**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 16 tháng 7 năm 2020***  **Giáo viên hướng dẫn**  **Nguyễn Lê Minh** |

**Lời nói đầu**

Trong cơn sốt dịch COVID-19 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. Và trong thời đại 4.0 hiện nay, đa số ai cũng sở hữu cho mình 1 chiếc điện thoại thông minh.

Vậy nên tôi đã nảy ra 1 ý tưởng là xây dựng nên một App trên điện thoại để giúp mọi người không phải đau đầu logic về nhưng gì phải làm để cũng cố sức khỏe.

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**

**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 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" \o "This (lập trình máy tính) (trang chưa được viết))" 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" \l "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" \l "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)" \o "Java (ngôn ngữ lập trình)), [C++](https://vi.wikipedia.org/wiki/C%2B%2B" \o "C++), [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng" \o "C thăng), [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)" \o "Python (ngôn ngữ lập trình)), [PHP](https://vi.wikipedia.org/wiki/PHP" \o "PHP), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)" \o "Ruby (ngôn ngữ lập trình)), [Perl](https://vi.wikipedia.org/wiki/Perl), [Object Pascal](https://vi.wikipedia.org/wiki/Object_Pascal" \o "Kiểu dữ liệu), [Objective-C](https://vi.wikipedia.org/w/index.php?title=Objective-C&action=edit&redlink=1" \o "Objective-C (trang chưa được viết)), [Dart](https://vi.wikipedia.org/wiki/Dart_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)" \o "Dart (ngôn ngữ lập trình)), [Swift](https://vi.wikipedia.org/wiki/Swift_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)" \o "Swift (ngôn ngữ lập trình)), [Scala](https://vi.wikipedia.org/wiki/Scala" \o "Scala), [Common Lisp](https://vi.wikipedia.org/wiki/Lisp" \o "Lisp), và [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk" \o "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.

* 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. SQLite 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 khoảng(User) gồm usernames, password và có 1 hoặc nhiều InfoUser. Thông tin người dùng (InfoUser) bao gồm infoID, height, weight, chest, waist, hip BMI(được tính từ weight và height).

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, solutionName.

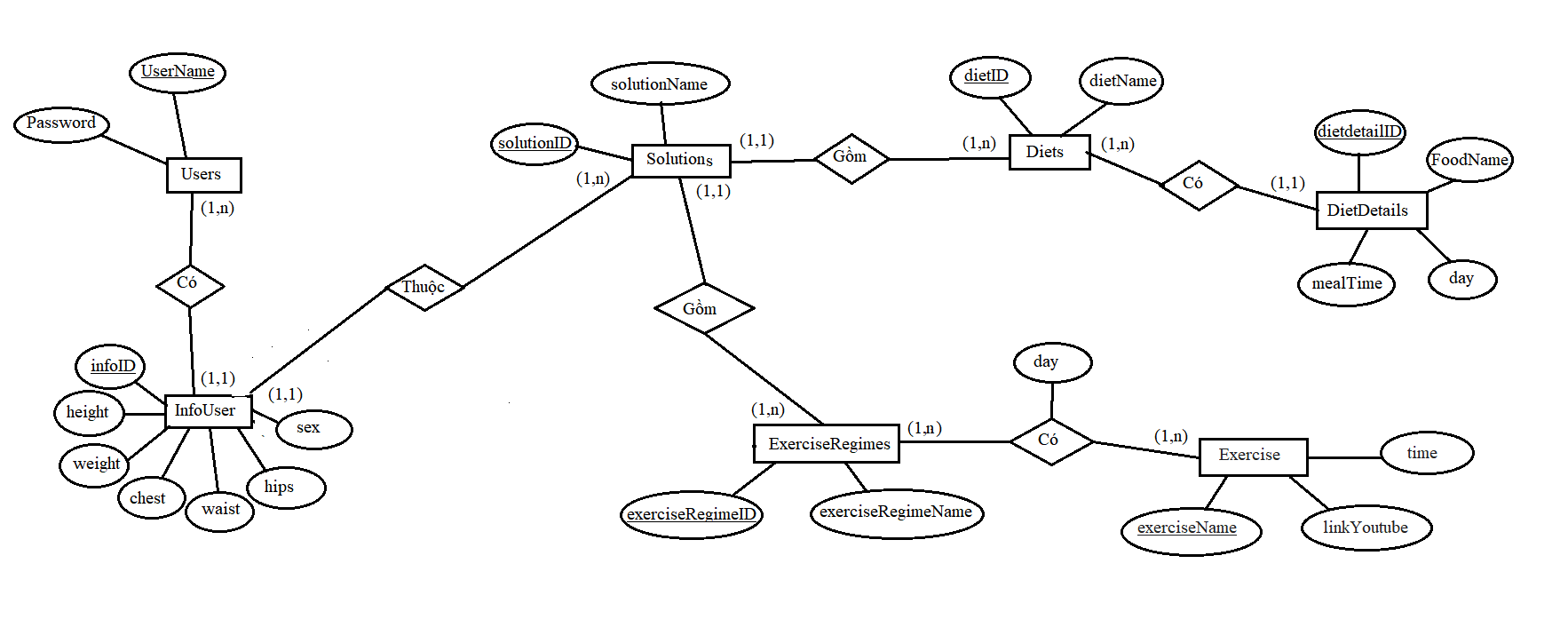
Trong các phương pháp trên mỗi pp bao gồm: một chế độ ăn(diet) sẽ có dietID, dietName bao gồm nhiều thông tin chế độ ăn(Dietdetail) chia theo ngày và một chế độ tập luyên(ExerciseRegimes) bao gồm những thông tin exerciseRegimeID, exerciseRegimeName và nhiều bài tập theo từng ngày(day).

