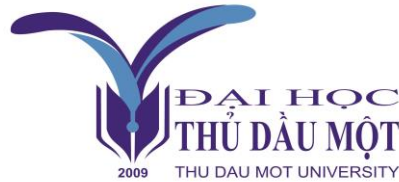


TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT
KHOA KỸ THUẬT – CÔNG NGHỆ



BÁO CÁO TỐT NGHIỆP

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ SỨC KHỎE
TRÊN REACT NATIVE**

Sinh viên thực hiện : NGUYỄN TẤN TRƯỜNG
Lớp : D16PM02
Khoá : 2016 – 2020
Ngành : KỸ THUẬT PHẦN MỀM
Giảng viên hướng dẫn : ThS. NGUYỄN HẢI VĨNH CƯỜNG

Bình Dương, tháng 07 / 2020

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT
KHOA KỸ THUẬT – CÔNG NGHỆ



BÁO CÁO TỐT NGHIỆP

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ SỨC KHỎE
TRÊN REACT NATIVE**

Sinh viên thực hiện : NGUYỄN TẤN TRƯỜNG
Lớp : D16PM02
Khoá : 2016 – 2020
Ngành : KỸ THUẬT PHẦN MỀM
Giảng viên hướng dẫn : ThS. NGUYỄN HẢI VĨNH CƯỜNG

Bình Dương, tháng 07 / 2020

QUYẾT ĐỊNH

**Về việc thực hiện Báo cáo tốt nghiệp trình độ đại học hệ chính quy và thường xuyên,
đợt 7, Năm học: 2019 - 2020**

HIỆU TRƯỞNG TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

Căn cứ Quyết định số 06/QĐ- HDTr ngày 02/7/2019 của Chủ tịch Hội đồng Trường về việc ban hành Quy chế tổ chức và hoạt động của Trường Đại học Thủ Dầu Một;

Căn cứ Quyết định số 1547/QĐ-ĐHTDM ngày 10 tháng 10 năm 2019 của Hiệu trưởng trường Đại học Thủ Dầu Một về việc ban hành Quy chế đào tạo đại học theo học chế tín chỉ;

Căn cứ Quyết định số 1309/QĐ-ĐHTDM ngày 06 tháng 10 năm 2017 và Quyết định số 761/QĐ-ĐHTDM ngày 10 tháng 5 năm 2019 của Hiệu trưởng trường Đại học Thủ Dầu Một về việc sửa đổi, bổ sung một số điều của Quy định về việc thực hiện Khóa luận/ đồ án/ Báo cáo tốt nghiệp;

Xét đề nghị của Trường Phòng Đảm bảo chất lượng,

QUYẾT ĐỊNH:

Điều 1. Giao đề tài Báo cáo tốt nghiệp và giảng viên hướng dẫn cho các Khoa:

1. Khoa Kinh tế

- Chương trình Quản trị Kinh doanh: 02 Báo cáo tốt nghiệp đại học, hệ chính quy;
- Chương trình Quản lý Công nghiệp: 04 Báo cáo tốt nghiệp đại học, hệ chính quy;
- Chương trình Kế toán: 01 Báo cáo tốt nghiệp đại học, hệ chính quy.

2. Khoa Kỹ thuật – Công nghệ

- Chương trình Hệ thống thông tin: 02 Báo cáo tốt nghiệp đại học, hệ chính quy;
- Chương trình Kỹ thuật Phần mềm: 60 Báo cáo tốt nghiệp đại học, hệ chính quy;

- Chương trình Kỹ thuật Điện: 37 Báo cáo tốt nghiệp đại học, hệ chính quy.

3. Khoa Công nghệ thực phẩm

- Chương trình Hóa học: 03 Báo cáo tốt nghiệp đại học, hệ chính quy;
- Chương trình Toán: 03 Báo cáo tốt nghiệp đại học, hệ chính quy.

4. Khoa Kiến trúc

Chương trình Kỹ thuật xây dựng: 03 Báo cáo tốt nghiệp đại học, hệ chính quy và 02 Báo cáo tốt nghiệp đại học, hệ thường xuyên;

5. Khoa Khoa học quản lý

Chương trình Luật: 05 Báo cáo tốt nghiệp đại học, hệ chính quy.

6. Khoa Khoa học Xã hội & Nhân văn

Chương trình Ngữ văn: 01 Báo cáo tốt nghiệp đại học, hệ chính quy.

7. Khoa Sư phạm

Chương trình Giáo dục tiểu học: 01 Báo cáo tốt nghiệp đại học, hệ chính quy.

8. Khoa Ngoại ngữ

- Chương trình Ngôn ngữ Anh: 05 Báo cáo tốt nghiệp đại học, hệ chính quy;
- Chương trình Ngôn ngữ Trung Quốc: 06 Báo cáo tốt nghiệp đại học, hệ chính quy.

(Danh sách kèm theo).

Điều 2. Giảng viên, sinh viên có tên trong danh sách chịu trách nhiệm thực hiện và hoàn thành đề tài Báo cáo tốt nghiệp theo quy định.

Điều 3. Trường Phòng Đảm bảo chất lượng, Trưởng các Phòng, Trưởng khoa, Giám đốc Chương trình có sinh viên làm Báo cáo tốt nghiệp và các cá nhân có tên tại Điều 1 chịu trách nhiệm thi hành Quyết định này.

Quyết định này có hiệu lực kể từ ngày ký./.

Nơi nhận:

- HT, các PHT;
- Như Điều 3;
- Lưu: VP, PĐBCL.

KT. HIỆU TRƯỞNG

PHÒNG HIỆU TRƯỞNG



TS. Ngô Hồng Điệp

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Về thái độ, ý thức của sinh viên:

.....
.....
.....

2. Về đạo đức, tác phong:

.....
.....
.....

3. Về kiến thức:

.....
.....
.....

4. Kết luận :

.....
.....
.....

Điểm:

....., ngày tháng năm 2020

Giảng viên hướng dẫn

Ths. Nguyễn Hải Vĩnh Cường

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

1. Về thái độ, ý thức của sinh viên:

.....

.....

.....

2. Về đạo đức, tác phong:

.....

.....

.....

3. Về kiến thức:

.....

.....

.....

4. Kết luận :

.....

.....

.....

Điểm:

....., ngày tháng năm 2020

Giảng viên phản biện

ThS Trần Văn Hữu

LỜI CẢM ƠN

Để hoàn thành tốt đề tài tốt nghiệp này trước hết em xin gửi đến quý thầy, cô giáo trong khoa Kỹ Thuật - Công Nghệ trường Đại Học Thủ Dầu Một lời cảm ơn chân thành.

Xin cảm ơn đến ban lãnh đạo nhà trường đã tạo mọi điều kiện tốt nhất để người thực hiện đề tài có cơ hội được trình bày ý tưởng của mình, từ đó áp dụng kiến thức, kỹ năng mình có để đưa ý tưởng thành sản phẩm thực tế có thể áp dụng trong đời sống.

Đặc biệt, xin gửi đến thầy Nguyễn Hải Vĩnh Cường, người đã tận tình hướng dẫn, giúp đỡ trong việc nâng cao kiến thức chuyên môn, người luôn đốc thúc, hỗ trợ, hướng dẫn để phát huy cho người thực hiện tính tự học, tự tìm hiểu, từ đó hình thành cho em phương pháp nghiên cứu, tìm tòi, tiếp cận với khoa học công nghệ. Đồng thời trau dồi cho người thực hiện kỹ năng học tập năng động và sáng tạo, giúp cho việc tiếp cận các ứng dụng thông minh trong thực tế một cách thiết thực hơn cũng như hỗ trợ người thực hiện đề tài hoàn thành đề tài nghiên cứu này.

Vì kiến thức bản thân còn hạn chế, nên trong quá trình xây dựng đề tài nghiên cứu khoa học này người thực hiện không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ quý thầy cô.

Xin chân thành cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của ThS. Hồ Đắc Hưng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong luận văn còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung luận văn của mình. Trường Đại học Thủ Dầu Một không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

Bình Dương, ngày 15 tháng 07.năm 2020

Người thực hiện

(ký tên và ghi rõ họ tên)

Nguyễn Tấn Trường

TÓM TẮT

Với khoa học kỹ thuật và công nghệ hiện nay sức khỏe là yếu tố quan trọng nhất đối với mỗi con người, có sức khỏe cơ thể ta mới khỏe mạnh và tham gia các hoạt động tốt hơn. Đặc biệt những lúc khó khăn như hiện nay với dịch COVID-19 đang ảnh hưởng lớn trên toàn thế giới thì việc tự bảo vệ sức khỏe của bản thân được đặt lên hàng đầu. Khi nói đến việc bảo vệ sức khỏe thì ta cần phải có nhiều cách để bảo vệ như ăn uống hợp lý, nghỉ ngơi đúng giấc, kiểm soát được cân nặng, thường xuyên vận động cơ thể,... để cơ thể có được thể trạng tốt nhất trong việc bảo vệ bản thân.

Với việc tự bảo vệ sức khỏe thì chúng ta cần phải có một ứng dụng để bản thân mỗi người có thể theo dõi được tình hình ăn uống, tuần qua đã ăn những gì cũng như lượng calo mà cơ thể đã được nạp vào là bao nhiêu,... Vì vậy việc xây dựng một ứng dụng bảo vệ sức khỏe là rất cần thiết để mọi người có thể tự bảo vệ bản thân mình cũng như theo dõi sức khỏe của gia đình đòi hỏi ứng dụng phải thân thiện và dễ sử dụng nhất với người dùng, mọi người sở hữu Smart phone đều có thể dùng.

Ứng dụng quản lý sức khỏe bao gồm các chức năng chọn món ăn và lưu vào dữ liệu, mỗi món ăn đều có lượng calo riêng để người dùng có thể kiểm soát được calo cho bản thân. Sử dụng biểu đồ hiển thị trực quan cho người dùng dễ dàng thấy được lượng calo trong tuần. Chức năng gợi ý món ăn giúp người dùng có thể lựa chọn món ăn phù hợp với từng lượng calo mà bản thân đã nạp vào cơ thể các ngày qua. Chức năng gợi ý giấc ngủ dựa vào lượng calo trong ngày, độ tuổi người sử dụng mà có thể gợi ý thời gian ngủ phù hợp với bản thân để cơ thể có thể khỏe mạnh nhất sau mỗi giấc ngủ. Ứng dụng xây dựng giúp người dùng có cái nhìn trực quan hơn về sức khỏe và có thể kiểm soát được sức khỏe của mình tốt hơn.

Ứng dụng được xây dựng dựa trên nền tảng React Native, Visual studio code là công cụ lập trình, Dùng cơ sở dữ liệu Realtime Fire base cho việc lưu trữ dữ liệu thay đổi theo thời gian thực. Do sự hạn chế về thời gian cũng như sử dụng công cụ và ngôn ngữ mới nên ứng dụng xây dựng còn sự hạn chế về chức năng, nhưng những chức năng cơ bản cũng đã hoàn thành cho người dùng sử dụng để bảo vệ sức khỏe cho bản thân.

SUMMARY

Nowadays, with the development of science and technology, health is the most important factor for every human being, because a healthy body helps us to participate in activities better. Especially, during difficult times of COVID-19 epidemic, it is affecting seriously worldwide, the protection of our health is the top priority. When it comes to protecting our health, we need to have many ways to protect such as eating properly, resting at rest, controlling weight, regularly doing exercises,... so that the body can have the best condition in protecting yourself.

With the protection of our health, we need to have an application for each person to be able to monitor the situation of meals, namely, measuring the calories of food that the body has been loaded for a week. So, building a health protection app is essential so that everyone can protect themselves and monitor the health of the family that requires the app to be friendly and easy for users, everyone owning a smartphone can use.

Health management application includes the function of selecting dishes and saving data. Each dish has its own calorie so that users can control calories for themselves. Besides, visual graphs are used for users to easily see the calories of each week. The function of food suggestion helps users to choose kinds of food that suit each calorie that they have consumed in the past days. Sleep suggestion function is based on the amount of calories in a day. and the age of the user so that it gives suggestions of sleep time which suitable for themselves, so the body can be healthiest after each sleep. Building applications help users have a more intuitive view of health and control their health better.

The application is built on React Native platform. Visual studio code is a programming tool. The Realtime Fire base database is used for storing data changes in real-time. Due to time constraints and the use of new tools and languages, the application is built with many limitations of functions. However, the basic functions have been completed for users to use to protect their health

MỤC LỤC

Phần A Giới thiệu

LỜI CẢM ƠN	v
DANH MỤC HÌNH.....	xii
DANH MỤC BẢNG	xiii

Phần B Nội Dung

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI	1
1.1. TÊN ĐỀ TÀI.....	1
1.2. LÝ DO CHỌN ĐỀ TÀI.....	1
1.3. MỤC TIÊU CỦA ĐỀ TÀI.....	1
1.4. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU, CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU	2
1.4.1. Đối tượng.....	2
1.4.2. Phạm vi nghiên cứu	3
1.4.3. Cách tiếp cận	3
1.4.4. Phương pháp nghiên cứu.....	3
CHƯƠNG 2: GIẢI PHÁP CÔNG NGHỆ.....	4
2.1. CÔNG NGHỆ VÀ NỀN TẢNG	4
2.2. TỔNG QUAN VỀ NỀN TẢNG NODEJS	4
2.2.1. NodeJS là gì, đặc điểm NodeJS.....	4
2.2.2. Tổng Quan về React Native	6
2.3. TỔNG QUAN VỀ NỀN TẢNG ANDROID.....	9
2.3.1. Tổng quan.....	9
2.3.2. Lịch sử.....	10

2.4. TỔNG QUAN VỀ HỆ THỐNG ĐIỆN TOÁN ĐÁM MÂY GOOGLE FIREBASE.....	12
2.4.1. Firebase Authentication	12
2.4.2. Firebase Realtime Database	14
2.4.3. Firebase Storage.....	16
CHƯƠNG 3: XÁC ĐỊNH YÊU CẦU VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	18
3.1. YÊU CẦU TỔNG THỂ.....	18
3.1.1. Quản lý danh mục	18
3.1.2. Quản lý sức khỏe	18
3.2. YÊU CẦU CHỨC NĂNG ỨNG DỤNG	18
3.2.1. Chức năng tài khoản.	18
3.2.2. Chức năng quản lý sức khỏe.	18
3.3. THIẾT KẾ USECASE.....	19
3.3.1. Danh Sách Actor	19
3.3.2. Biểu đồ hệ thống.....	19
3.3.3. Danh sách Usecase.....	20
3.3.4. Biểu đồ Use case.....	21
3.3.5. Đặc tả Use case.....	22
3.3.6. Biểu đồ tuần tự các chức năng hệ thống	33
3.4. THIẾT KẾ CƠ SỞ DỮ LIỆU	41
3.4.1. Cơ sở dữ liệu biểu đồ	41
3.4.2. Cơ sở dữ liệu chi tiết lịch sử lượng calo	42
3.4.3. Dữ liệu gợi ý món ăn	43
3.4.4. Dữ liệu hình ảnh	44

3.4.5. Dữ liệu thông tin cá nhân	45
CHƯƠNG 4: GIAO DIỆN CHƯƠNG TRÌNH	46
4.1. GIAO DIỆN TÀI KHOẢN.....	46
4.1.1. Giao diện đăng nhập	46
4.1.2. Giao diện đăng ký	47
4.1.3. Giao diện quên mật khẩu.....	48
4.2. GIAO DIỆN QUẢN LÝ SỨC KHỎE	49
4.2.1. Giao diện thông tin cá nhân.....	49
4.2.2. Giao diện thêm, xóa, biểu đồ.....	50
4.2.3. Giao diện xem chi tiết lượng calo theo danh sách.....	51
4.2.4. Giao diện xem chi tiết lịch sử lượng calo	52
4.2.5. Giao diện gợi ý món ăn	53
4.2.6. Giao diện cách chế biến món ăn.....	54
4.2.7. Giao diện gợi ý giấc ngủ	55
CHƯƠNG 5: KẾT LUẬN	56
5.1. ƯU ĐIỂM CỦA ĐỀ TÀI.....	56
5.2. NHƯỢC ĐIỂM CỦA ĐỀ TÀI.....	56
5.3. HƯỚNG PHÁT TRIỂN ĐỀ TÀI.....	56
Phần C Tài liệu tham khảo	
TÀI LIỆU THAM KHẢO	57

