**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CHUYÊN MÔN**

**ĐỀ TÀI:**

**Manage body condition android app**

Giảng viên hướng dẫn: **Thầy Nguyễn Lê Minh**

Sinh viên thực hiện:

Tô Võ Như Quỳnh - 5951071089

CQ.59.CNTT

TP. Hồ Chí Minh, ngày 12 tháng 6 năm 2021

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CHUYÊN MÔN**

**ĐỀ TÀI:**

**Manage body condition android app**

Giảng viên hướng dẫn: **Thầy Nguyễn Lê Minh**

Sinh viên thực hiện:

Tô Võ Như Quỳnh - 5951071089

CQ.59.CNTT

TP. Hồ Chí Minh, ngày 12 tháng 6 năm 2021

**LỜI CẢM ƠN**

Em xin được chân thành cảm ơn thầy Nguyễn Lê Minh và các giảng viên trong bộ môn công nghệ thông tin đã hỗ trợ em hoàn thành xong bài báo cáo này, cảm ơn các thầy cô đã cung cấp những kiến thức, tài liệu bổ ích và sự chỉ dạy tận tình để em có thể hoàn thành nhiệm vụ được giao đối với phần thực tập chuyên môn này.

Trong quá trình thực hành và hoàn thành báo cáo của môn học sẽ không tránh khỏi những sai sót, rất mong các thầy cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không tránh khỏi thiếu sót, vì vậy em rất mong nhận được lời góp ý của thầy cô để em có thể hoàn thành tốt những bài báo cáo sắp tới.

Em xin chân thành cảm ơn!

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

|  |
| --- |
| ***TP. Hồ Chí Minh, ngày 12 tháng 6 năm 2021***  **Giáo viên hướng dẫn**  **Nguyễn Lê Minh** |

**Lí do chọn đề tài**

Trong cơn sốt dịch COVID-19 lan rộng ở khắp nơi trên thế giới và nguồn vác xin hạn hẹp, chúng ta nên chủ động tăng cường sức khỏe để chóng chọi lại dịch bệnh. Nhưng không phải ai cũng có thể tự mình lập ra nhưng chế độ ăn uống và tập luyện một cách logic và hiệu quả.

Là một người rất quan tâm đến sức khỏe và thể trạng của cơ thể tôi rất muốn chia sẻ cho mọi người cách bảo vệ sức khỏe và giữ gìn thể trạng của mình. Trong thời đại công nghệ phát triển hiện nay việc chia sẻ kiến thức cũng như các kinh nghiệm trở nên rất dễ dàng, đa số chúng ta ai cũng sẽ có cho mình một chiếc điện thoại thông minh. Nên tôi đã xây dựng nên một ứng dụng trên điện thoại để giúp mọi người không phải đau đầu logic về nhưng gì phải làm để cũng cố sức khỏe cho bản thân.

Chung tay vì sức khỏe của cộng đồng đẩy lùi dịch bệnh COVID-19.

Mục lục:

[Phần I: Tổng quan về đề tài và cơ sở lý thuyết 5](#_Toc75085908)

[I, Tổng quan về đề tài 5](#_Toc75085909)

[1, Giới thiệu . 5](#_Toc75085910)

[2, Đối tượng sử dụng 5](#_Toc75085911)

[II, Cơ sở lý thuyết 6](#_Toc75085912)

[1, Nội dung chuyên môn 6](#_Toc75085913)

[2, Các công nghệ và công cụ 9](#_Toc75085915)

[Phần II: Hệ thống cơ sở dữ liệu 14](#_Toc75085918)

[I, Mô tả 14](#_Toc75085919)

[II, Xây dựng cơ sở dữ liệu 16](#_Toc75085920)

[1, Sơ đồ ERD 16](#_Toc75085921)

[2. Mô hình quan hệ 17](#_Toc75085923)

[3, Bảng dữ liệu 17](#_Toc75085924)

[4, Sơ đồ phân rã chức năng 20](#_Toc75085934)

[Phần III: Giao diện và chức năng của hệ thống 21](#_Toc75085936)

[1, Các giao diện và ứng dụng từng giao diện. 21](#_Toc75085937)

[2, Kết luận: 32](#_Toc75085950)

Mục lục hình ảnh:

[Hình 1: Sơ đồ ERD 16](#_Toc75085922)

[Hình 2: Bảng user 17](#_Toc75085925)

[Hình 3: Bảng InfoUser 17](#_Toc75085926)

[Hình 4: Bảng Solutions 18](#_Toc75085927)

[Hình 5: Bảng Diets 18](#_Toc75085928)

[Hình 6: Bảng DietDetails 18](#_Toc75085929)

[Hình 7: Bảng ExerciseRegimes 19](#_Toc75085930)

[Hình 8: Bảng ExerciseRegimesDetails 19](#_Toc75085931)

[Hình 9: Bảng Exercise 19](#_Toc75085932)

[Hình 10: Bảng MonAn 19](#_Toc75085933)

[Hình 11: Sơ đồ phân rã chức năng 20](#_Toc75085935)

[Hình 11: Giao diện Welcome 21](#_Toc75085938)

[Hình 12: Giao diện đăng ký tài khoảng 22](#_Toc75085939)

[Hình 13: Giao diện lời khuyên và các sự lựa chọn 23](#_Toc75085940)

[Hình 14: Giao diện đăng nhập 24](#_Toc75085941)

[Hình 15: Giao diện thông tin người dùng 25](#_Toc75085942)

[Hình 16: Giao diện chế độ ăn 26](#_Toc75085943)

[Hình 17: Giao diện chế độ tập luyện 27](#_Toc75085944)

[Hình 18: Giao diện biểu đồ theo dõi thể trạng 28](#_Toc75085945)

[Hình 19: Giao diện những chức năng nhỏ 29](#_Toc75085946)

[Hình 20: Giao diện thay đổi ảnh đại diện 30](#_Toc75085947)

[Hình 21: Giao diện đổi mật khẩu 31](#_Toc75085948)

[Hình 22: Giao diện chỉnh sửa thông tin người dùng 31](#_Toc75085949)

# Phần I: Tổng quan về đề tài và cơ sở lý thuyết

## I, Tổng quan về đề tài

### 1, Giới thiệu .

Manage Body Condition Là một App ứng dụng trên điện thoại, nó giúp người dùng kiểm soát thể trạng cơ thể. Với những thông tin được người dùng nhập vào App sẽ đưa ra lời khuyên, những sự lựa chọn về chế độ ăn uống và luyện tập để người dùng, ứng dụng sẽ theo dõi thể trạng của bạn qua từng tuần và vẽ ra biểu đồ theo dõi để bạn biết được thể trạng của bạn đang có sự thây đổi theo chiều hướng nào, giúp bạn có 1 thể trạng cân đối và khỏe mạnh nhất.

### 2, Đối tượng sử dụng

Bất kì đối tượng nào cũng có thể sử dụng App chỉ cần có cho mình 1 chiếc smartphone.

Một App có thể dùng được nhiều đối tượng.