Thông tin chế độ ăn(Dietdetail) bao gồm dietdetailID, FoodName, mealTime, day.

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

**II, Xây dựng cơ sở dữ liệu**

**1, Sơ đồ ERD**

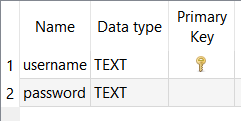


2. Mô hình quan hệ

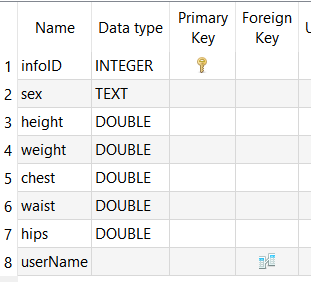
* User (usernames, password)
* InforUser (infoID, height, weight, chest, waist, hip, BMI, age, sex, username)
* Solutions(solutionID, solutionName, dietID, exerciseRegimeID)
* Diets (dietID, dietName,)
* DietDetails (dietdetailID, FoodName, mealTime, day, dietID)
* ExerciseRegimes (exerciseRegimeID,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

3, Bảng dữ liệu

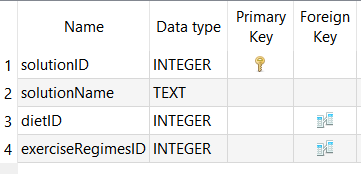
Bảng user:



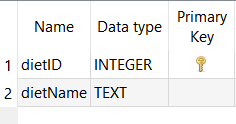
Bảng InfoUser:



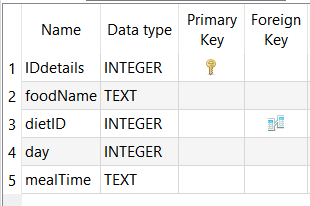
Bảng Solutions:



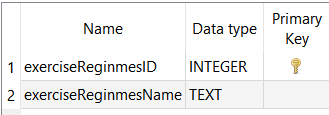
Bảng Diets:



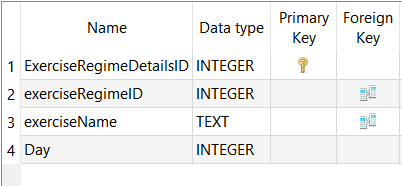
Bảng DietDetails:



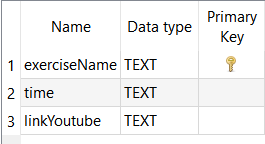
Bảng ExerciseRegimes:



Bảng ExerciseRegimesDetails:



Bảng Exercise:



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

1, Hướng dẫn sử dụng

2, Các layout và ứng dụng từng layout.

* Đăng nhập
* Đăng ký tài khoảng
* Thêm thông tin thể trạng của người dung
* Đưa ra lời khuyên và cho người dùng chọn chế độ ăn và tập luyện
* Hiển thị giao diện thông tin người dung
* Hiển thị danh sách bữa ăn theo chế độ ăn của người dùng đã chọn
* Theo từng bữa ăn sẽ có những món ăn được gợi ý
* Hiển thị danh sách các bài tập theo từng ngày của chế độ tập luyện người dùng đã chọn
* Theo từng bài tập sẽ có những video hướng dẫn cụ thể
* Hiển thị biểu đồ sự thay đổ chỉ sổ BMI của người dùng để theo dỏi được sự thay đổi của thể trạng người dùng
* Thay đổi thông tin người dùng (các chỉ số cơ thể, thông tin tài khoảng, ảnh đại diện)

Kết luận: