# BỘ THÔNG TIN VÀ TRUYỀN THÔNG

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------



**BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*:**

**THIẾT KẾ WEBSITE QUẢN LÝ VĂN THƯ**

### Giảng viên hƣớng dẫn : ThS.LƯU NGUYỄN KỲ THƯ Sinh viên thực hiện : NGUYỄN TRUNG ĐOÀN

### Mã số sinh viên: N14DCCN078

**Lớp : D14CQIS01**

**Khóa : 2014 - 2019**

**Hệ : ĐẠI HỌC CHÍNH QUI**

**TP.HCM, tháng 8 /2018**

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý thầy cô ở khoa Công Nghệ Thông Tin 2 – Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã tận tâm chỉ bảo, truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt trong học kỳ này, nếu không có những lời hướng dẫn, dạy bảo của các thầy cô thì em nghĩ bài báo cáo này của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy cô. Bài báo cáo thực tập thực hiện trong khoảng thời gian 10 tuần. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều bỡ ngỡ. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Em xin gởi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô ở khoa Công Nghệ Thông Tin 2 – Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã tạo điều kiện cho em để em có thể hoàn thành tốt bài báo cáo thực tập này. Đặc biệt em xin gửi lời cảm ơn đến thầy ThS.Huỳnh Trung Trụ với những buổi vừa học kết hợp với thực hành trên lớp cùng những buổi nói chuyện rất hữu ích đã giúp em định hướng và hoàn thành tốt đề tài này.

Trong quá trình làm bài báo cáo, cũng như là trong quá trình làm đề tài khó tránh khỏi 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 thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp Thầy, Cô để em học thêm được nhiều kinh nghiệm làm hành trang vững chắc để em tự tin theo đuổi sự nghiệp của mình.

Em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, tháng 8 năm 2017

Nguyễn Thị Hồng Nhung

# MỤC LỤC

[LỜI MỞ ĐẦU 1](#_bookmark0)

[CHƢƠNG 1. GIỚI THIỆU 2](#_bookmark1)

* 1. [Mục tiêu 2](#_bookmark2)
  2. [Lĩnh vực 2](#_bookmark3)
  3. [Tính năng 2](#_bookmark4)
  4. [Chức năng chính 2](#_bookmark5)
  5. [Yếu tố công nghệ 3](#_bookmark6)

[CHƢƠNG 2. CƠ SỞ LÝ THUYẾT 4](#_bookmark7)

* 1. [PHẦN MỀM HỖ TRỢ 4](#_bookmark8)
     1. [Cơ bản về Xcode 4](#_bookmark9)
        1. [Khái niệm Xcode 4](#_bookmark10)
        2. [Cơ bản về Xcode 4](#_bookmark11)
        3. [Cách sử dụng Xcode 7](#_bookmark15)
  2. [Cơ bản về Swift 8](#_bookmark16)
     1. [Khái niệm Swift 8](#_bookmark17)
     2. [Các đặc điểm cơ bản trong Swift 8](#_bookmark18)
  3. [Cơ bản về Realm và RealmSwift 14](#_bookmark30)
     1. [Khái niệm Realm và RealmSwift 14](#_bookmark31)
     2. [Một số công cụ hỗ trợ 14](#_bookmark32)
     3. [Cách sử dụng Realm và RealmSwift 16](#_bookmark36)

[CHƢƠNG 3. PHÂN TÍCH THIẾT KẾ XÂY DỰNG ỨNG DỤNG](#_bookmark41) [HEALTH](#_bookmark42) OF [CHILD 17](#_bookmark42)

* 1. [PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU 17](#_bookmark43)
     1. [Mô hình ngữ cảnh 17](#_bookmark44)
     2. [Mô hình DFD các mức 17](#_bookmark46)
        1. [Mô hình DFD mức 0 17](#_bookmark47)
        2. [Mô hình DFD mức 1: 18](#_bookmark49)
        3. [Mô hình DFD mức 2 19](#_bookmark51)
        4. [Mô hình DFD mức 3 19](#_bookmark53)
        5. [Mô hình DFD mức 4 20](#_bookmark55)
     3. [Lược đồ Diagram 20](#_bookmark57)
     4. [Lược đồ CSDL 21](#_bookmark59)
  2. [XÂY DỰNG ỨNG DỤNG HEALTH OF CHILD 23](#_bookmark60)
     1. [Màn hình Home 23](#_bookmark61)
     2. [Màn hình Menu 25](#_bookmark64)
     3. [Màn hình Chuẩn đoán 25](#_bookmark66)
     4. [Màn hình Read 27](#_bookmark71)
     5. [Màn hình Cause 28](#_bookmark73)