## II, Cơ sở lý thuyết

### 1, Nội dung chuyên môn

* Lập trình hướng đối tượng

Lập trình hướng đối tượng (tiếng Anh: Object-oriented programming, viết tắt: OOP) là một [mẫu hình lập trình](https://vi.wikipedia.org/wiki/M%E1%BA%ABu_h%C3%ACnh_l%E1%BA%ADp_tr%C3%ACnh" \o "Mẫu hình lập trình) dựa trên khái niệm "[công nghệ đối tượng](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%91i_t%C6%B0%E1%BB%A3ng_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)" \o "Đối tượng (khoa học máy tính))", mà trong đó, đối tượng chứa đựng các [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u" \o "Dữ liệu), trên các [trường](https://vi.wikipedia.org/wiki/Tr%C6%B0%E1%BB%9Dng_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)" \o "Trường (khoa học máy tính)), thường được gọi là các *thuộc tính;* và mã nguồn, được tổ chức thành các *[phương thức](https://vi.wikipedia.org/wiki/Ph%C6%B0%C6%A1ng_th%E1%BB%A9c_(l%E1%BA%ADp_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh)" \o "Phương thức (lập trình máy tính)).* Phương thức giúp cho đối tượng có thể truy xuất và hiệu chỉnh các trường dữ liệu của đối tượng khác, mà đối tượng hiện tại có tương tác (đối tượng được hỗ trợ các phương thức "[this](https://vi.wikipedia.org/w/index.php?title=This_(l%E1%BA%ADp_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh)&action=edit&redlink=1)" hoặc "self"). Trong lập trình hướng đối tượng, chương trình máy tính được thiết kế bằng cách tách nó ra khỏi phạm vi các đối tượng tương tác với nhau.[[1]](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng#cite_note-1)[[2]](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng#cite_note-2) Ngôn ngữ lập trình hướng đối tượng khá đa dạng, phần lớn là các [ngôn ngữ lập trình theo lớp](https://vi.wikipedia.org/wiki/L%E1%BB%9Bp_(l%E1%BA%ADp_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh)" \o "Lớp (lập trình máy tính)), nghĩa là các đối tượng trong các ngôn ngữ này được xem như [thực thể](https://vi.wikipedia.org/wiki/Th%E1%BB%B1c_th%E1%BB%83_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)" \o "Thực thể (khoa học máy tính)) của một [lớp](https://vi.wikipedia.org/wiki/L%E1%BB%9Bp_(l%E1%BA%ADp_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh)" \o "Lớp (lập trình máy tính)), được dùng để định nghĩa một [kiểu dữ liệu](https://vi.wikipedia.org/wiki/Ki%E1%BB%83u_d%E1%BB%AF_li%E1%BB%87u" \o ").

OOP được xem là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép [lập trình viên](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_vi%C3%AAn" \o "Lập trình viên) tập trung vào các đối tượng phần mềm ở bậc cao hơn. Ngoài ra, nhiều người còn cho rằng OOP dễ tiếp thu hơn cho những người mới học về [lập trình](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh" \o "Lập trình máy tính) hơn là các phương pháp trước đó. Một cách giản lược, đây là khái niệm và là một nỗ lực nhằm giảm nhẹ các thao tác viết mã cho người lập trình, cho phép họ tạo ra các [ứng dụng](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_%E1%BB%A9ng_d%E1%BB%A5ng" \o "Phần mềm ứng dụng) mà các yếu tố bên ngoài có thể tương tác với các chương trình đó giống như là tương tác với các đối tượng vật lý.

Những đối tượng trong một ngôn ngữ OOP là các kết hợp giữa mã và dữ liệu mà chúng được nhìn nhận như là một đơn vị duy nhất. Mỗi đối tượng có một tên riêng biệt và tất cả các tham chiếu đến đối tượng đó được tiến hành qua tên của nó. Như vậy, mỗi đối tượng có khả năng nhận vào các thông báo, xử lý dữ liệu (bên trong của nó), và gửi ra hay trả lời đến các đối tượng khác hay đến môi trường.

Đa phần các ngôn ngữ lập trình thông dụng nhất hiện nay (như C++, Delphi, Java, Python etc.) là các ngôn ngữ lập trình đa mẫu hình và đều hỗ trợ lập trình hướng đối tượng ở nhiều mức độ khác nhau, thường được kết hợp với [lập trình mệnh lệnh](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_m%E1%BB%87nh_l%E1%BB%87nh" \o "Lập trình mệnh lệnh), [lập trình thủ tục](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_th%E1%BB%A7_t%E1%BB%A5c" \o "Lập trình thủ tục). Các ngôn ngữ lập trình hướng đối tượng đáng chú ý gồm có [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [C++](https://vi.wikipedia.org/wiki/C%2B%2B), [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng), [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [PHP](https://vi.wikipedia.org/wiki/PHP), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Perl](https://vi.wikipedia.org/wiki/Perl), [Object Pascal](https://vi.wikipedia.org/wiki/Object_Pascal), [Objective-C](https://vi.wikipedia.org/w/index.php?title=Objective-C&action=edit&redlink=1), [Dart](https://vi.wikipedia.org/wiki/Dart_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Swift](https://vi.wikipedia.org/wiki/Swift_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Scala](https://vi.wikipedia.org/wiki/Scala), [Common Lisp](https://vi.wikipedia.org/wiki/Lisp), và [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk).

* Phân tích thiết kế cơ sở dữ liệu

Tất cả mọi hệ thống đều phải sử dụng một cơ sở dữ liệu của mình, đó có thể là một cơ sở dữ liệu đã có hoặc một cơ sở dữ liệu được xây dựng mới. Cũng có những hệ thống sử dụng cả cơ sở dữ liệu cũ và mới. Việc phân tích và thiết kế cơ sở dữ liệu cho một hệ thống có thể tiến hành đồng thời với việc phân tích và thiết kế hệ thống hoặc có thể tiến hành riêng. Vấn đề đặt ra là cần xây dựng một cơ sở dữ liệu giảm được tối đa sự dư thừa dữ liệu đồng thời phải dễ khôi phục và bảo trì.

Cơ sở dữ liệu : Cơ sở dữ liệu máy tính là một kho chứa một bộ sưu tập

có tổ chức các file dữ liệu, các bản ghi và các trường.

Hệ quản trị cơ sở dữ liệu là một phần mềm  điều khiển mọi truy nhập đối với cơ sở dữ liệu.

Các bước tiến hành phân tích và thiết kế CSDL

a. Phân tích CSDL

  Bước phân tích CSDL độc lập với các hệ quản trị CSDL, bước này thực hiện các công việc sau:

+  Xác định các yêu cầu về dữ liệu: Phân tích các yêu cầu dữ liệu của hệ thống để xác định các yêu cầu về dữ liệu.

+  Mô hình hoá dữ liệu: Xây dựng mô hình thực thể liên kết biểu diễn các yêu cầu về dữ liệu.

b. Thiết kế CSDL quan hệ

Thiết kế CSDL bao gồm:

+  Thiết kế logic CSDL: độc lập với một hệ quản trị CSDL.

•  Xác định các quan hệ: Chuyển từ mô hình thực thể liên kết sang

mô hình quan hệ.

•  Chuẩn hoá các quan hệ: chuẩn hoá các quan hệ về dạng chẩn ít

nhất là chuẩn 3 (3NF)

+  Thiết kế vật lý CSDL: dựa trên một hệ quản trị CSDL cụ thể.

•  Xây dựng các bảng  trong CSDL quan hệ:  quyết  định cấu  trúc

thực tế của các bảng lưu trữ trong mô hình quan hệ.

•  Hỗ trợ các cài  đặt vật lý trong CSDL: cài   đặt chi tiết trong

HQTCSDL lựa chọn.

* Lập trình sử dụng API

**API** là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của **Application Programming Interface** – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

## **Đặc điểm nổi bật của API**

* API sử dụng mã nguồn mở, dùng được với mọi client hỗ trợ XML, JSON.
* API có khả năng đáp ứng đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma…. Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS.
* Mô hình web API dùng để hỗ trợ MVC như: unit test, injection, ioc container, model binder, action result, filter, routing, controller. Ngoài ra, nó cũng hỗ trợ RESTful đầy đủ các phương thức như: GET, POST, PUT, DELETE các dữ liệu.
* Được đánh giá là một trong những kiểu kiến trúc hỗ trợ tốt nhất với các thiết bị có lượng băng thông bị giới hạn như smartphone, tablet…
* Lập trình android

**Hệ điều hành Android** đang chiếm hơn 80% thị trường thiết bị di động hiện đại ngày nay. Nhu cầu sử dụng ứng dụng di động cũng ngày càng cao. Nhu cầu việc làm về lập trình trên hệ điều hành mobile phổ biến nhất thế giới này cũng vì thế mà tăng mạnh do tính mở và dễ tiếp cận của nó.

**Android**là hệ điều hành mã nguồn mở, dựa trên **Linux Kerne**l, dành cho các thiết bị di động nói chung (điện thoại, máy tính bảng, đồng hồ thông minh, máy nghe nhạc,…).

Có nghĩa là Android không chỉ giới hạn trong phạm vi một hệ điều hành cho điện thoại! Nó có thể được nhà sản xuất cài đặt lên đồng hồ, máy nghe nhạc, thiết bị định vị GPS, thậm chí là ô tô (các thiết bị Android Auto).

Kiến trúc :

* Tầng Applications: Là tầng chứa các ứng dụng Danh bạ, Gọi điện, Trình duyệt, Nghe nhạc,… các ứng dụng này thường mua máy về chúng ta đã có sẵn rồi.
* Tầng Framework:Là tầng chứa các API để làm việc với hệ điều hành như lấy thông tin danh bạ, quản lý các Activity (Activity là gì thì giờ chúng ta chưa cần quan tâm, các bài sau sẽ giải thích kĩ), quản lý địa điểm, quản lý các View (cũng chưa cần quan tâm).
* Tầng Libraries: Chứa các thư viện, API gần như là cốt lõi của Android, bao gồm bộ quản lý bề mặt cảm ứng (Surface Manager), OpenGL (phục vụ cho việc dựng đồ họa phức tạp),…
* Tầng Android Runtime:Chứa các thư viện lõi của Android và máy ảo Dalvik Virtual Machine (từ Android 4 trở lên chúng ta có thêm máy ảo ART).
* Tầng Kernel:Là nhân lõi của hệ điều hành, chứa các tập lệnh, driver giao tiếp giữa phần cứng và phần mềm của Android.

### 2, Các công nghệ và công cụ

* Java(Android Studio)

Tháng 5 năm 2013, Google công bố Android Studio, một môi trường phát triển ứng dụng tích hợp (IDE) dành riêng cho Android, mã nguồn mở, dựa trên IDE**Java IntelliJ** của hãng **JetBrains**(đối thủ với Eclipse và Netbeans, vốn khá quen thuộc với dân lập trình Java).

**Android Studio** chạy trên Windows, Mac và Linux, nhằm thay thế cho **Eclipse Android Development Tool**(ADT) vốn được sử dụng làm IDE chính trong các năm trước đó.

**Ưu điểm:**  
•    Dễ học hơn c ++  
•    Dễ đọc mã hơn c ++  
•    Phát triển nhanh hơn và có khả năng ít lỗi hơn (do quản lý bộ nhớ ít hơn) so với c ++.  
•    Hầu hết các nền tảng chéo kể từ khi nó được thiết kế cho nó và Sun tạo ra JRE và JDK cho hầu hết các nền tảng  
•    Tất cả mọi thứ được truyền bằng tham chiếu ngoại trừ các loại giá trị, theo mặc định  
•    Công cụ thu gom rác dọn dẹp đồ vật một khi chúng không còn được sử dụng, vì vậy bạn không phải tự mình theo dõi mọi thứ  
•    Rõ ràng java 6 chạy (một số?) Công cụ nhanh hơn một chút so với c ++ (mã kiểm tra được chuyển từ c ++ chạy nhanh hơn mặc dù phải làm những việc có vẻ lãng phí!)  
•    Chương trình được biên dịch và tối ưu hóa cho nền tảng, nếu cần?  
•    Dễ dàng tạo nhiều luồng và bạn có thể khai báo các chức năng được đồng bộ hóa để chỉ một luồng có thể có trong một luồng (IIRC)  
•    Bạn có thể sử dụng Wait () và notifyAll () và thông báo () để tạm dừng các luồng để chờ tín hiệu và đánh thức chúng lại.  
•    Các IDE miễn phí khác nhau có sẵn trên tất cả các nền tảng, chẳng hạn như Eclipse  
•    Nếu chương trình gặp sự cố, nó sẽ báo cho người dùng biết mã trong đó ở đâu và tại sao, hiển thị dấu vết ngăn xếp bao gồm số dòng (nếu biểu tượng gỡ lỗi được bật hoặc một cái gì đó tương tự) hoặc có thể được đổ vào tệp nhật ký lỗi.  
•    Các chương trình có thể được đóng gói thành các tệp jar với các tệp dữ liệu khác nhau trong các tệp jar khác nhau và được người dùng truy cập bằng Java WebStart, sẽ tự động cập nhật chương trình và mọi tệp jar cần thiết khác lên phiên bản mới nhất, chỉ tải xuống các tệp jar đã thay đổi.  
•    Mạng dễ dàng hơn / nhiều nền tảng hơn c ++.  
•    Các tệp được đọc và ghi dưới dạng big endian trên tất cả các nền tảng (là thứ tự byte mạng) đảm bảo rằng giao tiếp giữa các chương trình java và tải tệp không bao giờ yêu cầu hoán đổi thứ tự byte (trừ khi nói chuyện với chương trình không phải java hoặc sử dụng tệp được viết bởi một ).

**Nhược điểm:**  
•    Sử dụng nhiều bộ nhớ hơn c ++ hoặc c #  
•    JNI thật đáng sợ, nhưng nếu chúng tôi là nền tảng chéo, có lẽ chúng tôi sẽ không sử dụng nó, (ngoại trừ công cụ 3d sẽ?)  
•    Trình thu gom rác sử dụng chu kỳ và bộ nhớ CPU - (nhưng hầu như không có, ít nhất là thời gian CPU)  
•    Không có số nguyên không dấu  
•    Không có con trỏ, không có tham số ref hoặc out (thay thế con trỏ trong c #). Để trả về nhiều thứ từ một hàm java, bạn có xu hướng phải trả về một đối tượng [] đang giữ từng thứ hoặc tạo một lớp chứa từng thứ và trả về một đối tượng của lớp đó.  
•    Một số thứ đã được đổi tên - const là cuối cùng trong java, ví dụ  
•    Java không cho phép người lập chỉ mục hoặc toán tử trên các đối tượng, ví dụ: nếu bạn sử dụng danh sách có thể thay đổi kích thước, v.v., bạn phải thực hiện list.get (index) thay vì list [index] và bạn không thể thực hiện someVector + someOtherVector.  
•    Java không cho phép bạn quyết định xem một loại nên là một giá trị hay một loại tham chiếu (xác định xem nó nên được truyền theo giá trị hay chuyển qua tham chiếu), không giống như c #.  
•    Các tệp Jar được sử dụng bởi java webstart phải được ký bằng chứng chỉ. Chứng chỉ ký mã có giá hàng trăm đô la mỗi năm hoặc bạn có thể tự tạo một ký tự và sử dụng nó, điều này sẽ khiến Java Webstart bật lên hộp thoại "Chứng nhận này không thể được xác minh" khi người dùng cố chạy chương trình. (Cách khác, không sử dụng Java Webstart và chỉ phân phối các tệp jar trực tiếp - yêu cầu người dùng tự kiểm tra các bản cập nhật.)  
•    Yêu cầu JRE được cài đặt để chạy chương trình, tải xuống vài giờ khi quay số.  
•    Phải sử dụng java 5 cho generic, 1.4 không có chúng. Java 6 mới nhất và đã được vài năm tuổi, nhưng - Apple phát hành phiên bản java của riêng họ cho người dùng Mac và vẫn chưa phát hành phiên bản java 6 "ổn định" cho người dùng.

* XML

XML (viết tắt từ tiếng Anh eXtensible Markup Language, "Ngôn ngữ Đánh dấu khả mở") là ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị, để tạo ra các ngôn ngữ đánh dấu khác. ... Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet.

