dụng bottom navigation bar hoặc navigation drawer
- Còn ít các thống kê chỉ số hiển thị khi đang tập. Ví dụ: tốc độ, nhịp tim hiện tại của người dùng, ...
- Chưa có chức năng Pause/Resume tự động việc tính toán các chỉ số khi đang thực hiện các exercise.
 - Giải pháp: Tìm hiểu giải pháp để có thể detect được khi nào nên Pause hoặc Resume tính toán.
- Chức năng báo cáo chưa có phần filter, tổng hợp những thông tin cần thiết liên quan đến việc nâng cao suy nghĩ của người dùng về sức khỏe.
 - Giải pháp: tìm hiểu, khảo sát và phân tích kỹ hơn nhu cầu xem báo cáo của người dùng nhằm tạo ra một chức năng báo cáo có tính ứng dụng cao hơn.
- Chưa có phần chức năng cài đặt - người dùng chưa có khả năng personalize hóa ứng dụng.
 - Giải pháp: tìm ra các solution giúp người dùng có khả năng custom giao diện, cài đặt → tăng tính personal của người dùng.
- Các chức năng có độ hoàn thiện chưa cao. 1 vài chức năng quan trọng chưa thể đáp ứng hết tất cả các nhu cầu của người dùng
 - Giải pháp: nâng cấp các chức năng, tạo ra nhiều chức năng nhỏ hơn để người dùng có thể thoải mái sử dụng. Tăng tính ứng dụng của các chức năng.
- Luồng xử lý dữ liệu còn rườm rà, khó hiểu, khó maintain. Vì lần đầu tìm hiểu cũng như sử dụng một UI Framework mới khiến cho cách xử lý luồng dữ liệu còn rườm rà. Một số chưa áp dụng đúng với kiến trúc ứng dụng đã đề ra trước đó.

- Giải pháp: tìm hiểu nhiều hơn về kiến trúc ứng dụng. Thử nghiệm các example, các project mẫu khác nhau để có thể áp dụng vào project hiện tại. Refactor lại các luồng đã làm và áp dụng cho những luồng sắp làm trong tương lai.

TÀI LIỆU THAM KHẢO

- Jetpack Compose official site and documents: [Jetpack Compose UI App Development Toolkit - Android Developers](#)
- Android basic with Compose training course: [Android Basics with Compose course | Android Developers](#)
- Android developer medium: [medium.com](#)
- Kotlin and Android developer: [Kotlin and Android | Android Developers](#)
- Kotlin official document: [Kotlin Docs | Kotlin Documentation \(kotlinlang.org\)](#)
- Blog for Compose tutorials: [GeeksforGeeks | A computer science portal for geeks](#), [ProAndroidDev](#)
- Sample repository for architecture design: [android/architecture-samples: A collection of samples to discuss and showcase different architectural tools and patterns for Android apps. \(github.com\)](#)
- Firebase official document: [Firebase Documentation \(google.com\)](#)