DANH MỤC HÌNH

Hình 2.1. Logo NodeJS	4
Hình 2.2 Logo react-native	6
Hình 2.3. Cách cài đặt NodeJS, Python2, jdk8.....	8
Hình 2.4. Cài đặt react-native-CLI.....	9
Hình 2.5. Logo Firebase	12
Hình 3.1. Biểu đồ hệ thống	19
Hình 3.1. Biểu đồ Use case.....	21
Hình 3.2. Biểu đồ tuần tự use case đăng nhập	34
Hình 3.3. Biểu đồ tuần tự use case đăng ký.....	34
Hình 3.4. Biểu đồ tuần tự use case quên mật khẩu	35
Hình 3.5. Biểu đồ tuần tự use case đăng xuất.....	35
Hình 3.6. Biểu đồ tuần tự use case xem lượng calo	36
Hình 3.7. Biểu đồ tuần tự use case thêm món ăn.....	36
Hình 3.8. Biểu đồ tuần tự use case xóa món ăn.....	37
Hình 3.9. Biểu đồ tuần tự use case xem biểu đồ	37
Hình 3.10. Biểu đồ tuần tự use case xem chi tiết bản đồ	38
Hình 3.11. Biểu đồ tuần tự use case xem lịch sử lượng calo.....	38
Hình 3.12. Biểu đồ tuần tự use case gợi ý món ăn.....	39
Hình 3.13. Biểu đồ tuần tự use case xem gợi ý giấc ngủ	39
Hình 3.14. Biểu đồ tuần tự use case xem thông tin cá nhân.....	40
Hình 3.15. Biểu đồ tuần tự use case cập nhật thông tin cá nhân	40
Hình 3.16. Cơ sở dữ liệu biểu đồ	41
Hình 3.17. Cơ sở dữ liệu chi tiết lịch sử lượng calo	42

Hình 3.18. Cơ sở dữ liệu gợi ý món ăn	43
Hình 3.19. Cơ sở dữ liệu hình ảnh	44
Hình 3.20. Cơ sở dữ liệu thông tin cá nhân	45
Hình 4.1. Giao diện đăng nhập.....	46
Hình 4.2. Giao diện đăng ký	47
Hình 4.3. Giao diện quên mật khẩu.....	48
Hình 4.4. Giao diện thông tin cá nhân.....	49
Hình 4.5. Giao diện thêm, xóa, biểu đồ.....	50
Hình 4.6. Giao diện xem chi tiết lượng calo theo danh sách.....	51
Hình 4.7. Giao diện xem chi tiết lịch sử lượng calo	52
Hình 4.8. Giao diện gợi ý món ăn	53
Hình 4.9. Giao diện cách chế biến món ăn	54
Hình 4.10. Giao diện gợi ý giấc ngủ	55

DANH MỤC BẢNG

Bảng 3.1. Danh Sách Actor.....	19
Bảng 3.2. Danh sách Use case	21

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

1.1. TÊN ĐỀ TÀI

“ Xây dựng ứng dụng quản lý sức khỏe trên React native”

1.2. LÝ DO CHỌN ĐỀ TÀI

Thời đại công nghệ hiện nay vấn đề sức khỏe là vấn đề cần phải chú ý nhiều nhất. Phải khỏe mạnh thì chúng ta mới tận hưởng cuộc sống một cách trọn vẹn nhất, làm điều mình thích, ăn món mình muốn ăn, đi nơi mình muốn,... Bởi nếu bệnh tật, ốm đau, chúng ta thường sẽ không còn đủ sức khỏe, tâm trí nào mà lo lắng, suy nghĩ đến những việc khác. Đó là chưa nói đến chuyện bệnh tật còn khiến con người tiêu hao tiền bạc, của cải, ảnh hưởng đến những người thân trong gia đình, xã hội mất đi một người khỏe mạnh.

Do đó, sức khỏe chính là chiếc chìa khóa quan trọng nhất mở ra cánh cửa và việc quản lý sức khỏe cũng là một điều khá là quan trọng đối với mỗi con người.

Các chương trình truyền hình, các tờ báo, các trung tâm điều ưu tiên về vấn đề sức khỏe, mỗi người nên tự chăm sóc và quản lý sức khỏe của bản thân là tự bảo vệ mình cho mình, bảo vệ cho cộng đồng.

Vì vậy việc chọn đề tài xây dựng ứng dụng quản lý sức khỏe là điều nên làm để mỗi người có thể tự quan sát, nhìn nhận và xem xét về sức khỏe của bản thân.

1.3. MỤC TIÊU CỦA ĐỀ TÀI

Mục tiêu chính của đề tài là nghiên cứu, xây dựng một ứng dụng quản lý sức khỏe riêng cho bản thân của người dùng:

- ✓ Người dùng có thể xem những món ăn được gợi ý trên ứng dụng bao gồm: Tên món ăn, nguyên liệu, lượng calo cho món ăn, cách làm, cách chế biến, có hình ảnh trực quan để người dùng có thể theo dõi, học hỏi và chế biến cho bản thân hoặc cả gia đình cùng thưởng thức,
- ✓ Đa dạng về món ăn, mỗi ngày người dùng đều có nhiều lựa chọn về món ăn, Ứng dụng sẽ tự động gợi ý từng món ăn theo từng ngày khác nhau, người dùng có thể thoải mái lựa chọn mà không phải sợ sự hạn chế về món ăn.
- ✓ Giấc ngủ khác là quan trọng với mỗi con người, một người muốn làm việc tốt hay học tập, vận động điều cần phải quản lý giấc ngủ của mình một cách hợp lý, đối với mỗi người để bản thân có sức khỏe thật tốt trong ngày mới.
- ✓ Có 2 loại gợi ý giấc ngủ cho mọi người để mọi người có thể sắp xếp thời gian, giấc ngủ sao cho thoải mái nhất, để cơ thể có thể tràn đầy năng lượng mà không phải gò bó vào duy nhất một mốc thời gian nào đó.
- ✓ Ngoài việc quản lý sức khỏe, hệ thống còn cho phép người dùng có thể quản lý thông tin cá nhân của mình trên hệ thống, có thể upload hình ảnh lên hệ thống.

1.4. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU, CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

1.4.1. Đối tượng

- Tất cả mọi người có sử dụng Smart phone đều có thể tải về ứng dụng và quản lý sức khỏe của mình.
- Đặc biệt tốt đối với những người làm nội trợ có thể lựa chọn những món ăn phù hợp với gia đình của mình hơn.

1.4.2. Phạm vi nghiên cứu

- Thu thập dữ liệu trên các trang về sức khỏe như Sức khỏe và đời sống, blog sức khỏe,... các nhóm quan tâm về sức khỏe, vận dụng kinh nghiệm bản thân từ đó phân tích và áp dụng vào ứng dụng quản lý sức khỏe.

1.4.3. Cách tiếp cận

- Tìm hiểu về ứng dụng di động trên các nền tảng và lựa chọn nền tảng thích hợp cho đề tài.
- Tìm hiểu về các cách quản lý sức khỏe.
- Thu thập ý kiến của bản thân, bạn bè, các trang mạng về sức khỏe, các trang báo, trang thể thao và giấc ngủ.

1.4.4. Phương pháp nghiên cứu

- Nghiên cứu lý thuyết:
 - + Nghiên cứu xây dựng ứng dụng thực hiện trên Android.
 - + Lý thuyết phân tích thiết kế hệ thống.
- Phương pháp thực nghiệm
 - + Lập trình ứng dụng Android.
 - + Tổng hợp thông tin cách quản lý sức khỏe.
 - + Tổng hợp nguồn thực đơn mỗi ngày.
 - + Tổng hợp thông tin về thời gian ngủ.
 - + Đưa dữ liệu vào trong database.

CHƯƠNG 2: GIẢI PHÁP CÔNG NGHỆ

2.1. CÔNG NGHỆ VÀ NỀN TẢNG

Hệ thống quản lý sức khỏe được xây dựng trên:

- Sử dụng Framework React Native dựa trên công nghệ NodeJS
- Sử dụng Android để xây dựng ứng dụng trên di động.
- Sử dụng hệ thống điện toán đám mây Firebase để lưu trữ dữ liệu và tệp.

2.2. TỔNG QUAN VỀ NỀN TẢNG NODEJS

2.2.1. NodeJS là gì, đặc điểm NodeJS



Hình 2.1. Logo NodeJS

- NodeJs là một nền tảng được phát triển độc lập được xây dựng trên javascript runtime của chrome's, với NodeJs chúng ta có thể xây dựng được ứng dụng mạng nhanh chóng và dễ dàng mở rộng. Nodejs được xây dựng và phát triển từ năm 2009 và được bảo trợ từ công ty Joyent. Khả năng đáp ứng của NodeJs là rất nhanh bởi NodeJs được viết hầu hết bởi ngôn ngữ C.

- NodeJs là một ngôn ngữ Server. Nghĩa là NodeJs có đủ khả năng để tạo mọi trang web theo nhu cầu. Giống như nhiều ngôn ngữ Server khác, các page được tạo bằng NodeJs cũng sử dụng Html, Css, Javascript còn NodeJs đóng vai trò là code điều khiển.
- Ưu điểm:
 - + Cú pháp ngắn gọn, dễ nhớ.
 - + Xử lý không đồng bộ tốt
 - + Khả năng xử lý song song tuyệt vời
 - + Có nhiều packages hỗ trợ để tạo nhiều chức năng cho site, các packages được cài đặt bằng câu lệnh giống như cài đặt gem trong ruby on rails
 - + Hỗ trợ nhiều template engine để render view
 - + Có nhiều framework. Được sử dụng phổ biến nhất là: ExpressJs, SailsJS
 - + Khả năng sử dụng lại code: dễ dàng tạo ra các module, helper, library để sử dụng ở nhiều nơi
 - + Hỗ trợ security tốt. NodeJs mặc định hỗ trợ chống tấn công Cross-site script
- Nhược điểm:
 - + Chưa phân tách rõ ràng 3 phần model, view, controller theo mô hình MVC, ngay cả khi sử dụng ExpressJs framework thì vẫn phải thủ công cấu trúc theo MVC
 - + Việc truy xuất dữ liệu thông qua nhiều bảng gặp nhiều hạn chế.

- + Việc custom url tương đối khó khăn.
- + Khi include một module, hay require một class vào một nơi nào đó để dùng cần phải khai báo đường dẫn tuyệt đối, hoặc tương đối tới file chứa class, module đó. Điều này khá bất tiện nếu độ sâu của file chứa class, module lớn
- + Khai báo routing thủ công, nghĩa là muốn một url nào đó có hiệu lực cần phải khai báo thủ công routing cho nó, không có chế độ auto.

2.2.2. Tổng Quan về React Native

2.2.2.1. Tổng quan



Hình 2.2 Logo react-native

- React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động

- Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai.
- Một điểm hay nữa mà mình có đề cập là giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là bạn không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc build và run ứng dụng nữa.
- Và điểm lợi hại kế tiếp của React Native đó chính là chúng ta chỉ cần sử dụng JS để phát triển được một ứng dụng di động hoàn chỉnh, đồng thời giải quyết được các vấn đề mà Native App gặp phải, rồi còn cả kết hợp với code native như Swift, Java, v.v...

2.2.2.2. *Ưu điểm và nhược điểm của React Native*

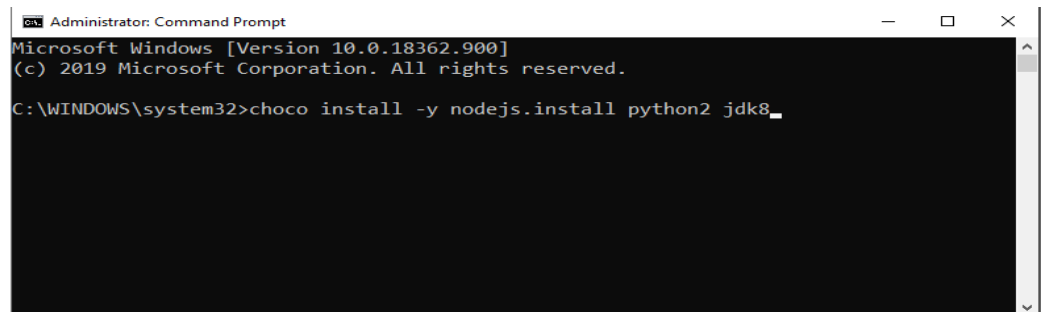
- Ưu điểm:
 - + Hiệu quả về mặt thời gian khi mà bạn muốn phát triển một ứng dụng nhanh chóng.
 - + Hiệu năng tương đối ổn định.
 - + Cộng đồng phát triển mạnh.
 - + Tiết kiệm tiền.
 - + Team phát triển nhỏ.

- + Ứng dụng tin cậy và ổn định.
- + Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
- + Trải nghiệm người dùng tốt hơn là hybrid app
- Nhược điểm:
 - + Vẫn đòi hỏi native code.
 - + Hiệu năng sẽ thấp hơn với app thuần native code.
 - + Bảo mật không cao do dựa trên JS.
 - + Quản lý bộ nhớ.
 - + Khả năng tùy biến cũng không thực sự tốt đối với một vài module.

2.2.2.3. *Cài đặt và sử dụng React Native*

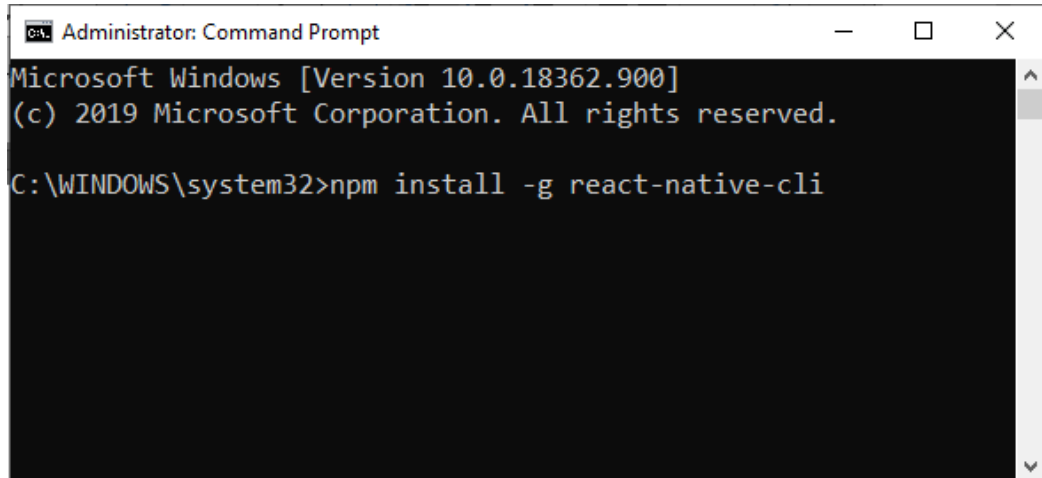
- Cài đặt Node JS, Java SE Development Kit (JDK) và Python2
 - + Truy cập cmd run với quyền admin và nhập câu lệnh và nhấn Enter để cài đặt.

```
choco install -y nodejs.install python2 jdk8
```



Hình 2.3. Cách cài đặt NodeJS, Python2, jdk8

- Cài đặt React Native CLI: `npm install -g react-native-cli`



Hình 2.4. Cài đặt react-native-CLI

2.3. TỔNG QUAN VỀ NỀN TẢNG ANDROID

2.3.1. Tổng quan

- 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 Tổng 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 cầm 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. Vào 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.

2.3.2. Lịch sử

- Tổng công ty Android (Android, Inc.) được thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin (đồng sáng lập công ty Danger), Rich Miner (đồng sáng lập Tổng công ty Viễn thông Wildfire), Nick Sears (từng là Phó giám đốc T-Mobile), và Chris White (trưởng thiết kế và giao diện tại WebTV) để phát triển, theo lời của Rubin, "các thiết bị di động thông minh hơn có thể biết được vị trí và sở thích của người dùng". Dù những người thành lập và nhân viên đều là những người có tiếng tăm, Tổng công ty Android hoạt động một cách âm thầm, chỉ tiết lộ rằng họ đang làm phần mềm dành cho điện thoại di động.
- Google mua lại Tổng công ty Android vào ngày 17 tháng 8 năm 2005, biến nó thành một bộ phận trực thuộc Google. Những nhân viên của chủ chốt của Tổng công ty Android, gồm Rubin, Miner và White, vẫn tiếp tục ở lại công ty làm việc sau thương vụ này.
- Vào thời điểm đó không có nhiều thông tin về Tổng công ty, nhưng nhiều người đồn đoán rằng Google dự tính tham gia thị trường điện thoại di động sau bước đi này.
- Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động phát triển trên nền nhân Linux. Google quảng bá nền tảng này cho các nhà sản xuất điện thoại và các nhà mạng với lời hứa sẽ cung cấp một hệ thống uyển chuyển và có khả năng nâng cấp. Google đã liên hệ với hàng

loạt hãng phần cứng cũng như đối tác phần mềm, bắt tin cho các nhà mạng rằng họ sẵn sàng hợp tác với các cấp độ khác nhau.

- Ngày càng nhiều suy đoán rằng Google sẽ tham gia thị trường điện thoại di động xuất hiện trong tháng 12 năm 2006. Tin tức của BBC và Nhật báo phố Wall chú thích rằng Google muốn đưa công nghệ tìm kiếm và các ứng dụng của họ vào điện thoại di động và họ đang nỗ lực làm việc để thực hiện điều này. Tháng 9 năm 2007, InformationWeek đăng tải một nghiên cứu của Evalueserve cho biết Google đã nộp một số đơn xin cấp bằng sáng chế trong lĩnh vực điện thoại di động.
- Ngày 5 tháng 11 năm 2007, Liên minh thiết bị cầm tay mở (Open Handset Alliance), một hiệp hội bao gồm nhiều công ty trong đó có Texas Instruments, Tập đoàn Broadcom, Google, HTC, Intel, LG, Tập đoàn Marvell Technology, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile được thành lập với mục đích phát triển các tiêu chuẩn mở cho thiết bị di động. Cùng ngày, Android cũng được ra mắt với vai trò là sản phẩm đầu tiên của Liên minh, một nền tảng thiết bị di động được xây dựng trên nhân Linux phiên bản 2.6. Chiếc điện thoại chạy Android đầu tiên được bán ra là HTC Dream, phát hành ngày 22 tháng 10 năm 2008. Biểu trưng của hệ điều hành Android mới là một con rôbốt màu xanh lá cây do hãng thiết kế Irina Blok tại California vẽ.
- Từ năm 2008, Android đã trải qua nhiều lần cập nhật để dần dần cải tiến hệ điều hành, bổ sung các tính năng mới và sửa các lỗi trong những lần phát hành trước. Mỗi bản nâng cấp được đặt tên lần lượt theo thứ tự bảng chữ cái, theo tên của một món ăn tráng miệng; ví dụ như phiên bản 1.5 Cupcake (bánh bông lan nhỏ có kem) tiếp nối bằng phiên bản 1.6. Phiên bản mới nhất hiện nay là 9.0 Pie ra mắt vào tháng 8 năm 2018. Vào năm 2010, Google ra mắt loạt thiết bị Nexus—một dòng sản phẩm bao gồm điện thoại

thông minh và máy tính bảng chạy hệ điều hành Android, do các đối tác phần cứng sản xuất. HTC đã hợp tác với Google trong chiếc điện thoại thông minh Nexus đầu tiên, Nexus One. Kể từ đó nhiều thiết bị mới hơn đã gia nhập vào dòng sản phẩm này, như điện thoại Nexus 4 và máy tính bảng Nexus 10, lần lượt do LG và Samsung sản xuất. Google xem điện thoại và máy tính bảng Nexus là những thiết bị Android chủ lực của mình, với những tính năng phần cứng và phần mềm mới nhất của Android.

2.4. TỔNG QUAN VỀ HỆ THỐNG ĐIỆN TOÁN ĐÁM MÂY GOOGLE FIREBASE



Hình 2.5. Logo Firebase

2.4.1. Firebase Authentication

- Hầu hết các ứng dụng cần biết danh tính của người dùng. Biết danh tính của người dùng cho phép ứng dụng lưu dữ liệu người dùng trên đám mây một cách an toàn và cung cấp trải nghiệm được cá nhân hóa giống nhau trên tất cả các thiết bị của người dùng.
- Firebase Authentication cung cấp dịch vụ back-end, SDK dễ sử dụng và thư viện UI được tạo sẵn để xác thực người dùng với ứng dụng của bạn. Hỗ trợ xác thực bằng mật khẩu, số điện thoại, nhà cung cấp nhận dạng liên kết phổ biến như Google, Facebook và Twitter, ...

- Firebase Authentication tích hợp chặt chẽ với các dịch vụ Firebase khác và kết hợp với các tiêu chuẩn công nghiệp như OAuth 2.0 và OpenID Connect, do đó có thể dễ dàng tích hợp với phụ trợ tùy chỉnh.
- Khả năng chính
 - + FirebaseUI Auth cung cấp giải pháp xác thực thả xuống xử lý các luồng UI để đăng nhập người dùng bằng địa chỉ email và mật khẩu, số điện thoại và với các nhà cung cấp nhận dạng được liên kết phổ biến, bao gồm Đăng nhập Google và Đăng nhập Facebook.
 - + Firebase SDK Authentication
 - Xác thực dựa trên email và mật khẩu: Xác thực người dùng bằng địa chỉ email và mật khẩu của họ. Firebase Authentication SDK cung cấp các phương thức để tạo và quản lý người dùng sử dụng địa chỉ email và mật khẩu của họ để đăng nhập. Firebase Authentication cũng xử lý gửi email đặt lại mật khẩu.
 - Federated identity provider integration: Xác thực người dùng bằng cách tích hợp với các nhà cung cấp nhận dạng liên kết. Firebase Authentication SDK cung cấp các phương thức cho phép người dùng đăng nhập bằng tài khoản Google, Facebook, Twitter và GitHub của họ.
 - Xác thực số điện thoại: Xác thực người dùng bằng cách gửi tin nhắn SMS đến điện thoại của họ.
 - Tích hợp hệ thống xác thực tùy chỉnh: Kết nối hệ thống đăng nhập hiện có của ứng dụng của bạn với Firebase Authentication SDK và có quyền truy cập vào Firebase Realtime Database và các dịch vụ Firebase khác.

- Ẩn danh: Sử dụng các tính năng yêu cầu xác thực mà không yêu cầu người dùng đăng nhập trước bằng cách tạo tài khoản ẩn danh tạm thời. Nếu người dùng sau đó chọn đăng ký, bạn có thể nâng cấp tài khoản ẩn danh thành tài khoản thông thường, để người dùng có thể tiếp tục nơi họ rời đi.
- Cơ chế hoạt động
- + Để đăng nhập người dùng vào ứng dụng của bạn, trước tiên bạn phải có thông tin xác thực từ người dùng. Các thông tin đăng nhập này có thể là địa chỉ email và mật khẩu của người dùng hoặc mã thông báo OAuth từ nhà cung cấp nhận dạng được liên kết. Sau đó, bạn chuyển các thông tin đăng nhập này đến Firebase Authentication SDK. Các dịch vụ back-end của Firebase sau đó sẽ xác minh các thông tin đăng nhập đó và trả lời phản hồi cho khách hàng.
- + Sau khi đăng nhập thành công, bạn có thể truy cập thông tin hồ sơ cơ bản của người dùng và bạn có thể kiểm soát quyền truy cập của người dùng vào dữ liệu được lưu trữ trong các sản phẩm Firebase khác. Bạn cũng có thể sử dụng mã thông báo xác thực được cung cấp để xác minh danh tính của người dùng trong các dịch vụ phụ trợ của riêng bạn.

2.4.2. **Firebase Realtime Database**

- Firebase Realtime Database lưu trữ và đồng bộ dữ liệu dạng NoSQL trên cloud database. Dữ liệu được đồng bộ hóa trên tất cả các máy trạm trong thời gian thực và vẫn khả dụng khi ứng dụng trong trạng thái ngoại tuyến.
- Firebase Realtime Database là một cloud-hosted database. Dữ liệu được lưu trữ dưới dạng JSON và được đồng bộ hóa trong thời gian thực cho

mọi máy khách được kết nối. Khi xây dựng các ứng dụng đa nền tảng với SDK iOS, Android và JavaScript, tất cả các khách hàng của bạn sẽ chia sẻ một phiên bản Cơ sở dữ liệu thời gian thực và tự động nhận các bản cập nhật với dữ liệu mới nhất.

- Cơ chế hoạt động
 - + Firebase Realtime Database cho phép xây dựng các ứng dụng một cách phong phú bằng cách cho phép truy cập an toàn vào cơ sở dữ liệu trực tiếp từ mã phía máy khách. Dữ liệu được duy trì cục bộ và ngay cả khi ngoại tuyến, các sự kiện thời gian thực vẫn tiếp tục phát sinh, mang đến cho người dùng cuối trải nghiệm phản hồi. Khi thiết bị lấy lại kết nối, Cơ sở dữ liệu thời gian thực sẽ đồng bộ hóa các thay đổi dữ liệu cục bộ với các cập nhật từ xa xảy ra trong khi máy khách ngoại tuyến, tự động hợp nhất mọi xung đột.
 - + Realtime Database cung cấp một ngôn ngữ quy tắc dựa trên biểu thức linh hoạt, được gọi là Firebase Realtime Database Security Rules, để xác định cách cấu trúc dữ liệu của bạn và khi nào có thể đọc hoặc ghi dữ liệu. Khi được tích hợp với Firebase Authentication, nhà phát triển có thể xác định ai có quyền truy cập vào dữ liệu nào và cách họ có thể truy cập dữ liệu đó.
 - + Realtime Database là một cơ sở dữ liệu NoQuery và do đó có các tối ưu hóa và chức năng khác nhau so với cơ sở dữ liệu quan hệ. Realtime Database API được thiết kế để chỉ cho phép các hoạt động có thể được thực thi nhanh chóng. Điều này cho phép xây dựng trải nghiệm thời gian thực tuyệt vời có thể phục vụ hàng triệu người dùng mà không ảnh hưởng đến khả năng phản ứng. Do đó, điều quan trọng là phải suy nghĩ về cách người dùng cần truy cập dữ liệu của bạn và sau đó cấu trúc dữ liệu phù hợp.

2.4.3. Firebase Storage

- Cloud Storage được xây dựng cho các nhà phát triển ứng dụng cần lưu trữ và tương tác nội dung do người dùng tạo, chẳng hạn như ảnh hoặc video.
- Cloud Storage của Firebase là dịch vụ lưu trữ đối tượng mạnh mẽ, đơn giản và tiết kiệm chi phí được xây dựng cho quy mô Google. Firebase SDK cho Cloud Storage thêm bảo mật của Google vào tệp tải lên và tải xuống cho các ứng dụng Firebase, bất kể chất lượng mạng. Sử dụng SDK của Firebase để lưu trữ hình ảnh, âm thanh, video hoặc nội dung khác do người dùng tạo. Trên máy chủ, sử dụng Google Cloud Storage, để truy cập cùng các tệp.
- Khả năng chính
 - + Hoạt động mạnh mẽ: SDK Firebase cho Cloud Storage thực hiện tải lên và tải xuống bất kể chất lượng mạng. Tải lên và tải xuống mạnh mẽ, có nghĩa là chúng khởi động lại ở nơi chúng dừng lại, tiết kiệm thời gian và băng thông của người dùng của bạn.
 - + Độ bảo mật cao: SDK Firebase cho Cloud Storage tích hợp với Firebase Authentication để cung cấp xác thực đơn giản và trực quan cho các nhà phát triển. Bạn có thể sử dụng mô hình bảo mật khai báo của chúng tôi để cho phép truy cập dựa trên tên tệp, kích thước, loại nội dung và metadata.
 - + Khả năng mở rộng cao: SDK Firebase cho Cloud Storage được xây dựng cho quy mô exabyte khi ứng dụng của bạn bị virus. Dễ dàng phát triển từ nguyên mẫu đến sản xuất bằng cách sử dụng cùng một cơ sở hạ tầng hỗ trợ Spotify và Google Photos.
- Cơ chế hoạt động

- + Các nhà phát triển sử dụng SDK Firebase cho Cloud Storage để tải lên và tải xuống các tệp trực tiếp từ máy khách. Nếu kết nối mạng kém, máy khách có thể thử lại thao tác ngay tại nơi nó dừng, tiết kiệm thời gian và băng thông cho người dùng.
- + Cloud Storage lưu trữ các tệp trong nhóm Google Cloud Storage, giúp chúng có thể truy cập được thông qua cả Firebase và Google Cloud. Điều này cho phép linh hoạt tải lên và tải xuống các tệp từ ứng dụng khách di động thông qua SDK Firebase và xử lý phía máy chủ như lọc hình ảnh hoặc chuyển mã video bằng Google Cloud Platform. Cloud Storage tự động chia tỷ lệ, nghĩa là không cần phải di chuyển sang bất kỳ nhà cung cấp nào khác.
- + SDK Firebase cho Cloud Storage tích hợp hoàn hảo với Firebase Authentication để xác định người dùng và Firebase cung cấp ngôn ngữ bảo mật khai báo cho phép bạn đặt các điều khiển truy cập trên các tệp hoặc nhóm tệp riêng lẻ để có thể đặt tệp ở chế độ công khai hoặc riêng tư theo ý muốn.

CHƯƠNG 3: XÁC ĐỊNH YÊU CẦU VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. YÊU CẦU TỔNG THỂ

3.1.1.Quản lý danh mục

- Danh mục tài khoản.
- Danh mục quản lý sức khỏe.

3.1.2.Quản lý sức khỏe

- Quản lý lượng calo.
- Gợi ý món ăn.
- Gợi ý thời gian ngủ.

3.2. YÊU CẦU CHỨC NĂNG ỨNG DỤNG

3.2.1.Chức năng tài khoản.

- Đăng ký tài khoản.
- Đăng nhập tài khoản.
- Lấy lại mật khẩu tài khoản.

3.2.2.Chức năng quản lý sức khỏe.

- Quản lý lượng calo.
- Quản lý biểu đồ tuần qua.
- Quản lý lịch sử lượng calo.
- Thêm món ăn vào quản lý calo trong ngày.
- Xóa món ăn của quản lý calo trong ngày

- Gợi ý món ăn dựa vào ngày.
- Gợi ý thời gian thức cho giấc ngủ.

3.3. THIẾT KẾ USECASE

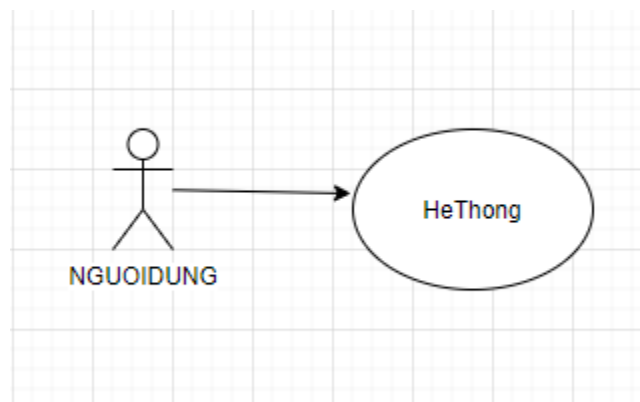
3.3.1. Danh Sách Actor

Một actor hay tác nhân ngoài là một vai trò của một hay nhiều người hay vật thể trong sự tương tác với hệ thống.

STT	Tác Nhân	Mô Tả
1	Người Dùng	Là người sử dụng hệ thống, Có quyền truy cập vào ứng dụng.

Bảng 3.1. Danh Sách Actor

3.3.2. Biểu đồ hệ thống.



Hình 3.1. Biểu đồ hệ thống

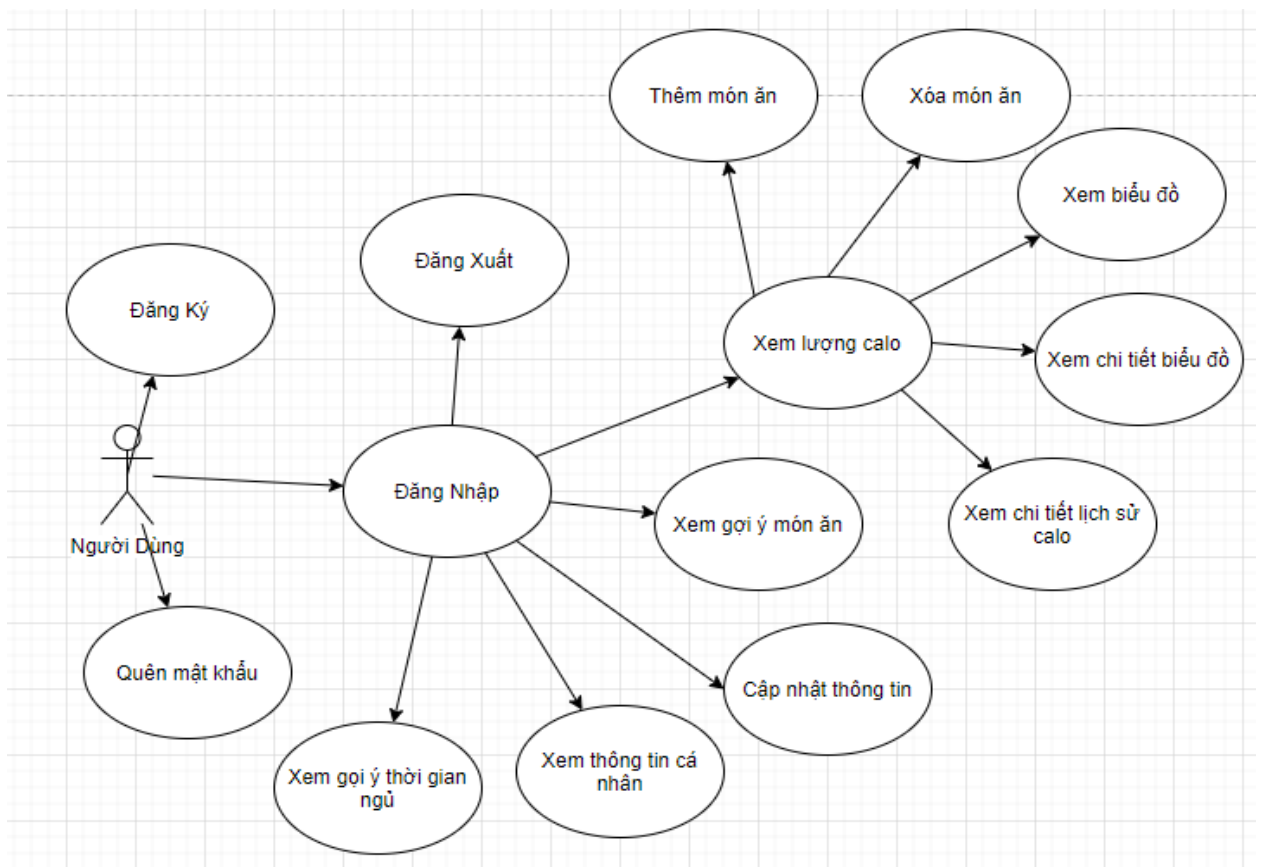
3.3.3.Danh sách Usecase

STT	Use case	Ý nghĩa	Actor
1	Đăng Nhập	Đăng nhập và truy cập hệ thống	Người dùng
2	Đăng ký	Người dùng đăng kí tài khoản	Người dùng
3	Quên mật khẩu	Người dùng nhập email và đổi mật khẩu	Người dùng
4	Đăng Xuất	Đăng xuất khỏi hệ thống	Người dùng
5	Xem Lượng calo	Xem lượng calo trong ngày	Người dùng
6	Thêm món ăn	Người dùng thêm món ăn vào các buổi sáng trưa tối	Người dùng
7	Xóa món ăn	Người dùng xóa món ăn vào các buổi sáng trưa tối	Người dùng
8	Xem biểu đồ	Người dùng xem biểu đồ lượng calo 7 ngày qua	Người dùng
9	Xem chi tiết biểu đồ	Người dùng xem chi tiết lượng calo theo các ngày	Người dùng
10	Xem lịch sử lượng calo theo ngày	Người dùng xem lượng calo của một ngày nào đó.	Người dùng
11	Xem gợi ý món	Người dùng có thể xem gợi	Người dùng

	ăn	ý món ăn	
12	Xem gợi ý thời gian ngủ	Người dùng có thể xem gợi ý thời gian ngủ	Người dùng
13	Xem thông tin cá nhân	Người dùng có thể xem thông tin của mình	Người dùng
14	Cập nhật thông tin cá nhân	Người dùng có thể cập nhật thông tin của mình	Người dùng

Bảng 3.2. Danh sách Use case

3.3.4. Biểu đồ Use case



Hình 3.1. Biểu đồ Use case

3.3.5.Đặc tả Use case

3.3.5.1. Use case Đăng kí

- Tóm tắt: Người dùng đăng kí tài khoản để truy cập vào hệ thống.
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Người dùng nhập thông tin gmail.
 - Người dùng nhập tên.
 - Người dùng nhập mật khẩu.
 - Người dùng nhấn đăng kí.
 - + Dòng sự kiện khác: Người dùng đăng kí sai tài khoản email hoặc trùng thì phải nhập lại từ đầu.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình đăng kí và chưa tạo tài khoản thành công trên hệ thống.
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình đăng kí và tạo thành công tài khoản trên hệ thống.
- Điểm mở rộng: không có.

3.3.5.2. Use case đăng nhập

- Tóm tắt: Người dùng đăng nhập tài khoản để vào hệ thống.
- Actor: Người dùng.

- Dòng sự kiện:
- + Dòng sự kiện chính:
 - Người dùng nhập thông tin gmail.
 - Người dùng nhập mật khẩu.
 - Người dùng nhấn đăng nhập.
- + Dòng sự kiện khác: người dùng đăng nhập sai thì phải nhập lại
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình đăng nhập, các chức năng k hiển thị.
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình Thông Tin cá nhân và truy cập thành công vào hệ thống.
- Điểm mở rộng: không có.

3.3.5.3. Use case đăng xuất

- Tóm tắt: Người dùng đăng xuất ra khỏi hệ thống.
- Actor: Người dùng.
- Dòng sự kiện:
- + Dòng sự kiện chính:
 - Người dùng nhấn đăng xuất
 - Giao diện về trang đăng nhập
- + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.

- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình thông tin cá nhân.
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình đăng nhập
- Điểm mở rộng: không có.

3.3.5.4. Use case quên mật khẩu

- Tóm tắt: Người dùng lấy lại mật khẩu tài khoản của mình.
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - o Người dùng nhấn quên mật khẩu.
 - o Người dùng nhập tài khoản email.
 - o Người dùng nhấp lấy lại mật khẩu.
 - o Người dùng kiểm tra email và cập nhật mật khẩu mới.
 - + Dòng sự kiện khác: Người dùng sẽ điền mật khẩu và nhập lại mật khẩu tại email vừa nhận được.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình quên mật khẩu.
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình quên mật khẩu.
- Điểm mở rộng: không có.

3.3.5.5. Xem thông tin cá nhân

- Tóm tắt: Người dùng xem thông tin cá nhân của mình
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Người dùng xem tài khoản email.
 - Người dùng xem ID.
 - Người dùng xem tên của mình.
 - Người dùng xem ảnh đại diện.
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình thông tin cá nhân.
- Trạng thái hệ thống sau khi thực hiện use case: Thông tin cá nhân
- Điểm mở rộng: không có.

3.3.5.6. Use case cập nhật thông tin cá nhân.

- Tóm tắt: Người dùng cập nhật thông tin cá nhân của mình
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:

- Người dùng cập nhật tên của mình.
- Người dùng cập nhật ảnh đại diện.
- + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình cập nhật thông tin cá nhân.
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình cập nhật thông tin cá nhân.
- Điểm mở rộng: không có.

3.3.5.7. Use case xem lượng calo

- Tóm tắt: Người dùng xem lượng calo trong ngày của mình
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Người dùng nhìn thấy lượng calo mà mình nạp vào cơ thể
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình xem lượng calo
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình xem lượng calo.

- Điểm mở rộng: không có.

3.3.5.8. Use case thêm món ăn

- Tóm tắt: Người dùng thêm món ăn vào các buổi: sáng, trưa, tối
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - o Người dùng nhấn giữ vào món ăn
 - o Người dùng tùy chọn thêm món ăn vào buổi sáng trưa hoặc tối
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình xem lượng calo
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình xem lượng calo và thấy danh sách món ăn đã được thêm vào.
- Điểm mở rộng: không có.

3.3.5.9. Use case xóa món ăn

- Tóm tắt: Người dùng xóa món ăn vào các buổi: sáng, trưa, tối
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - o Người dùng nhấn vào tên món ăn

- Người dùng nhấn chọn thông báo xóa thành công
- + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình xem lượng calo
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình xem lượng calo và thấy danh sách món ăn đã được thay đổi sau khi xóa món ăn
- Điểm mở rộng: không có.

3.3.5.10. Use case xem biểu đồ

- Tóm tắt: Người dùng xem lượng calo 7 ngày vừa qua bằng biểu đồ.
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Người dùng xem được thông tin lượng calo của 7 ngày qua
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình xem lượng calo
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình xem lượng calo và thấy biểu đồ lượng calo.
- Điểm mở rộng: không có.

3.3.5.11. Use case xem chi tiết biểu đồ

- Tóm tắt: Người dùng xem chi tiết lượng calo tiêu thụ qua danh sách các ngày
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Người dùng nhấn vào xem chi tiết
 - Người dùng nhìn thấy được danh sách lượng calo theo các ngày
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình xem lượng calo
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình xem chi tiết lượng và thấy danh sách lượng calo tiêu thụ theo danh sách ngày.
- Điểm mở rộng: không có.

3.3.5.12. Use case xem lịch sử lượng calo theo ngày

- Tóm tắt: Người dùng xem lượng calo tiêu thụ trong ngày hôm đó và các món ăn mình đã ăn.
- Actor: Người dùng.
- Dòng sự kiện:

+ Dòng sự kiện chính:

- Người dùng nhấn vào danh sách của lượng calo theo ngày
- Người dùng được chuyển qua màn hình xem lịch sử món ăn theo ngày
- Người dùng xem được chi tiết lịch sử lượng calo và món ăn ngày hôm đó

+ Dòng sự kiện khác: không có.

- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: Đứng ở màn hình xem lượng calo theo danh sách ngày
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình xem lịch sử lượng calo và thấy danh sách món ăn đã được nạp vào cơ thể ở ngày hôm đó
- Điểm mở rộng: không có.

3.3.5.13. Use case xem gợi ý món ăn

- Tóm tắt: Người dùng xem các món ăn được hệ thống gợi ý
- Actor: Người dùng.
- Dòng sự kiện:

+ Dòng sự kiện chính:

- Người dùng xem danh sách món ăn.
- Người dùng nhấn vào món ăn.
- Người dùng xem cách chế biến món ăn, Thông tin món ăn.

- + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: đứng ở màn hình danh sách món ăn.
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình thông tin chi tiết món ăn.
- Điểm mở rộng: không có.

3.3.5.14. Use case xem gợi ý giấc ngủ

- Tóm tắt: Người dùng xem các gợi ý giấc ngủ của hệ thống.
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Người dùng chọn số giờ mình dự tính đi ngủ.
 - Người dùng chọn phút mà mình chuẩn bị ngủ.
 - Người dùng nhấn vào xem gợi ý giấc ngủ.
 - Người dùng thấy được thời gian ngủ được gợi ý trên giao diện giấc ngủ
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: đứng ở màn hình gợi ý giấc ngủ.

- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình gọi ý giấc ngủ và nhìn thấy thời gian thức giấc được gợi ý.
- Điểm mở rộng: không có.

3.3.5.15. Use xem thông tin cá nhân

- Tóm tắt: Người dùng xem thông tin cá nhân của mình
- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - o Người dùng xem họ và tên
 - o Người dùng xem năm sinh
 - o Người dùng xem chiều cao
 - o Người dùng xem cân nặng
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: đứng ở màn hình thông tin cá nhân
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình thông tin cá nhân
- Điểm mở rộng: không có.

3.3.5.16. Use case cập nhật thông tin cá nhân

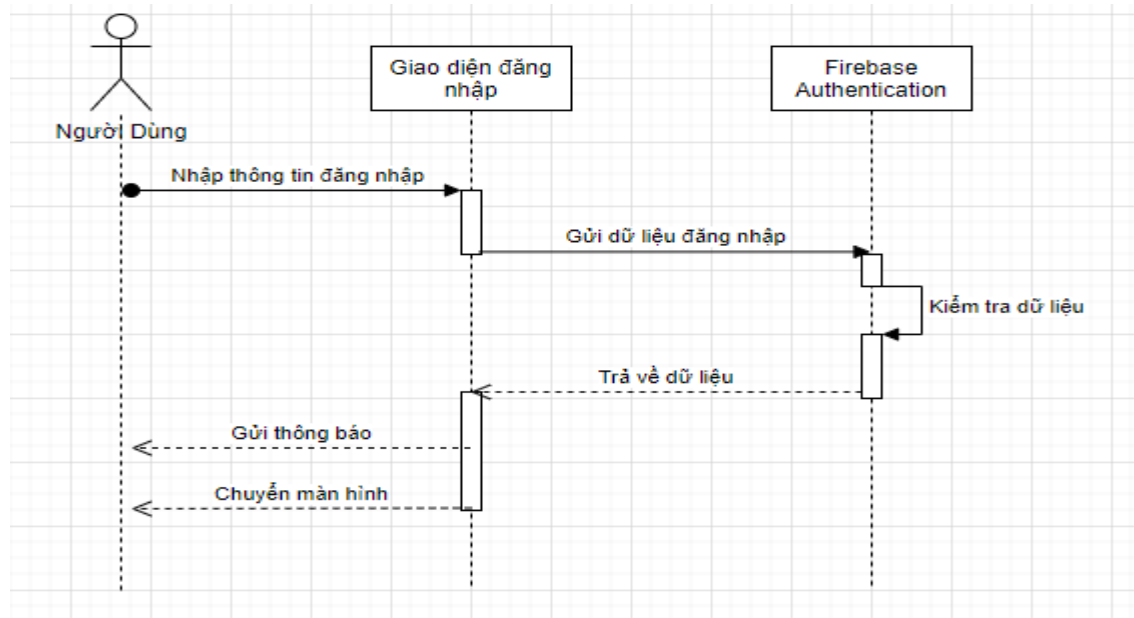
- Tóm tắt: Người dùng cập nhật thông tin cá nhân của mình

- Actor: Người dùng.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Người dùng nhập họ và tên.
 - Người dùng nhập năm sinh
 - Người dùng nhập chiều cao
 - Người dùng nhập cân nặng
 - Người dùng nhấn xác nhận cập nhật.
 - + Dòng sự kiện khác: không có.
- Yêu cầu đặc biệt: không có.
- Trạng thái hệ thống trước khi thực hiện use case: đứng ở màn hình cập nhật thông tin cá nhân.
- Trạng thái hệ thống sau khi thực hiện use case: Đứng ở màn hình xem thông tin cá nhân.
- Điểm mở rộng: không có.

3.3.6. Biểu đồ tuần tự các chức năng hệ thống

3.3.6.1. Biểu đồ tuần tự use case đăng nhập

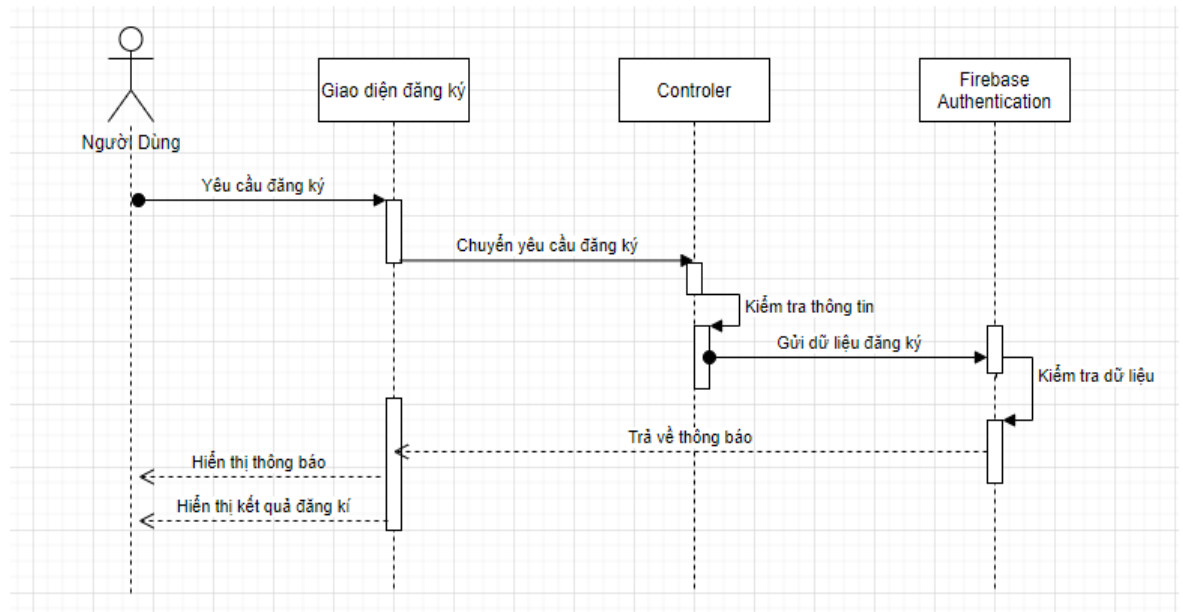
- Biểu đồ tuần tự của use case đăng nhập chỉ rõ hoạt động của người dùng và luồng dữ liệu khi đăng nhập



Hình 3.2. Biểu đồ tuần tự use case đăng nhập

3.3.6.2. Biểu đồ tuần tự use case đăng ký

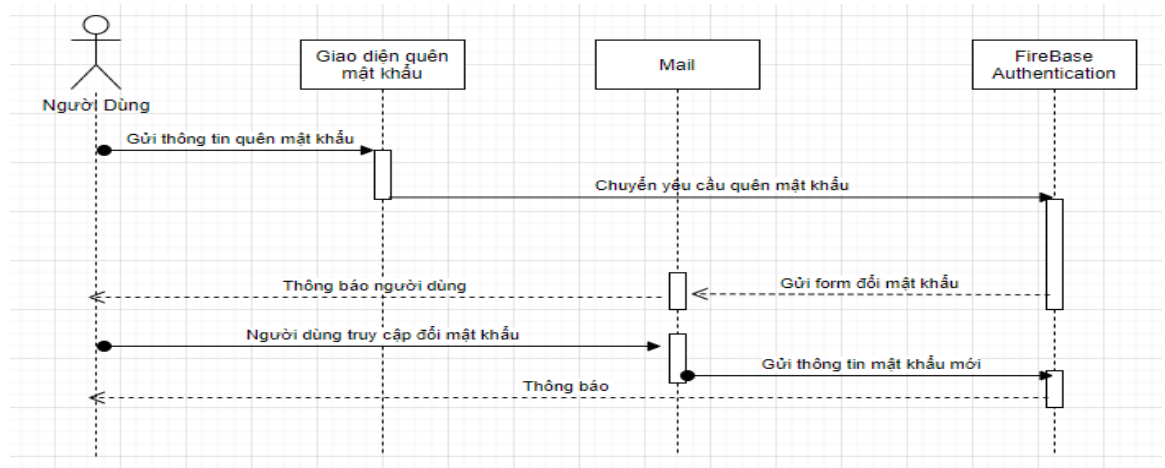
- Biểu đồ tuần tự của use case đăng ký chỉ rõ hoạt động và luồng xử lý dữ liệu sau khi người dùng đăng ký tài khoản



Hình 3.3. Biểu đồ tuần tự use case đăng ký

3.3.6.3. Biểu đồ tuần tự use case quên mật khẩu

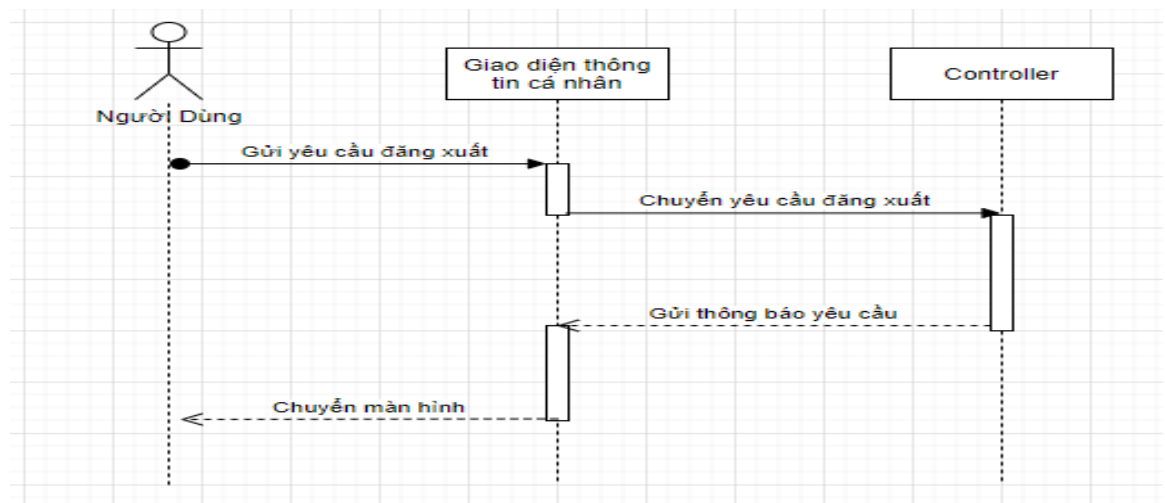
- Biểu đồ tuần tự của use case quên mật khẩu chỉ rõ hoạt động và luồng xử lý dữ liệu sau khi người dùng nhập thông tin và xác nhận để có thể đổi mật khẩu tại đường dẫn được gửi về mail



Hình 3.4. Biểu đồ tuần tự use case quên mật khẩu

3.3.6.4. Biểu đồ tuần tự use case đăng xuất

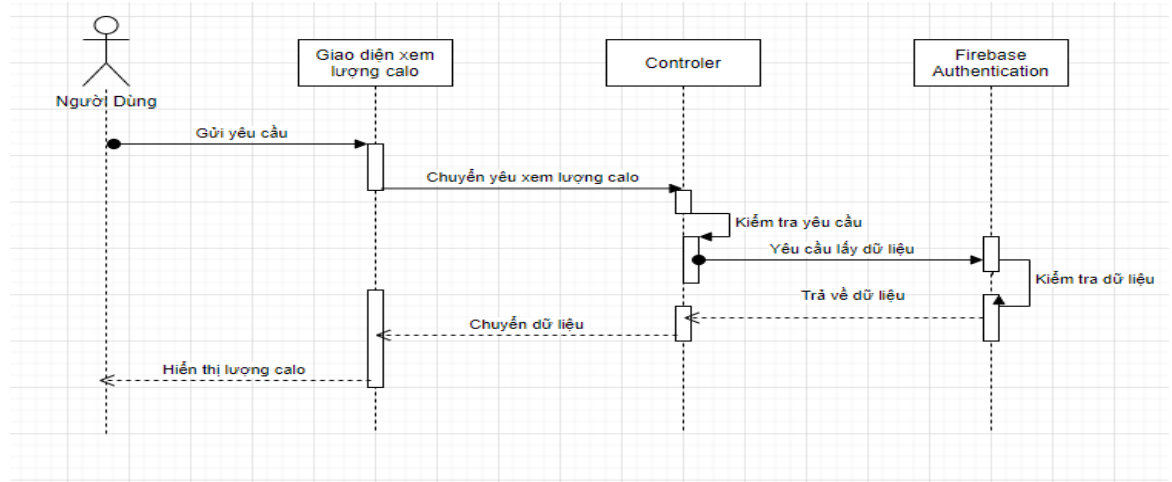
- Biểu đồ tuần tự của use case đăng xuất chỉ rõ hoạt động và luồng xử lý dữ liệu sau khi người dùng đăng xuất tài khoản



Hình 3.5. Biểu đồ tuần tự use case đăng xuất

3.3.6.5. Biểu đồ tuần tự use case xem lượng calo

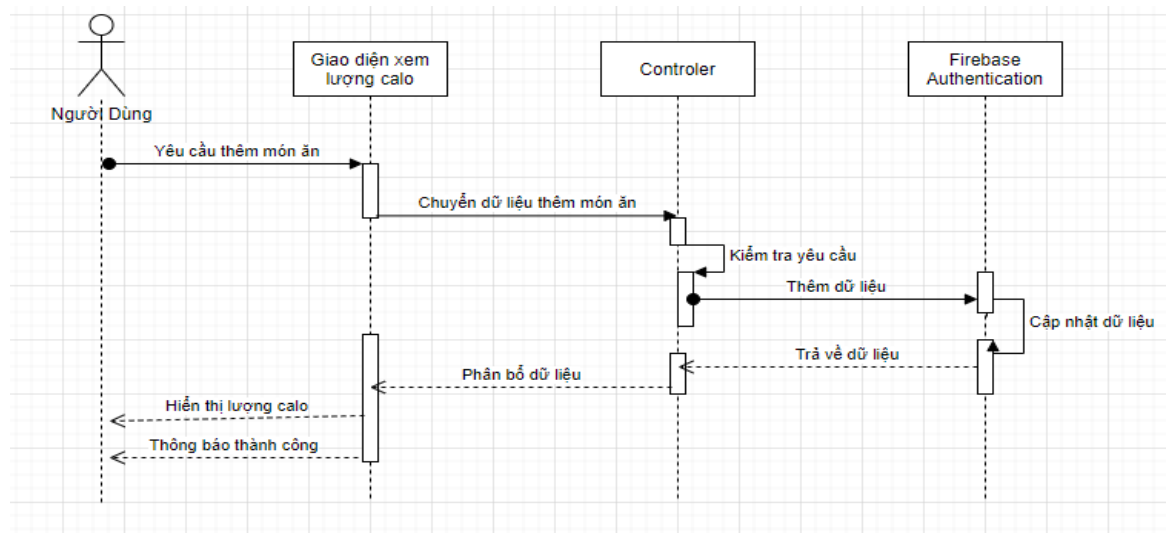
- Biểu đồ tuần tự của use case xem lượng calo hiển thị lượng calo của người dùng và luồng xử lý, hiển thị dữ liệu



Hình 3.6. Biểu đồ tuần tự use case xem lượng calo

3.3.6.6. Biểu đồ tuần tự use case thêm món ăn

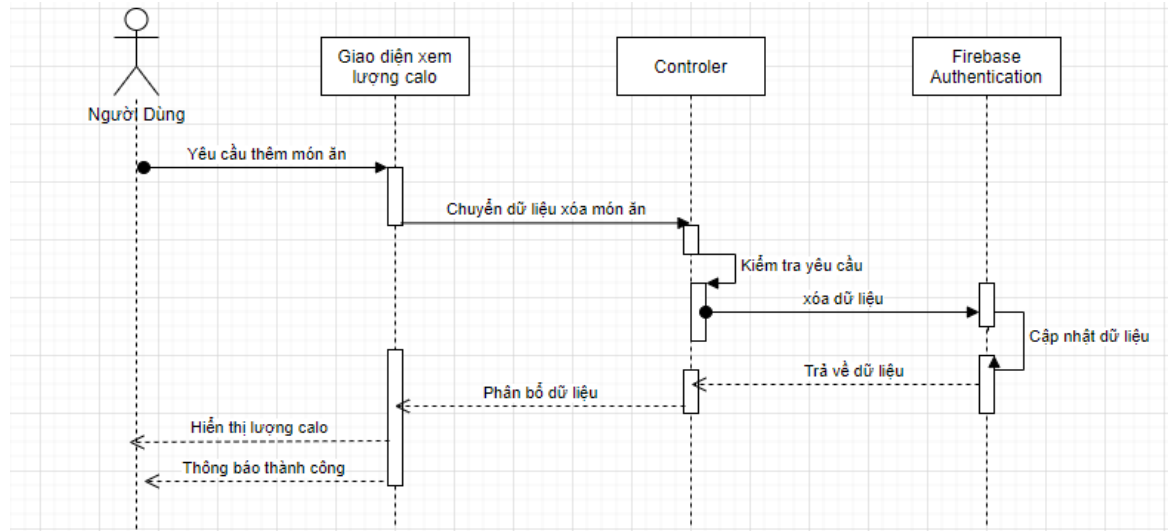
- Biểu đồ tuần tự use case thêm món ăn thể hiện sự xử lý và hiển thị trên giao diện



Hình 3.7. Biểu đồ tuần tự use case thêm món ăn

3.3.6.7. Biểu đồ tuần tự use case xóa món ăn

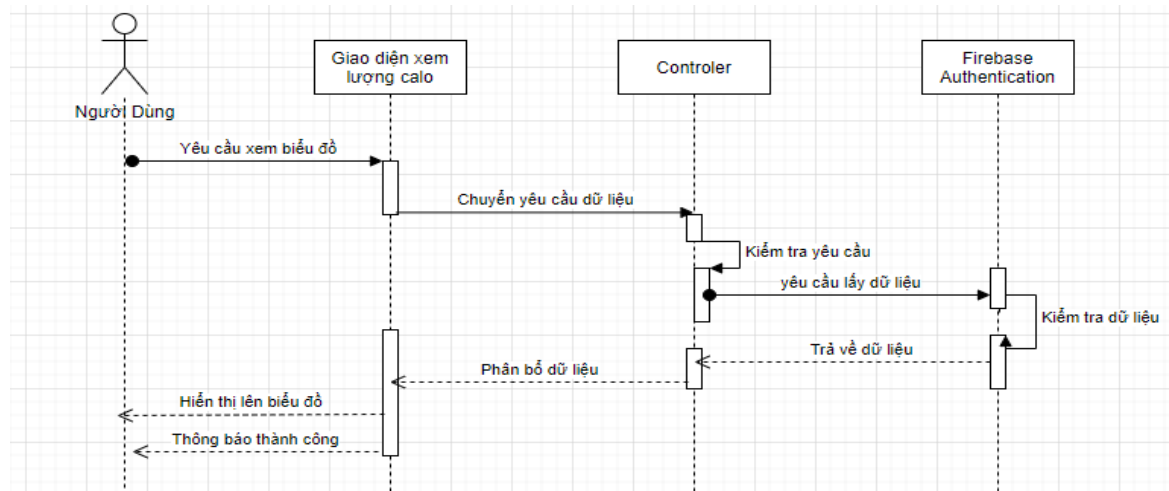
- Biểu đồ tuần tự use case xóa món ăn thể hiện yêu cầu xóa món của người dùng và cách thức hoạt động của dữ liệu



Hình 3.8. Biểu đồ tuần tự use case xóa món ăn

3.3.6.8. Biểu đồ tuần tự use case xem biểu đồ

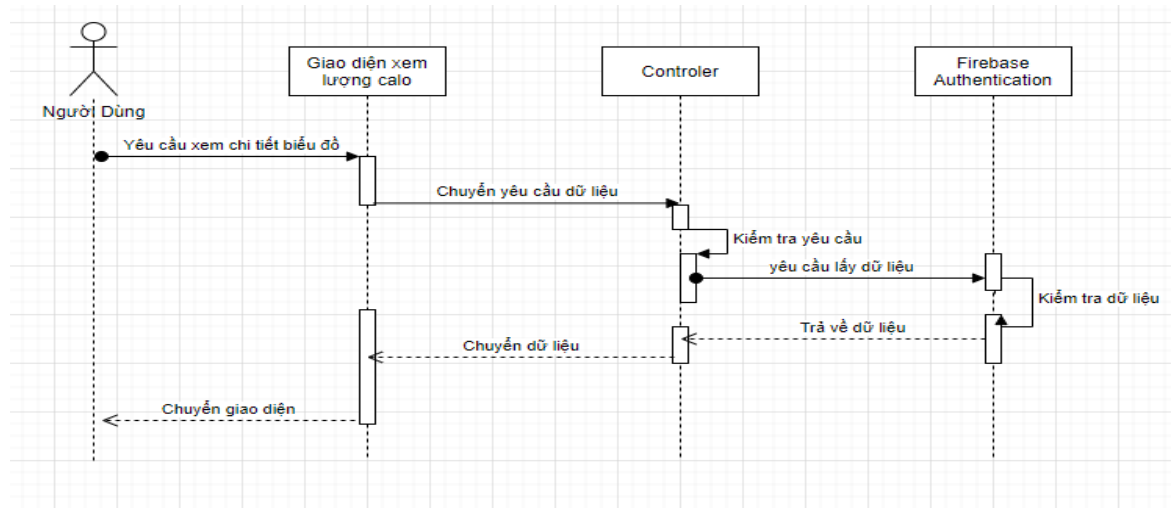
- Biểu đồ tuần tự use case xem biểu đồ thể hiện lượng sự lấy dữ liệu và hiển thị cho người dùng



Hình 3.9. Biểu đồ tuần tự use case xem biểu đồ

3.3.6.9. Biểu đồ tuần tự use case xem chi tiết biểu đồ

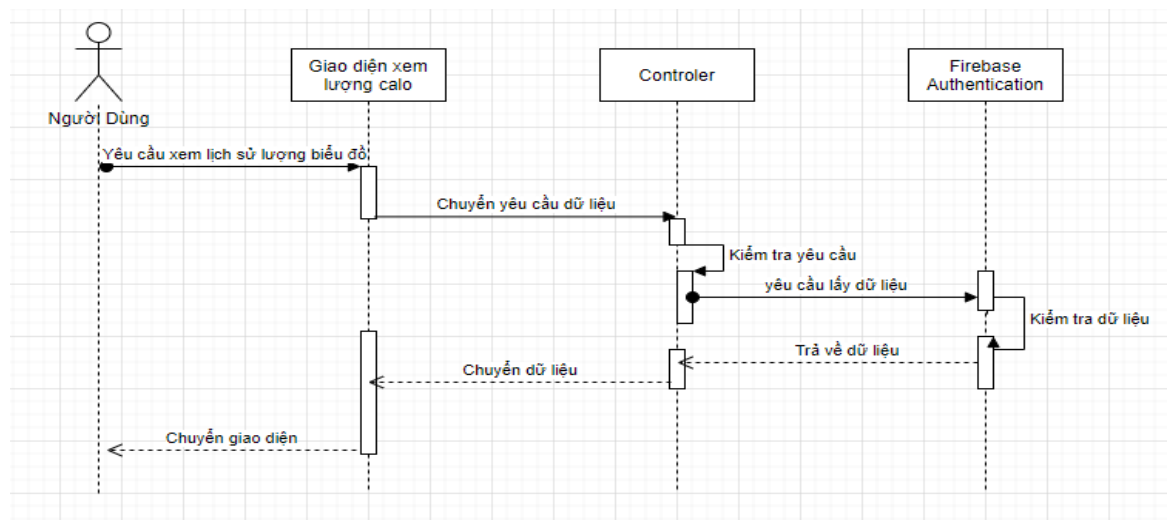
- Biểu đồ tuần tự use case xem chi tiết biểu đồ chỉ ra cách thức mà ứng dụng xử lý yêu cầu xem chi tiết biểu đồ của người dùng



Hình 3.10. Biểu đồ tuần tự use case xem chi tiết bản đồ

3.3.6.10. Biểu đồ tuần tự use case xem lịch sử lượng calo

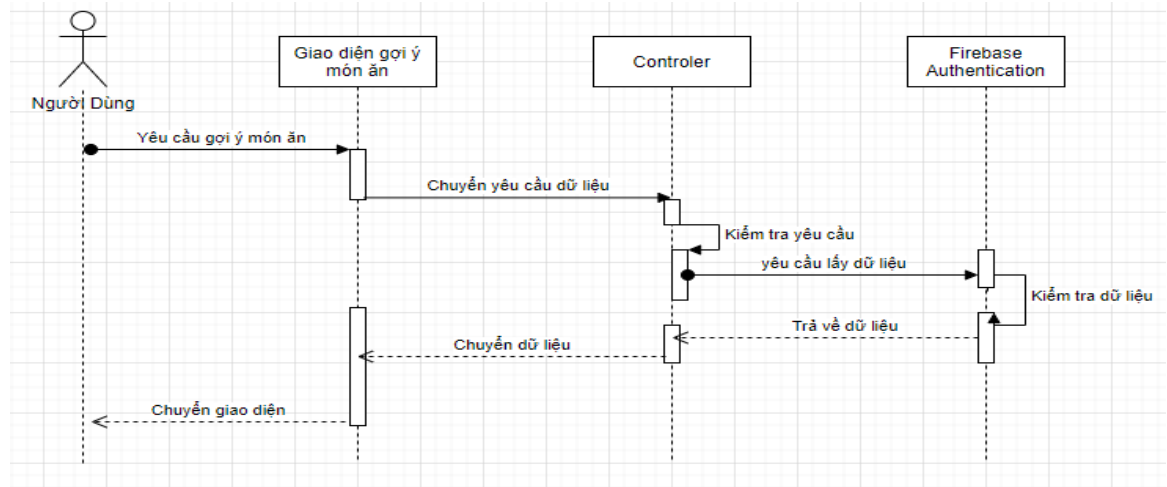
- Biểu đồ tuần tự use case xem lịch sử lượng calo biểu hiện yêu cầu xem lịch sử lượng calo của người dùng và hiển thị



Hình 3.11. Biểu đồ tuần tự use case xem lịch sử lượng calo

3.3.6.11. Biểu đồ tuần tự use case gọi ý món ăn

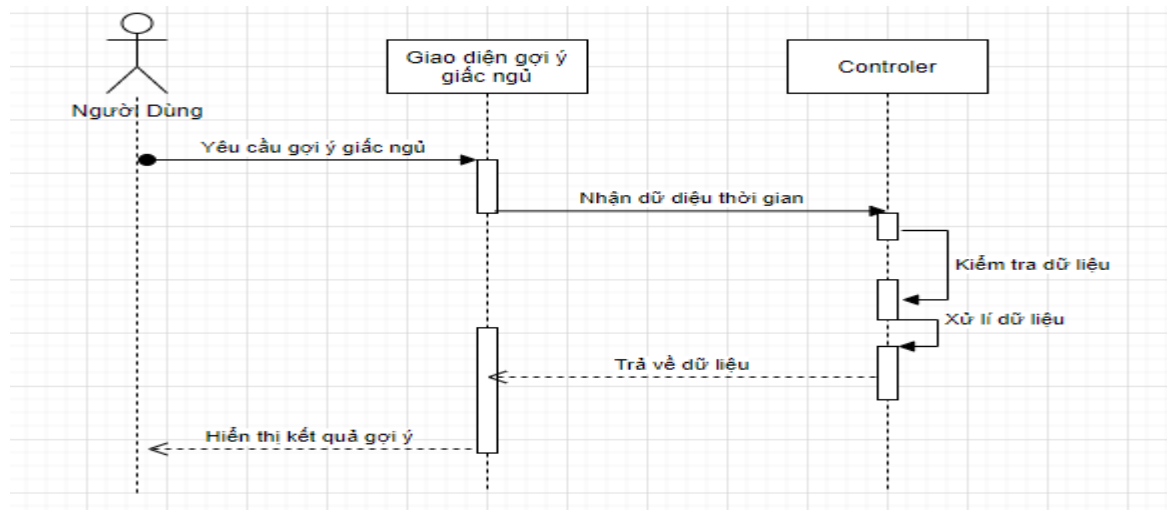
- Biểu đồ tuần tự use case gọi ý món ăn thể hiện được hướng đi của yêu cầu và luồng xử lý dữ liệu



Hình 3.12. Biểu đồ tuần tự use case gọi ý món ăn

3.3.6.12. Biểu đồ tuần tự use case xem gợi ý giấc ngủ

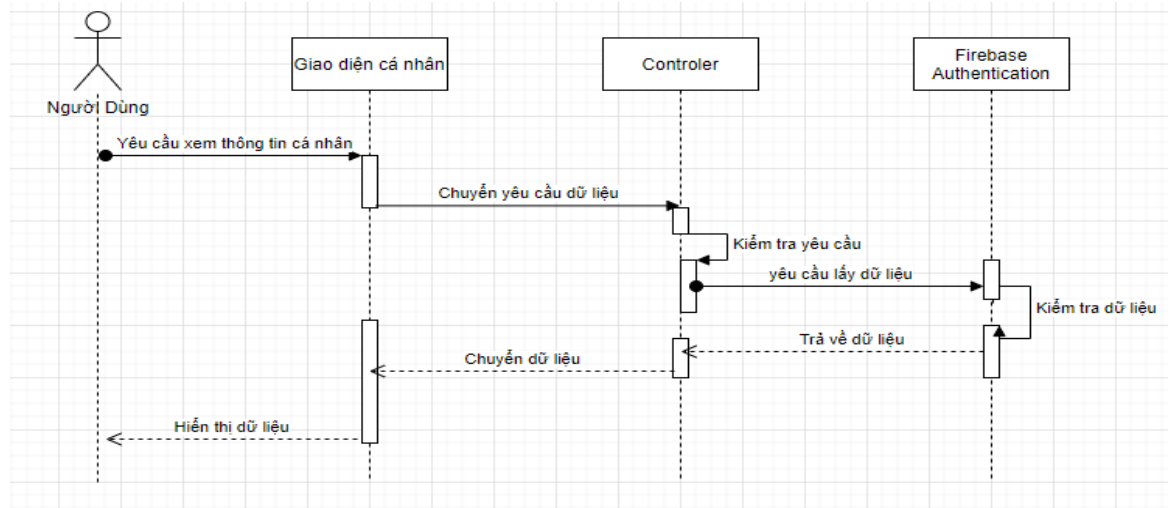
- Biểu đồ tuần tự use case xem gợi ý giấc ngủ hiển thị cách kiểm tra dữ liệu và hiển thị dữ liệu cho người dùng



Hình 3.13. Biểu đồ tuần tự use case xem gợi ý giấc ngủ

3.3.6.13. Biểu đồ tuần tự use case xem thông tin cá nhân

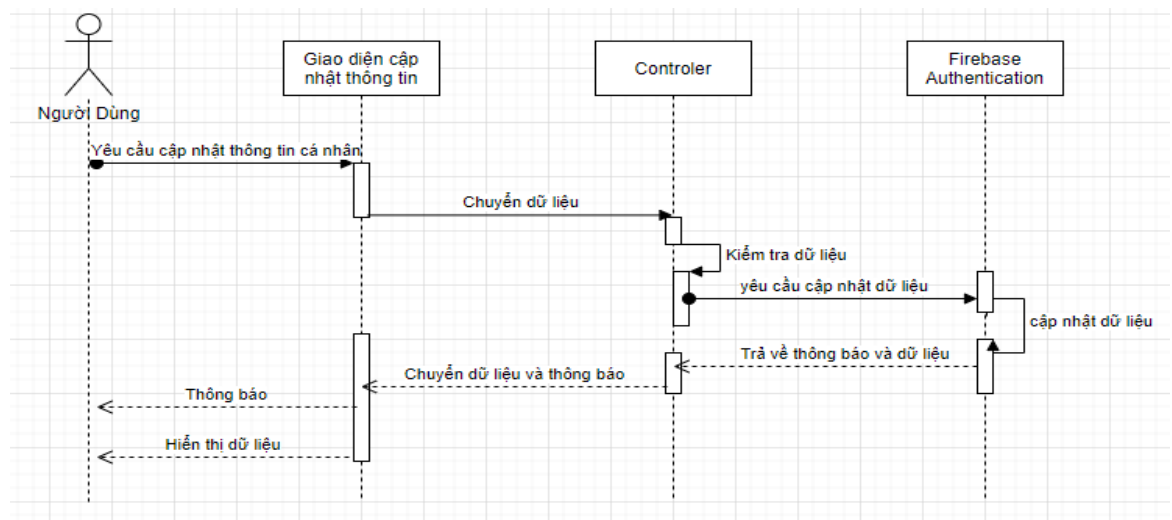
- Biểu đồ tuần tự use case xem thông tin cá nhân hiển thị cách mà ứng dụng xử lý yêu cầu người dùng và hiển thị thông tin



Hình 3.14. Biểu đồ tuần tự use case xem thông tin cá nhân

3.3.6.14. Biểu đồ tuần tự use case cập nhật thông tin cá nhân

- Biểu đồ tuần tự use case cập nhật thông tin cá nhân của người dùng sau khi người dùng yêu cầu và nhập thông tin mới.

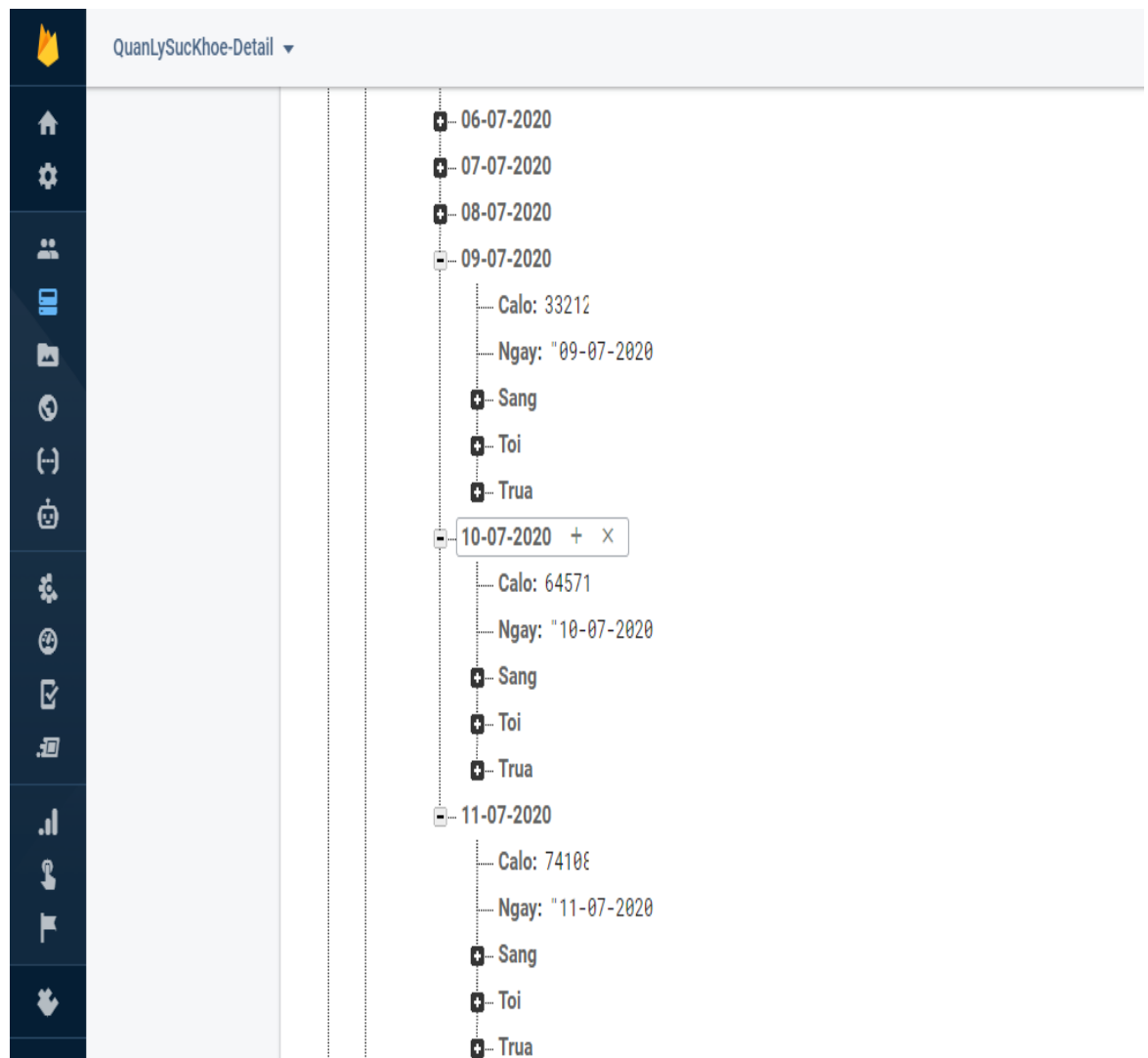


Hình 3.15. Biểu đồ tuần tự use case cập nhật thông tin cá nhân

3.4. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.4.1. Cơ sở dữ liệu biểu đồ

- Cơ sở dữ liệu biểu đồ bao gồm thông tin về lượng calo của ngày hôm đó. Nơi này chứa tất cả các ngày mà người dùng truy cập hệ thống.



Hình 3.16. Cơ sở dữ liệu biểu đồ

3.4.2. Cơ sở dữ liệu chi tiết lịch sử lượng calo

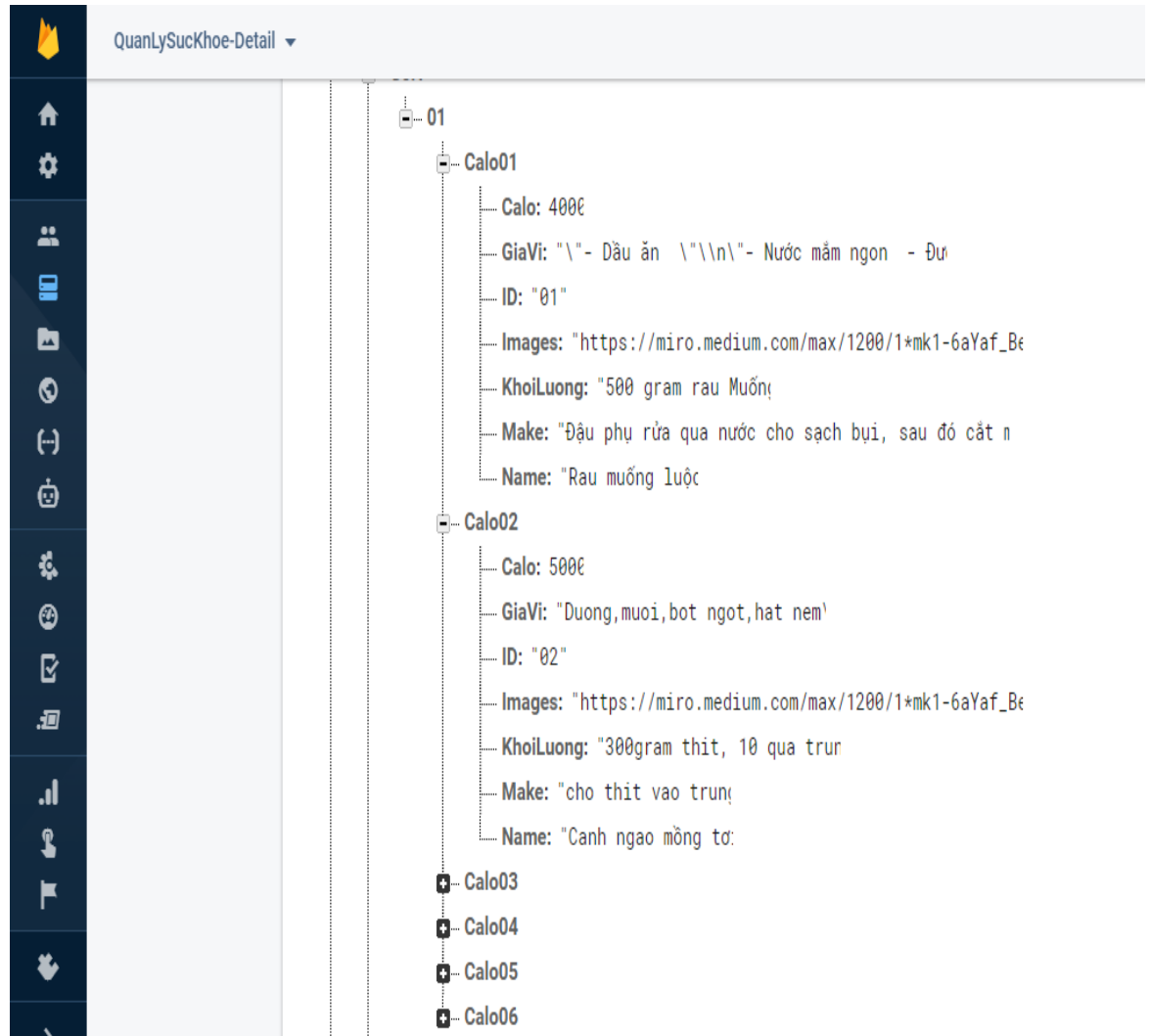
- Cơ sở dữ liệu chi tiết lịch sử lượng calo chứa đựng thông tin của từng ngày, lượng calo, các món ăn trong các buổi của họ.



Hình 3.17. Cơ sở dữ liệu chi tiết lịch sử lượng calo

3.4.3. Dữ liệu gợi ý món ăn

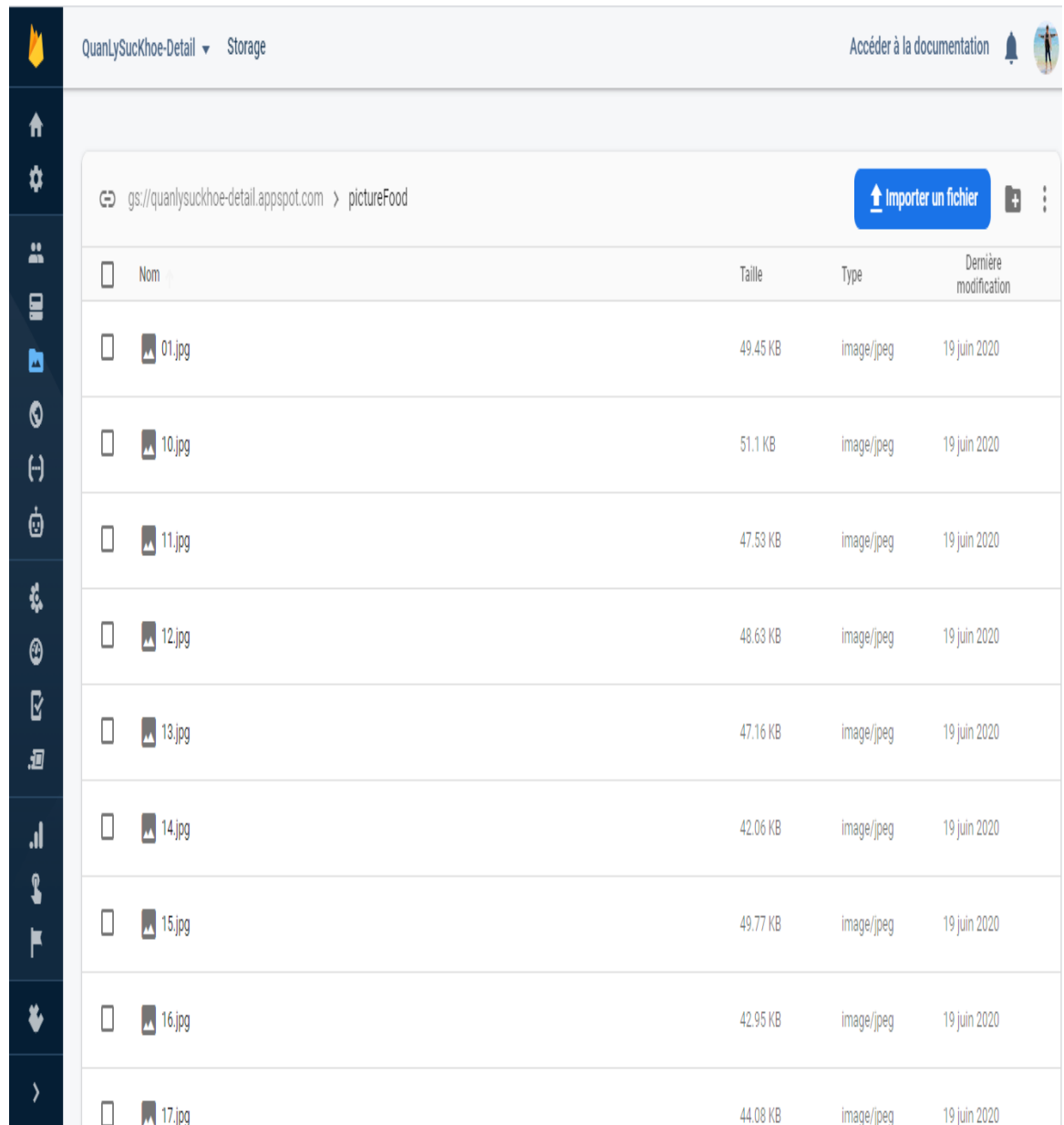
- Dữ liệu gợi ý món ăn bao gồm danh sách món ăn được gợi ý sẵn trên hệ thống, nó chứa các thông tin như lượng calo, hình ảnh, tên món ăn, cách làm, nguyên liệu,... để có thể hiển thị tốt nhất thông tin cho người dùng



Hình 3.18. Cơ sở dữ liệu gợi ý món ăn

3.4.4. Dữ liệu hình ảnh

- Dữ liệu hình ảnh chứa các hình ảnh của từng món ăn, chứa chi tiết cách chế biến, thông tin người dùng, khoảnh khắc người sử dụng.

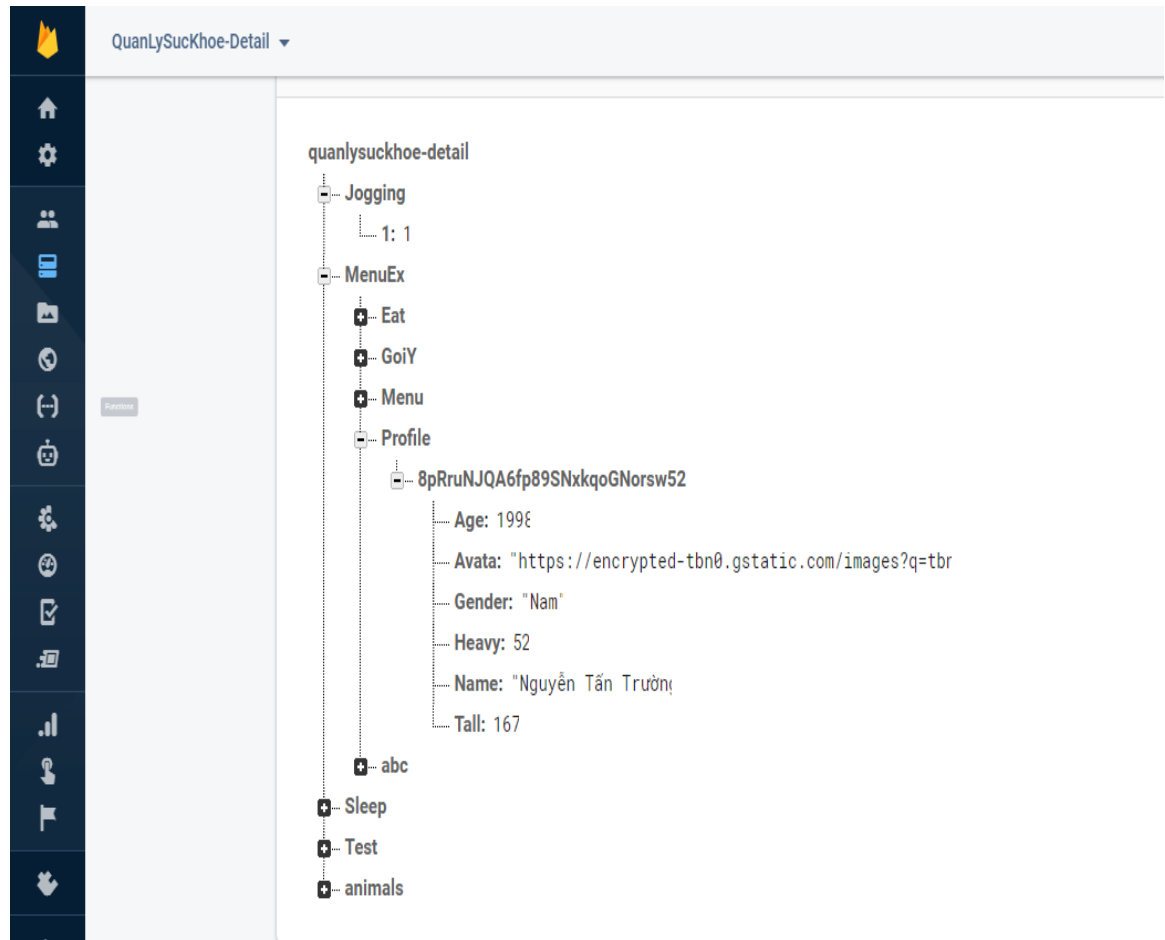


	Nom	Taille	Type	Dernière modification
	01.jpg	49.45 KB	image/jpeg	19 juin 2020
	10.jpg	51.1 KB	image/jpeg	19 juin 2020
	11.jpg	47.53 KB	image/jpeg	19 juin 2020
	12.jpg	48.63 KB	image/jpeg	19 juin 2020
	13.jpg	47.16 KB	image/jpeg	19 juin 2020
	14.jpg	42.06 KB	image/jpeg	19 juin 2020
	15.jpg	49.77 KB	image/jpeg	19 juin 2020
	16.jpg	42.95 KB	image/jpeg	19 juin 2020
	17.jpg	44.08 KB	image/jpeg	19 juin 2020

Hình 3.19. Cơ sở dữ liệu hình ảnh

3.4.5. Dữ liệu thông tin cá nhân

- Dữ liệu thông tin cá nhân bao gồm thông tin của người dùng gồm họ tên, năm sinh, chiều cao, cân nặng, hình ảnh



Hình 3.20. Cơ sở dữ liệu thông tin cá nhân

CHƯƠNG 4: GIAO DIỆN CHƯƠNG TRÌNH

4.1. GIAO DIỆN TÀI KHOẢN

4.1.1. Giao diện đăng nhập

- Hiển thị giao diện đăng nhập để người dùng truy cập vào ứng dụng và sử dụng các chức năng của ứng dụng

Vui lòng đăng nhập!

EMAIL

MẬT KHẨU

Đăng nhập

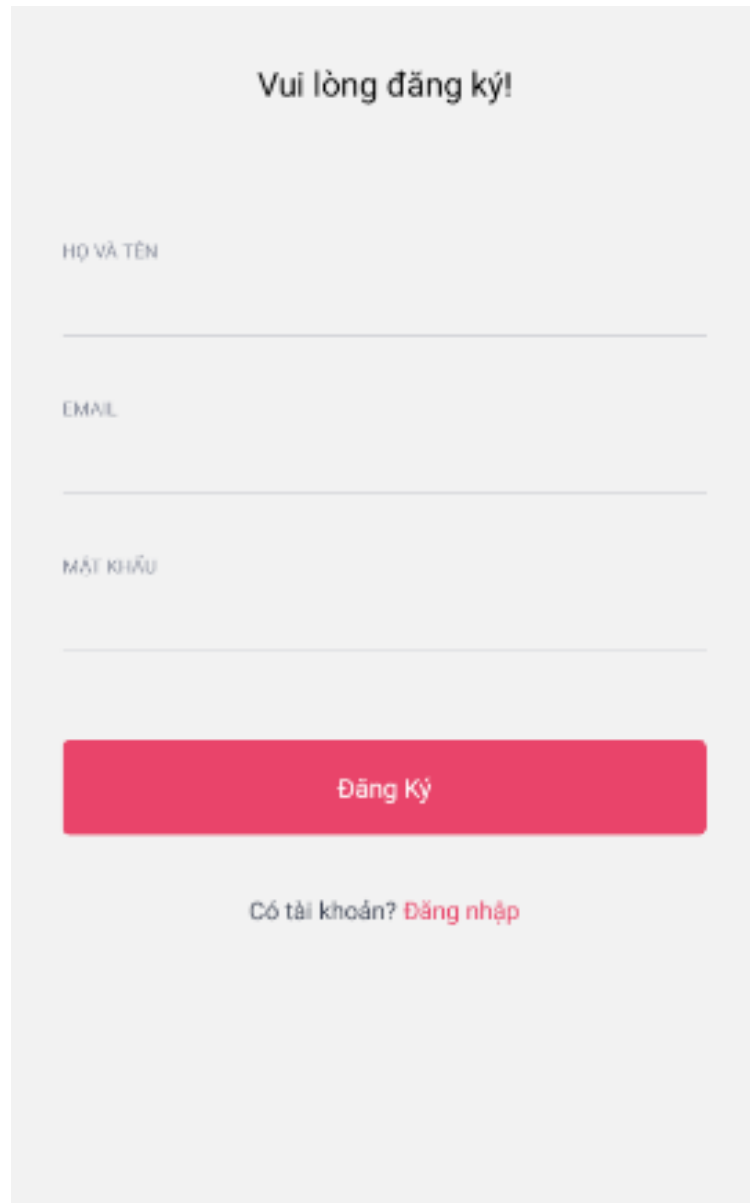
Chưa có tài khoản? [Đăng ký](#)

Quên mật khẩu? [Nhận lại mật khẩu](#)

Hình 4.1. Giao diện đăng nhập

4.1.2. Giao diện đăng ký

- Là giao diện đăng ký cho phép người dùng tự tạo cho mình một tài khoản và sử dụng cho việc đăng ký tài khoản truy cập ứng dụng

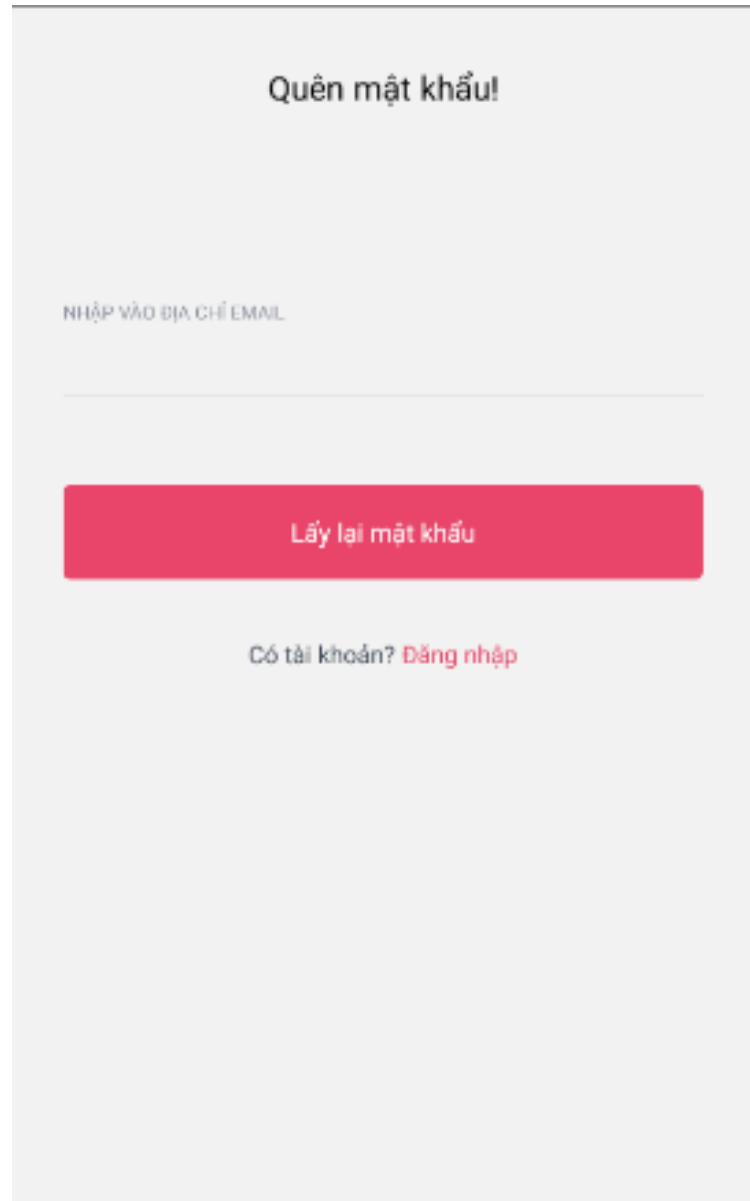


The image shows a registration form with a light gray background. At the top, the text "Vui lòng đăng ký!" is centered. Below it are three input fields: "HỌ VÀ TÊN", "EMAIL", and "MẬT KHẨU". Each field has a horizontal line for text entry. Below the input fields is a red button with the text "Đăng Ký". At the bottom, there is a link that says "Có tài khoản? Đăng nhập".

Hình 4.2. Giao diện đăng ký

4.1.3. Giao diện quên mật khẩu

- Giao diện cho phép người dùng nhập thông tin địa chỉ email và nhấp vào lấy lại mật khẩu để có thể tiếp tục truy cập vào ứng dụng



Quên mật khẩu!

NHẬP VÀO ĐỊA CHỈ EMAIL

Lấy lại mật khẩu

Có tài khoản? [Đăng nhập](#)

Hình 4.3. Giao diện quên mật khẩu

4.2. GIAO DIỆN QUẢN LÝ SỨC KHỎE

4.2.1. Giao diện thông tin cá nhân

- Giao diện thông tin cá nhân hiển thị thông tin của người sử dụng ứng dụng này bao gồm một vài thông tin quan trọng.

The screenshot displays a mobile application interface for personal information management. The main content area has a light gray background and contains the following text:

- Thông Tin Cá Nhân** (Personal Information)
- Họ Và Tên: Nguyễn Tấn Trường
- Năm Sinh: 1998
- Cân Nặng: 52
- Chiều Cao: 167

Below the text is a blue button labeled **ĐĂNG XUẤT** (Logout). At the bottom of the screen is a navigation bar with four items: **Cá Nhân** (Personal), **Biểu Đồ** (Charts), **Gợi Ý Món** (Suggested Dishes), and **Giấc Ngủ** (Sleep).

Hình 4.4. Giao diện thông tin cá nhân

4.2.2. Giao diện thêm, xóa, biểu đồ

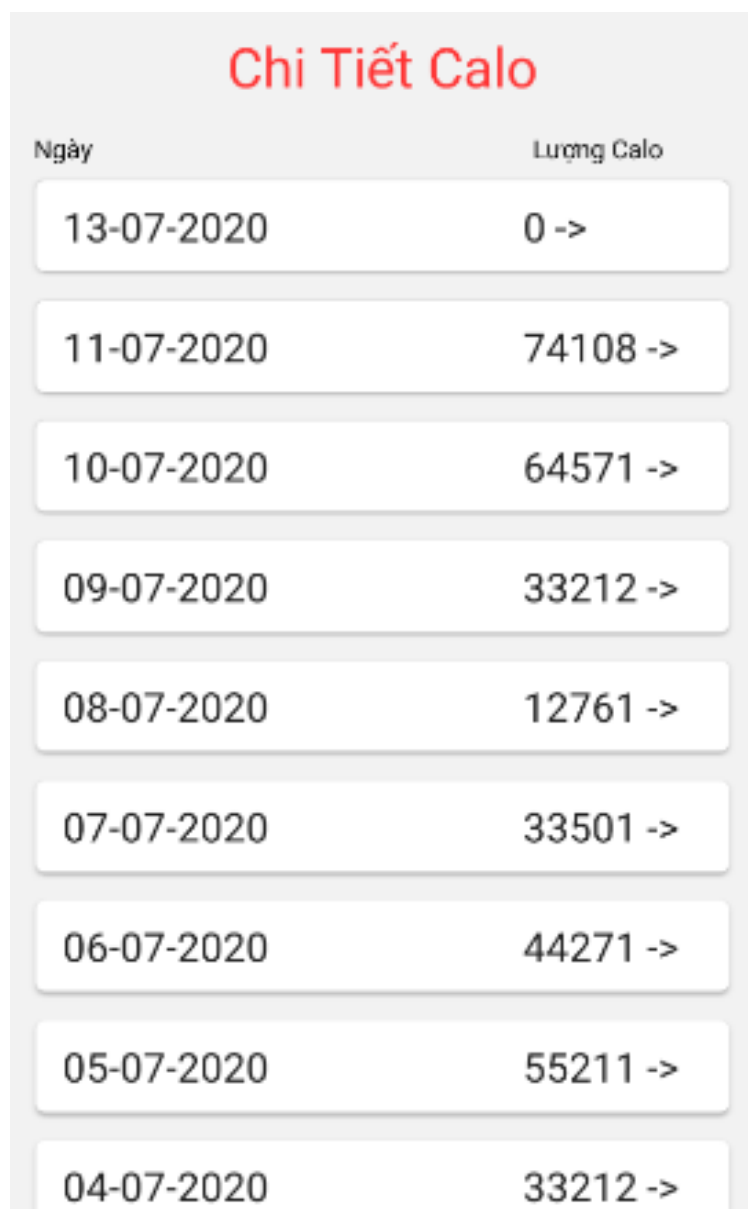
- Giao diện này bao gồm thêm món ăn vào các buổi, xóa món ăn, biểu đồ hiển thị thông tin của 7 ngày vừa qua để người dùng có cái nhìn trực quan hơn về lượng calo và sức khỏe của mình



Hình 4.5. Giao diện thêm, xóa, biểu đồ

4.2.3. Giao diện xem chi tiết lượng calo theo danh sách

- Giao diện xem chi tiết lượng calo theo danh sách các ngày mà người dùng truy cập vào hệ thống

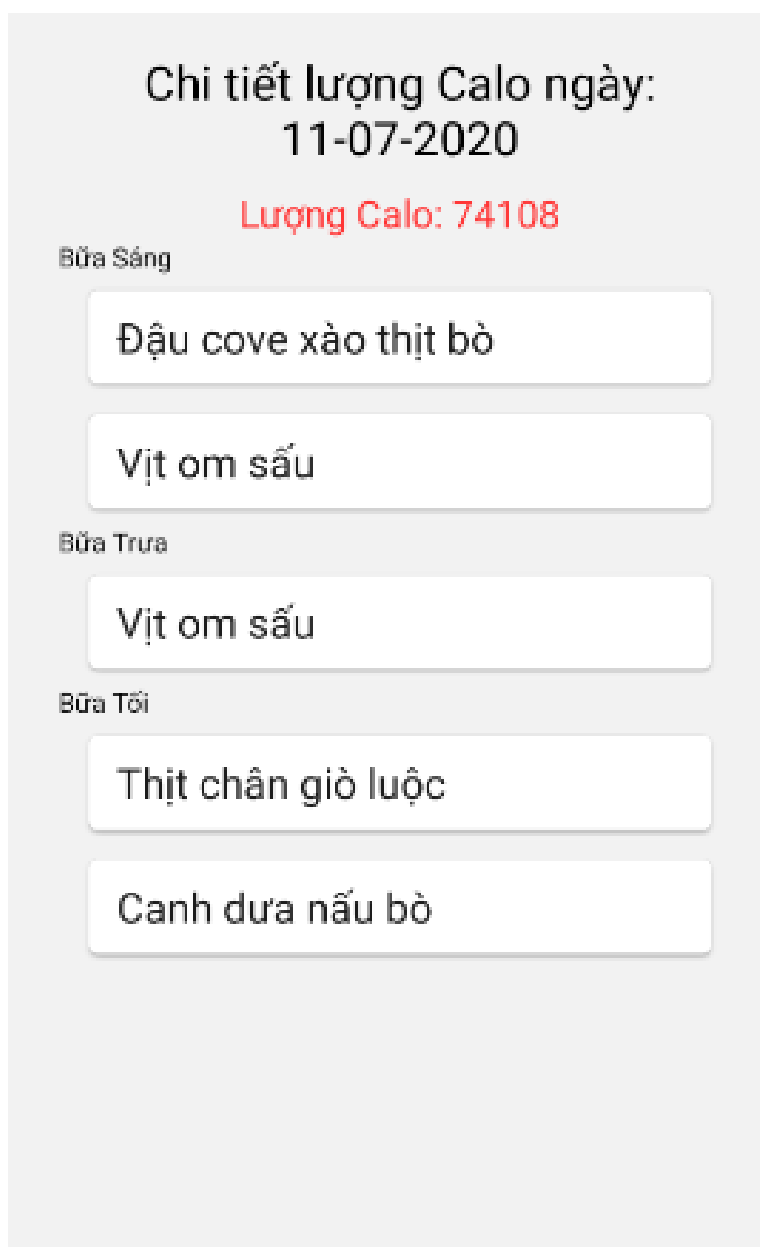


Ngày	Lượng Calo
13-07-2020	0 ->
11-07-2020	74108 ->
10-07-2020	64571 ->
09-07-2020	33212 ->
08-07-2020	12761 ->
07-07-2020	33501 ->
06-07-2020	44271 ->
05-07-2020	55211 ->
04-07-2020	33212 ->

Hình 4.6. Giao diện xem chi tiết lượng calo theo danh sách

4.2.4. Giao diện xem chi tiết lịch sử lượng calo

- Giao diện xem chi tiết lịch sử lượng calo dùng để người dùng có thể xem rõ được chi tiết người sử dụng bao gồm các món ăn theo các buổi, lượng calo,...



Chi tiết lượng Calo ngày:
11-07-2020

Lượng Calo: 74108

Bữa Sáng

Đậu cove xào thịt bò

Vịt om sấu

Bữa Trưa

Vịt om sấu

Bữa Tối

Thịt chân giò luộc

Canh dưa nấu bò

Hình 4.7. Giao diện xem chi tiết lịch sử lượng calo

4.2.5. Giao diện gợi ý món ăn

- Giao diện gợi ý món ăn dùng để hiển thị các món ăn cho người sử dụng có thể tham khảo và chế biến món ăn theo công thức.



Hình 4.8. Giao diện gợi ý món ăn

4.2.6. Giao diện cách chế biến món ăn

- Giao diện cách chế biến món ăn hiển thị được thông tin của món ăn bao gồm: Tên món ăn, lượng calo của món ăn, cách chế biến, nguyên liệu, gia vị dành cho món ăn



Hình 4.9. Giao diện cách chế biến món ăn

4.2.7. Giao diện gợi ý giấc ngủ

- Giao diện gợi ý giấc ngủ nhận vào thông tin người dùng xác định thời gian mình ngủ và thực hiện gợi ý cho người dùng chọn khoảng thời gian ngủ thích hợp với bản thân mình



Hình 4.10. Giao diện gợi ý giấc ngủ

CHƯƠNG 5: KẾT LUẬN

5.1. ƯU ĐIỂM CỦA ĐỀ TÀI

- Ứng dụng gọn nhẹ, dễ dàng sử dụng
- Có thể quản lý được lượng calo hằng ngày
- Có biểu đồ thống kê 7 ngày trực quan cho người dùng dễ nắm rõ
- Có khả năng chứa được nguồn dữ liệu lớn, truy xuất nhanh
- Ứng dụng có khả năng mở rộng để thêm nhiều chức năng khác tích hợp vào
- Ứng dụng không sử dụng Sever nên dễ dàng truy xuất dữ liệu nhanh

5.2. NHƯỢC ĐIỂM CỦA ĐỀ TÀI

- Chưa sử dụng được hết nguồn dữ liệu bên ngoài
- Số liệu thống kê chỉ mang tính tương đối cho người sử dụng
- Chỉ nằm ở mức là các gợi ý cho người dùng sử dụng theo
- Nguồn database còn ít cho ứng dụng

5.3. HƯỚNG PHÁT TRIỂN ĐỀ TÀI

- Khắc phục được các nhược điểm
- Sử dụng Firebase Analytics trong việc phân tích nhu cầu và thị hiếu người sử dụng
- Tích hợp thêm các chức năng để quản lý được sức khỏe tốt hơn

TÀI LIỆU THAM KHẢO

- [1]. <https://viblo.asia/p/xay-dung-ung-dung-voi-nodejs-amOG81Jbvz8P>
- [2]. <https://reactnative.dev/docs/components-and-apis>
- [3]. <http://www.dotplays.com/tich-hop-firebase-react-native-phan-2/>
- [4]. <https://medium.com/react-native-custom-alert/react-native-custom-alert-c85514311cc4>
- [5]. <https://pasgo.vn/blog/goi-y-mam-com-hang-ngay-khoi-phai-vat-oc-nghi-mon-an-1398>
- [6]. <https://pasgo.vn/blog/thuc-don-hang-ngay-cho-1-thang-day-du-dinh-duong-1306>
- [7]. <https://reactnavigation.org/docs/navigation-prop/>
- [8]. <https://www.w3schools.com/>
- [9]. <https://firebase.google.com/docs/database/>
- [10]. <https://stackoverflow.com/>