* Ưu điểm của XML

Ưu điểm lớn nhất của XML là sự độc lập. XML được sử dụng để mô tả dữ liệu dưới dạng text, nên hầu hết các phần mềm hay các chương trình bình thường đều có thể đọc được chúng.

XML có thể đọc và phân tích nguồn dữ liệu khá dễ dàng nên nó được sử dụng với mục đích chính là trao đổi dữ liệu giữa các chương trình, các hệ thống khác nhau. XML được sử dụng cho Remote Procedure Calls nhằm phục vụ cho các dịch vụ của website.

* Nhược điểm của XML

Tỷ lệ sai sót khi sử dụng XML để truyền dữ liệu khoảng từ 5-7%. Con số này tuy không quá cao, nhưng trên thực tế, người ta vẫn cần cân nhắc trước khi sử dụng nó để trao đổi thông tin.

* Youtube API

 API YouTube, các nhà phát triển có thể thêm một loạt các chức năng và tính năng của YouTube có thể tạo ra giá trị to lớn.

Mặc dù quản lý và xử lý dữ liệu là một trong những khía cạnh quan trọng của quản lý kinh doanh, nhưng sự gia tăng theo cấp số nhân của dữ liệu đã khiến cho việc quản lý thông tin bằng các hệ thống CNTT truyền thống trở nên vô cùng khó khăn.

API giúp phá vỡ các rào cản giữa các hệ thống và đơn giản hóa các quy trình bằng cách nâng cao hiệu quả.

Hiện tại, API dữ liệu của YouTube có thể được sử dụng cho nhiều mục đích khác nhau bao gồm tìm kiếm nội dung, kiểm tra các yêu cầu được phép và trái phép, cũng như các tác vụ đơn giản khác như quản lý danh sách phát của bạn, tải video lên và cập nhật cài đặt kênh.

Mặc dù YouTube cung cấp dịch vụ này, nhưng nó không làm điều đó mà không có sự kiểm soát. Nó cung cấp cho mọi người dùng một khóa truy cập duy nhất được gọi là Khóa API để giúp họ kết nối với giao diện của trang web.

* SQLite(SQLite Studio)

SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine, không cần máy chủ, không cần cấu hình, khép kín và nhỏ gọn. Nó là một cơ sở dữ liệu, không cần cấu hình, có nghĩa là giống như các cơ sở dữ liệu khác mà bạn không cần phải cấu hình nó trong hệ thống của mình.

SQLite Engine không phải là một quy trình độc lập (standalone process) như các cơ sở dữ liệu khác, bạn có thể liên kết nó một cách tĩnh hoặc động tùy theo yêu cầu của bạn với ứng dụng của bạn. SQLiet truy cập trực tiếp các file lưu trữ (storage files) của nó.

## Ưu điểm của SQLite.

* SQLite không cần mô hình client – server để hoạt động.
* SQLite không cần phải cấu hình tức là bạn không cần phải cài đặt.
* Với SQLite database được lưu trữ trên một tập tin duy nhất.
* SQLite hỗ trợ hầu hết các tính năng của ngôn ngữ truy vấn SQL theo chuẩn SQL92.
* SQLite rất nhỏ gọn bản đầy đủ các tính năng nhỏ hơn 500kb, và có thể nhỏ hơn nếu lược bớt một số tính năng.
* Các thao tác dữ liệu trên SQLite chạy nhanh hơn so với các hệ quản trị cơ sở dữ liệu theo mô hình client – server.
* SQLite rất đơn giản và dễ dàng sử dụng.
* SQLite tuân thủ 4 tính chất ACID (là tính nguyên tổ (Atomic), tính nhất quán (Consistent), tính cô lập (Isolated), và tính bền vững (Durable) ).
* Với đặc tính nhỏ gọn, truy xuất dữ liệu nhanh SQLite thường được sử dụng để nhúng vào các dự án.

## Nhược điểm của SQLite

* Ngoài những ưu điểm đã kể ra ở trên SQLite cũng có một số mặt hạn chế nếu đem so sánh với các hệ quản trị khác.
* Do sử dụng cơ chế coarse-gained locking nên trong cùng một thời điểm SQLite có thể hỗ trợ nhiều người đọc dữ liệu, nhưng chỉ có 1 người có thể ghi dữ liệu.
* SQLite không phải là lựa chọn hoàn hảo để đáp ứng các nhu cầu xử lý trên một khối lượng dữ liệu lớn, phát sinh liên tục.

# Phần II: Hệ thống cơ sở dữ liệu

## I, Mô tả

Mỗi người dùng sẽ có 1 tài khoản (User) gồm tên tài khoản (usernames), mật khẩu (password) và có 1 hoặc nhiều InfoUser. Thông tin người dùng (InfoUser) bao gồm infoID, chiều cao (height), cân nặng (weight), vòng ngực (chest), vong eo (waist), vòng mông (hips), BMI(được tính từ chiều cao và cân nặng).

Từ những thông tin của người dùng (InfoUser) sau khi được đưa ra lời khuyên ta sẽ có những phương pháp(Solutions) để lựa chọn được quản lý bởi các thôn tin solutionID,tên sự lựa chon (solutionName).

Trong các phương pháp trên mỗi phương pháp bao gồm: một chế độ ăn(diet) sẽ có dietID, têm của chế độ ăn (dietName) bao gồm nhiều thông tin chế độ ăn(Dietdetail) chia theo ngày và một chế độ tập luyên(ExerciseRegimes) bao gồm những thông tin exerciseRegimeID, tên chế độ tập luyện (exerciseRegimeName) và nhiều bài tập theo từng ngày(day).

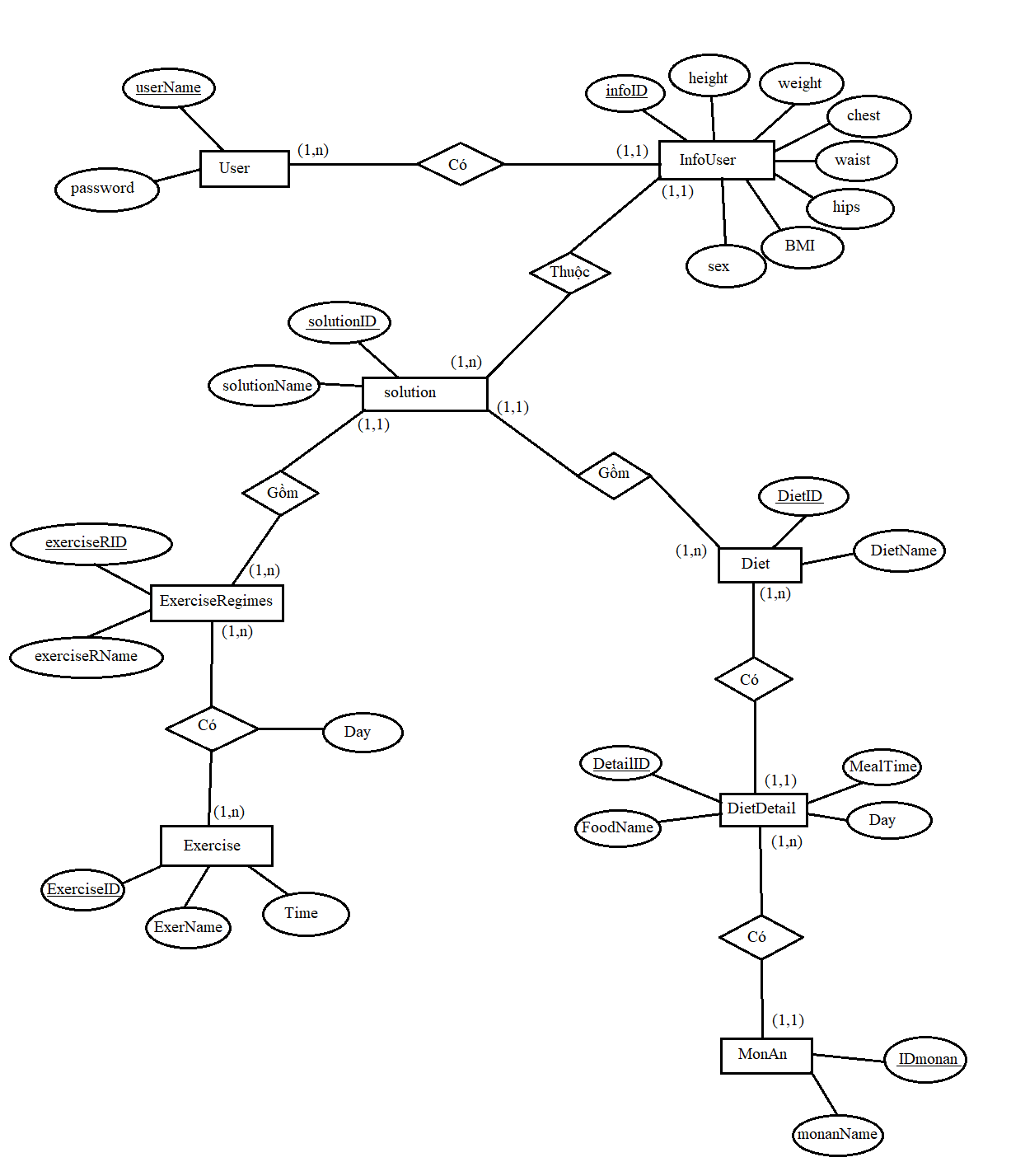
Thông tin chế độ ăn(Dietdetail) bao gồm id của thông tin chế độ ăn (dietdetailID), tên các loại thực phẩm (FoodName),bữa ăn (mealTime),ngày ăn thứ (day). Từ tên các loại thức ăn (FoodName) ta sẽ có những món ăn gợi ý.

Món ăn gới ý(MonAn) bao gồm IdMonAn, tên món ăn (MonAnName).

Thông tin của bài tập theo từng ngày(Exercise) bao gồm tên bài tập (exerciseName), thời gian tập luyên (time), linkYoutube.

## II, Xây dựng cơ sở dữ liệu

### 1, Sơ đồ ERD



# Hình 1: Sơ đồ ERD

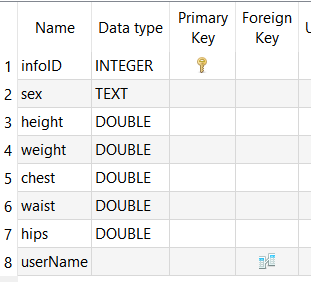
### 2. Mô hình quan hệ

* User (usernames, password)
* InforUser (infoID, height, weight, chest, waist, hips, BMI, sex, username)
* Solutions(solutionID, solutionName, dietID, exerciseRegimeID)
* Diets (dietID, dietName,)
* DietDetails (dietdetailID, FoodName, mealTime, day, dietID)
* MonAn (IdMonAn, MonAnName, dietdetailID)
* ExerciseRegimes (exerciseRegimeID, exerciseRegimeName)
* ExerciseRegimesDetails (ExerciseRegimesDetailID, exerciseRegimeID, exerciseName , day)
* Exercise (exerciseName, time, linkYoutube)

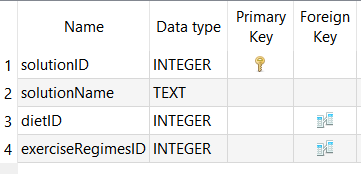
### 3, Bảng dữ liệu

# Hình 2: Bảng user

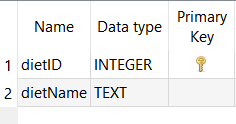
# Hình 3: Bảng InfoUser



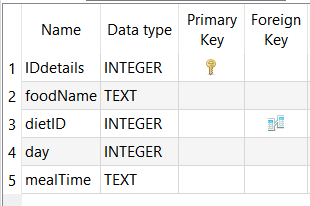
# Hình 4: Bảng Solutions



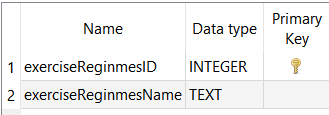
# Hình 5: Bảng Diets



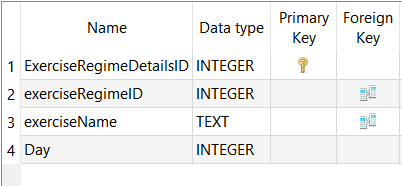
# Hình 6: Bảng DietDetails



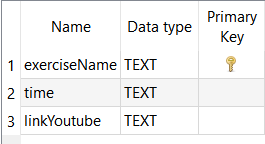
# Hình 7: Bảng ExerciseRegimes



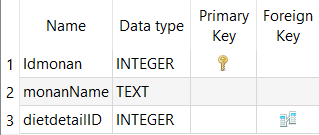
# Hình 8: Bảng ExerciseRegimesDetails



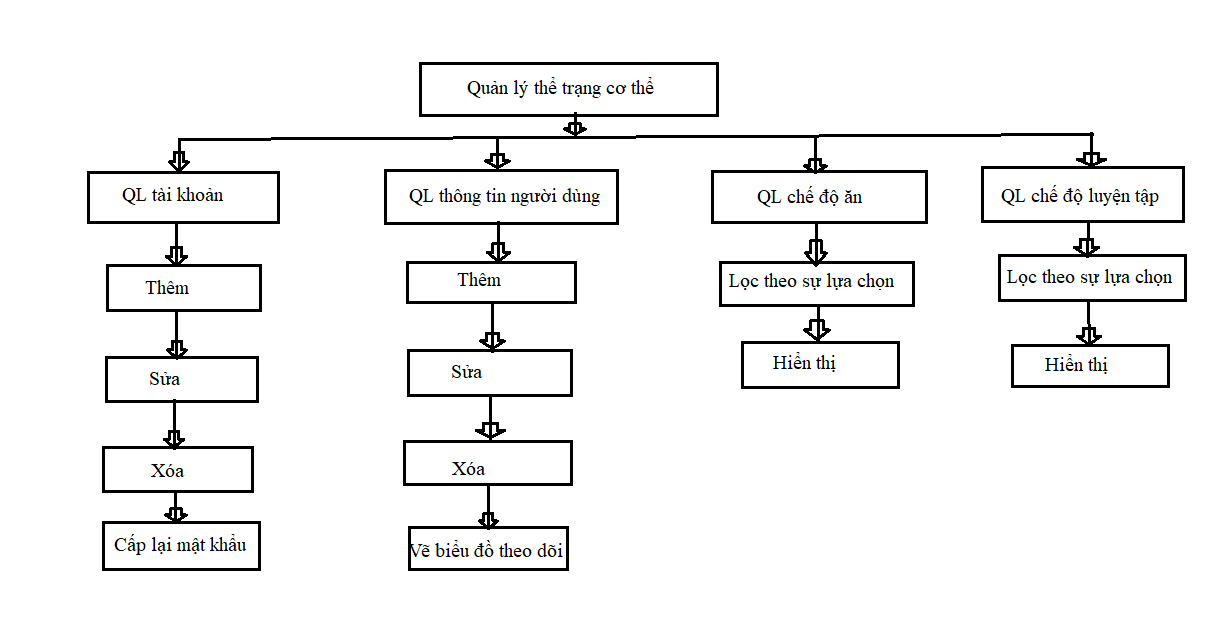
# Hình 9: Bảng Exercise



# Hình 10: Bảng MonAn



### 4, Sơ đồ phân rã chức năng



# Hình 11: Sơ đồ phân rã chức năng

# Phần III: Giao diện và chức năng của hệ thống

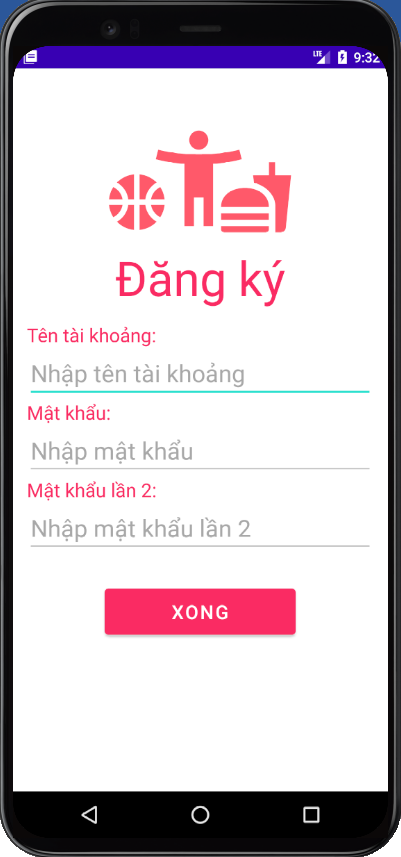
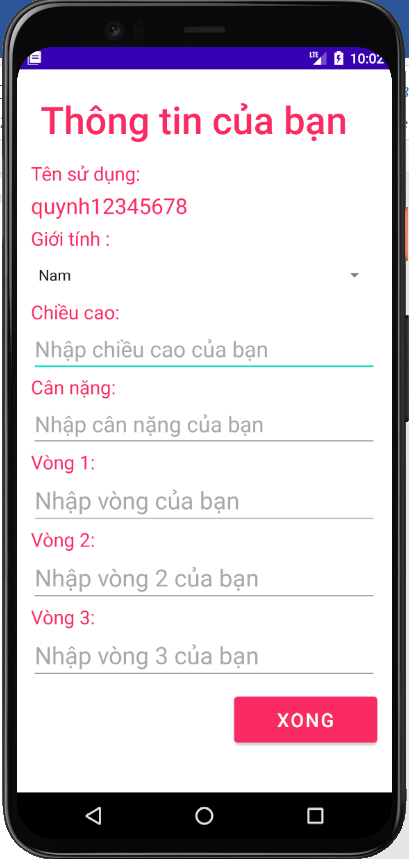
## 1, Các giao diện và ứng dụng từng giao diện.

* Giao diện Welcome : Lời chào hỏi thân thiện và đưa ra cho bạn 2 sự lựa chọn về tài khoảng. Nếu đã có tài khoảng hãy vào và sử dụng ngay, nếu chưa có bạn có thể đăng ký 1 cách nhanh chóng và trải nghiệm app.



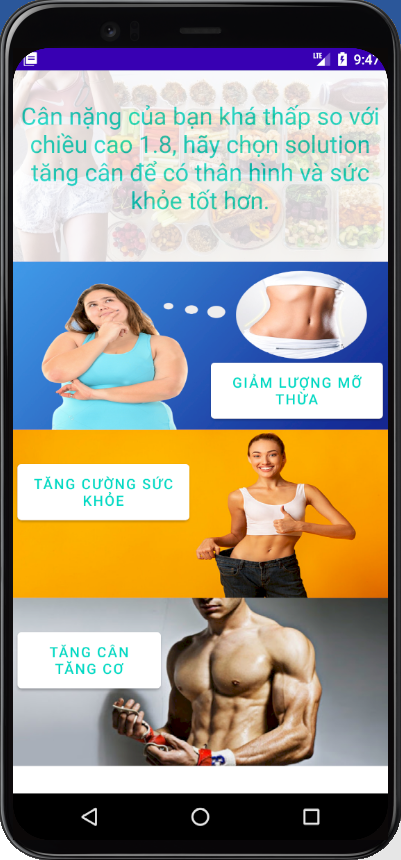
# Hình 11: Giao diện Welcome

* Đăng ký tài khoảng: Sau khi chọn vào nút đăng ký sẽ xuất hiện cho bạn tạo tên tài khoản và mật khẩu theo ý thích nhưng phải hợp với qui định (không trùng với tài khoản đã có, không bỏ trống, mật khẩu lần 2 không trùng khớp với lần 1,..). Sau khi tạo xong tài khoảng bạn sẽ phải tiếp tục nhập các thông số của cơ thể (nhập càng cụ thể và chính xác app sẽ đưa ra chế độ phù hợp nhất với bạn).



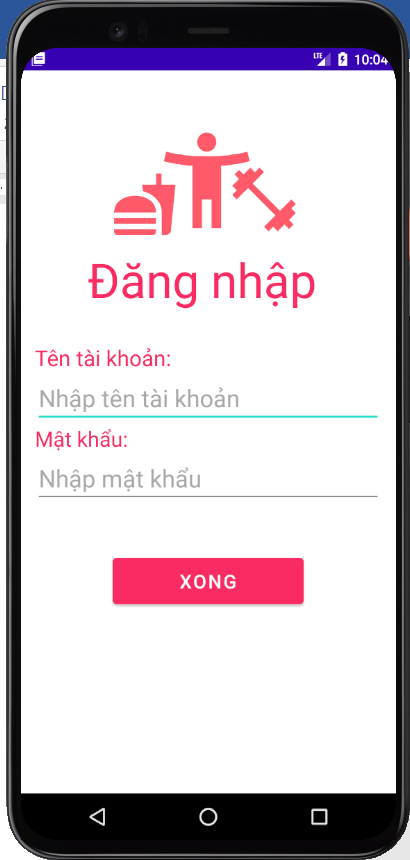
# Hình 12: Giao diện đăng ký tài khoảng

* Giao diện lời khuyên và các sự lựa chọn về các chế độ ăn uống và tập luyện: Sau khi hoàn thành việc tạo tài khoản và điền đầy đủ thông tin bạn sẽ được ứng dụng đưa ra lời khuyên chân thành để bạn có thể chọn chế độ ăn uống và tập luyện một cách thích hợp nhất.



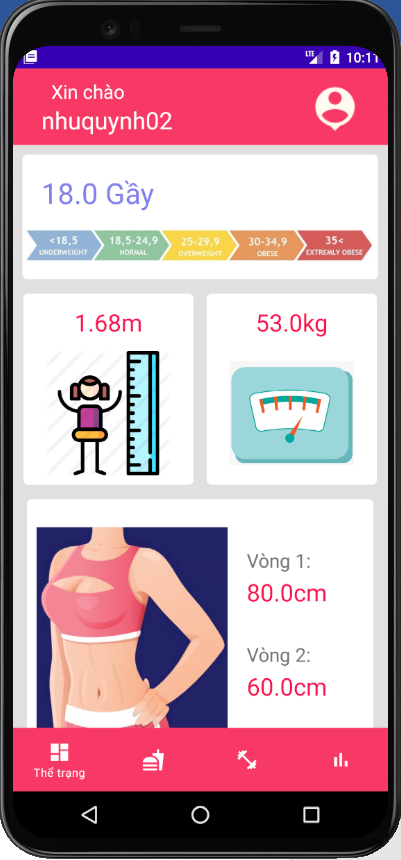
# Hình 13: Giao diện lời khuyên và các sự lựa chọn

* Đăng nhập: Bạn đã có tài khoản, đăng nhập 1 cách nhanh chóng và dễ dàng. Với điều kiện bạn phải nhớ đúng tài khoảng của mình, các lỗi không nhập và sai tài khoảng hoặc mật khẩu sẽ được app thông báo cho bạn.



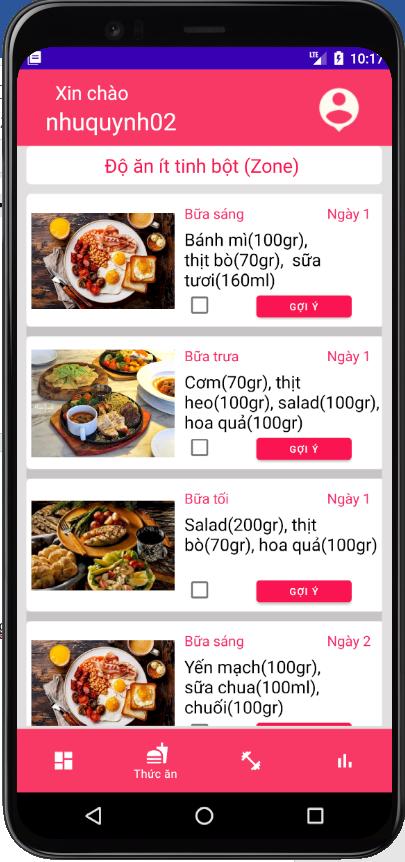
# Hình 14: Giao diện đăng nhập

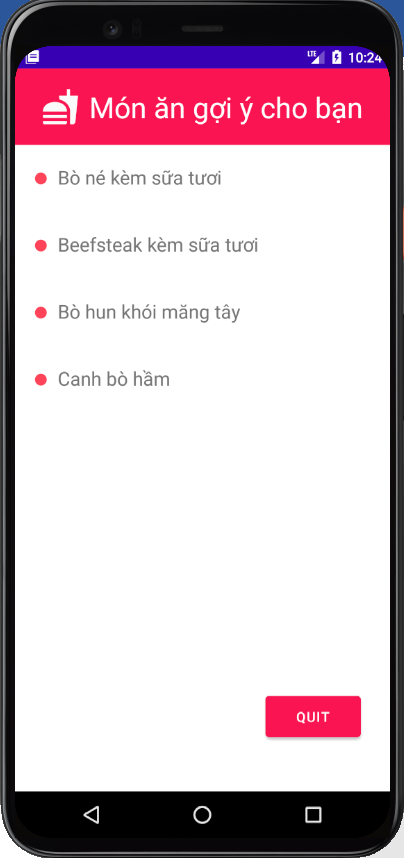
* Hiển thị thông tin người dùng: Sau khi hoàn thành phần đăng nhập hoặc đăng kí ứng dụng sẽ hiện thị thông tin thể trạng gần đây nhất của bạn, bao gồm chỉ số BMI, chiều cao, cân nặng và số đo 3 vòng của bạn.



# Hình 15: Giao diện thông tin người dùng

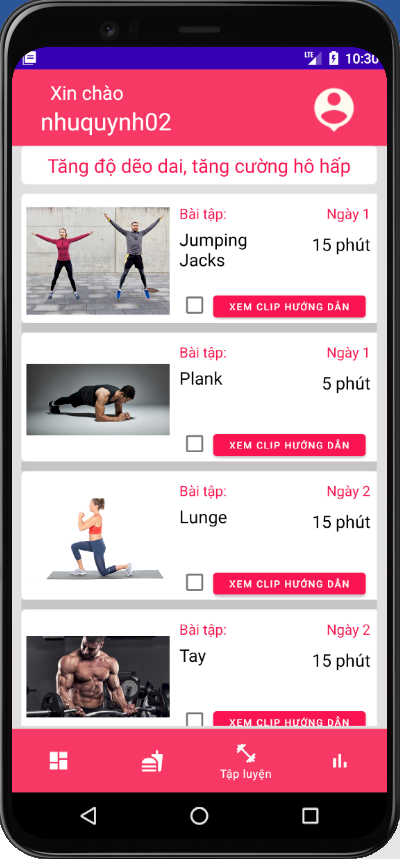
* Hiển thị danh sách bữa ăn theo chế độ ăn của người dùng đã chọn: Trên giao diện này gồm có tên của chế độ ăn, danh sách bữa ăn hằng ngày cho bạn, trong mỗi item bữa ăn sẽ có 1 checkbox để đánh dấu những bữa ăn bạn đã hoàn thành, 1 nút gợi ý khi nhân vào sẽ hiện ra giao diện chứa tên các món ăn được gợi ý.

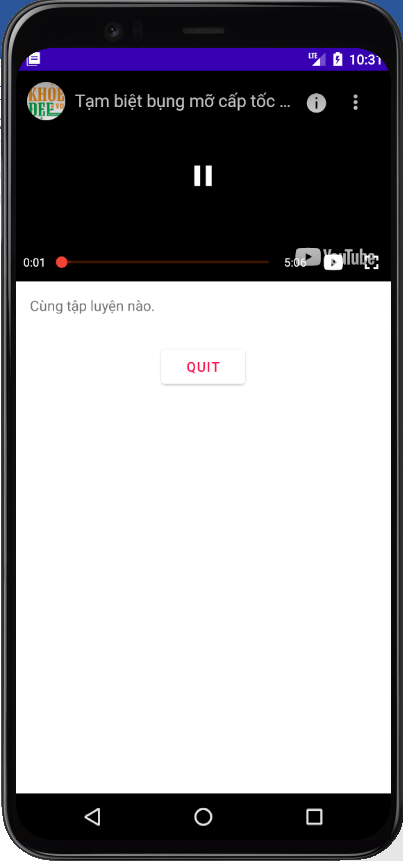




# Hình 16: Giao diện chế độ ăn

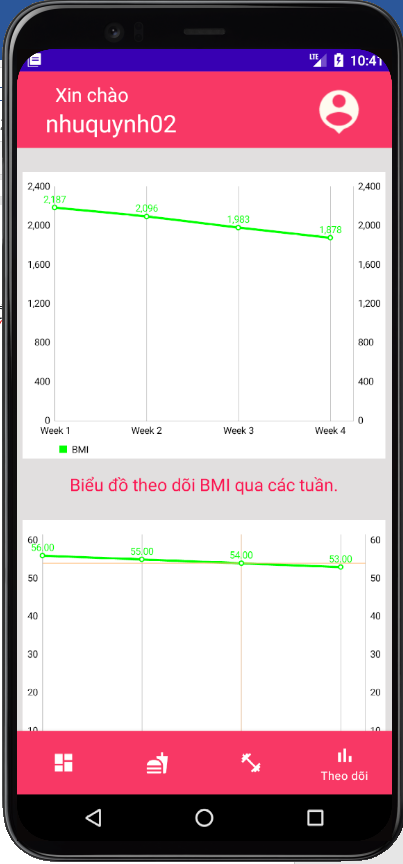
* Hiển thị danh sách các bài tập theo từng ngày của chế độ tập luyện người dùng đã chọn: Trên giao diện này gồm có tên của chế tập luyên, danh sách bài tập hằng ngày cho bạn, trong mỗi item bài tập sẽ có 1 checkbox để đánh dấu những bài tập bạn đã hoàn thành, 1 nút xem video khi nhân vào sẽ hiện ra giao diện chứa video hướng dẫn tập luyện được lấy từ youtube nhờ sử dụng youtubeAPI.





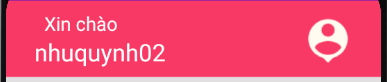
# Hình 17: Giao diện chế độ tập luyện

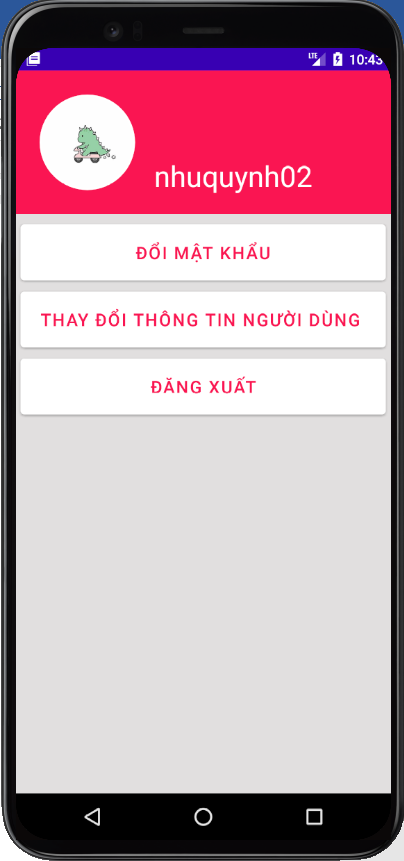
* Hiển thị biểu đồ sự thay đổ chỉ sổ BMI, chiều cao, cân nặng, số đo 3 vòng của người dùng để theo dõi được sự thay đổi của thể trạng người dùng và đưa ra lời khuyên chân thành để người dùng có thể ăn uống cà tập luyện 1 cách tốt hơn.



# Hình 18: Giao diện biểu đồ theo dõi thể trạng

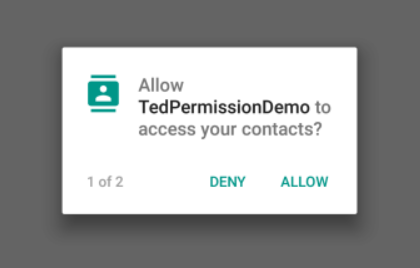
* Thay đổi thông tin người dùng (các chỉ số cơ thể, thông tin tài khoảng, ảnh đại diện, đăng xuất tài khoản)



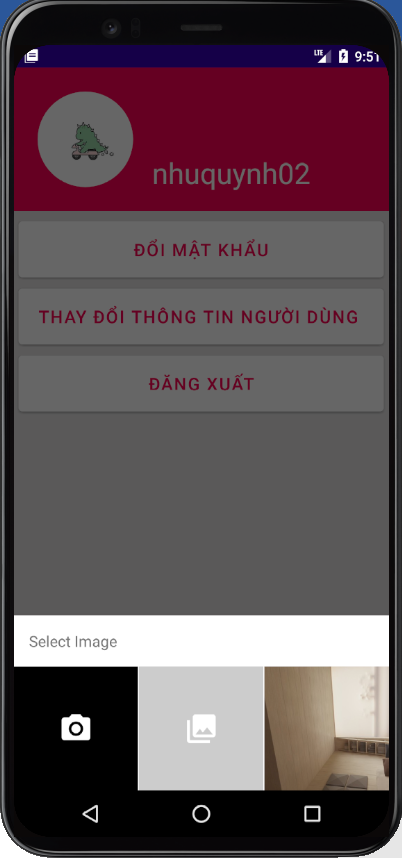


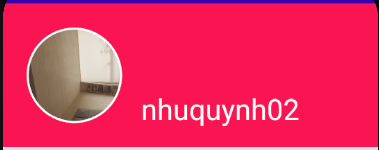
# Hình 19: Giao diện những chức năng nhỏ

* Đổi ảnh đại diện của người dùng: Nhấn chọn vào ảnh đại diện của người dùng sẽ hiện lệ hộp thoại xin quyền truy cập vào camera và album ảnh để có thể lấy ảnh làm ảnh đại diện cho người dùng.

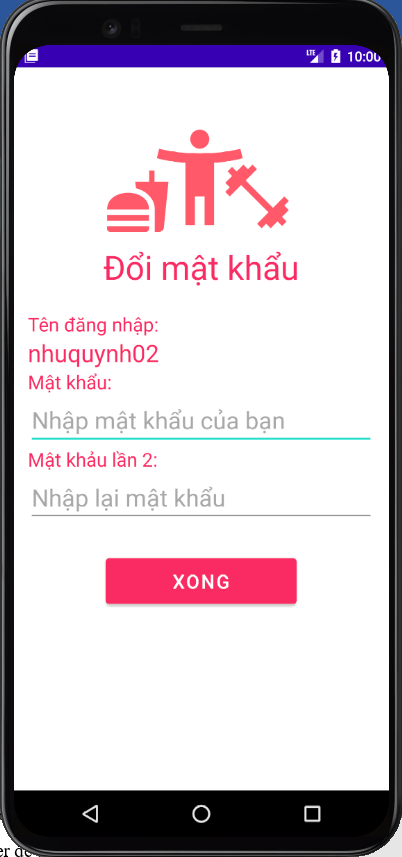








# Hình 20: Giao diện thay đổi ảnh đại diện

* Đổi mật khẩu: Cũng khá giống với giao diện

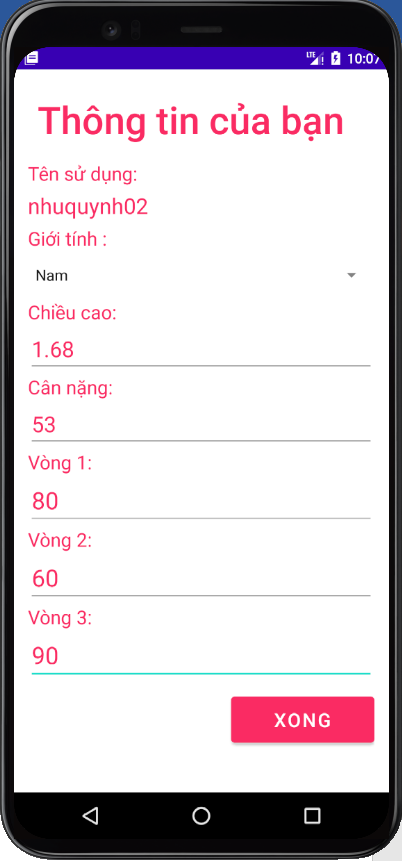
đăng nhập nhưng ở giao diện đổi mật khẩu

tên tài khoản được giữ mặc định là tên tài

khoản người dùng trước đó và tiền hành đổi

lại mật khẩu như người dùng muốn.

# Hình 21: Giao diện đổi mật khẩu



* Chỉnh sửa thông tin thể trạng người dùng:

Cũng giống như giao diện nhập thông tin

nhưng ở giao diện này các thông sẽ được hiển

thị trước để bạn có thể dễ dàng chỉnh sửa hơn.

# Hình 22: Giao diện chỉnh sửa thông tin người dùng

* Đăng xuất: Đăng xuất khỏi tài khoản và trở về

giao diện Welcome

## 2, Kết luận:

Mặt tự tin: Các chế độ ăn và tập luyện trên đã được kiểm chứng độ hiệu quả thông qua chính người xây dựng nên ứng dụng này, ứng dụng chạy khá ổn định, không giật lag.

Mặt hạn chế: Giao diện chưa được thu hút, dữ liệu nên được đẩy lên server để giảm dung lượng của ứng dụng và chỉnh sửa theo từng thời kì.

Link Github: [QuynhUTC2/BaoCaoTienDoThucTapChuyenMon (github.com)](https://github.com/QuynhUTC2/BaoCaoTienDoThucTapChuyenMon)

Tài liệu tham khảo:

[Tai Lieu Lap Trinh Android .zip - Google Drive](https://drive.google.com/file/d/1_Gw2-cX1RsCZgHEEeyogixNOpdjQk2xX/view)

[Tiểu luận môn Lập trình hướng đối tượng: Cơ bản về lập trình ANDROID và ứng dụng - TaiLieu.VN](https://tailieu.vn/doc/tieu-luan-mon-lap-trinh-huong-doi-tuong-co-ban-ve-lap-trinh-android-va-ung-dung-2381874.html)

[android notes for professionals - TaiLieu.VN](https://tailieu.vn/doc/android-notes-for-professionals-2095799.html)

[Hướng dẫn lập trình cơ bản với Android - Phần 2: Android Virtual Device - TaiLieu.VN](https://tailieu.vn/doc/huong-dan-lap-trinh-co-ban-voi-android-phan-2-android-virtual-device-2048346.html)

[Hướng dẫn lập trình cơ bản với Android - Phần 3: Device của Android - TaiLieu.VN](https://tailieu.vn/doc/huong-dan-lap-trinh-co-ban-voi-android-phan-3-device-cua-android-2048347.html)