[3.3 Thuật toán chuẩn đoán 28](#_bookmark74)

[CHƢƠNG 4. GIỚI THIỆU ỨNG DỤNG HEALTH OF CHILD 29](#_bookmark76)

* 1. [Giới thiệu ứng dụng Health Of Child 29](#_bookmark77)
  2. [Các chức năng của ứng dụng 29](#_bookmark78)

[CHƢƠNG 5. KẾT LUẬN VÀ HƢỚNG PHÁT TRIỂN 35](#_bookmark86)

* 1. [Kết luận 35](#_bookmark87)
  2. [Hướng phát triển 35](#_bookmark88)

[5.2.1 Hướng khắc phục 35](#_bookmark89)

[5.2.2. Hướng mở rộng 35](#_bookmark90)

[DANH MỤC TÀI LIỆU THAM KHẢO 36](#_TOC_250000)

**DANH MỤC HÌNH**

[Hình 1: Màn hình 1 project mới được khởi tạo 5](#_bookmark12)

[Hình 2: Màn hình Simulator khi khởi động lên 6](#_bookmark13)

[Hình 3: Name Project trên màn hình Simulator 7](#_bookmark14)

[Hình 4: Mô hình Top-Level 8](#_bookmark19)

[Hình 5: Ví dụ về biến và khai báo biến trong Swift 9](#_bookmark20)

[Hình 6: Bảng toán tử trong Swift 10](#_bookmark21)

[Hình 7: Ví dụ If-Else trong Swift 10](#_bookmark22)

[Hình 8: Ví dụ Switch – Case trong swift 11](#_bookmark23)

[Hình 9: Cú pháp for trong Swift 11](#_bookmark24)

[Hình 10: Ví dụ vòng lặp for trong Swift 12](#_bookmark25)

[Hình 11: Cú pháp While trong Swift 12](#_bookmark26)

[Hình 12: Cú pháp repeat-while trong Swift 12](#_bookmark27)

[Hình 13: Mô hình Break trong vòng lặp 13](#_bookmark28)

[Hình 14: Mô hình Continue trong vòng lặp 13](#_bookmark29)

[Hình 15: Realm Plugin trong Xcode 14](#_bookmark33)

[Hình 16: Realm File trong Xcode 15](#_bookmark34)

[Hình 17: Realm Browser trong Xcode 15](#_bookmark35)

[Hình 18: Ghi dữ liệu vào Realm 16](#_bookmark37)

[Hình 19: Ghi dữ liệu vào Realm từ 1 file Realm 16](#_bookmark38)

[Hình 20: Đọc dữ liệu vào Realm 16](#_bookmark39)

[Hình 21: Thêm sửa xóa dữ liệu vào Realm 16](#_bookmark40)

[Hình 22: Mô hình ngữ cảnh 17](#_bookmark45)

[Hình 23: Mô hình DFD mức 0 17](#_bookmark48)

[Hình 24: Mô hình DFD mức 1 18](#_bookmark50)

[Hình 25: Mô hình DFD mức 2 19](#_bookmark52)

[Hình 26: Mô hình DFD mức 3 19](#_bookmark54)

[Hình 27: Mô hình DFD mức 4 20](#_bookmark56)

[Hình 28: Lược đồ Diagram 20](#_bookmark58)

[Hình 29: Source Code Giao Diện Màn hình Home 24](#_bookmark62)

[Hình 30: Source Code Danh mục Màn hình Home 24](#_bookmark63)

[Hình 31: Source Code Danh Dách nội dung triệu chứng màn hình Menu 25](#_bookmark65)

[Hình 32: Mô hình Delegate 26](#_bookmark67)

[Hình 33: Gọi và thực thi Delegate 26](#_bookmark68)

[Hình 34: Delegate Câu hỏi và câu trả lời 26](#_bookmark69)

[Hình 35: Xử lý câu hỏi và câu trả lời 27](#_bookmark70)

[Hình 36: Cách đọc file PDF 27](#_bookmark72)

[Hình 37: Thuật toán chuẩn đoán 28](#_bookmark75)

[Hình 38: Màn hình Home 29](#_bookmark79)

[Hình 39: Màn hình Menu 30](#_bookmark80)

[Hình 40: Màn hình đọc nội dung 31](#_bookmark81)

[Hình 41: Màn hình mô tả triệu chứng 32](#_bookmark82)

[Hình 42: Màn hình chuẩn đoán 32](#_bookmark83)

[Hình 43: Màn hình kết quả 33](#_bookmark84)

[Hình 44: Màn hình nguyên nhân và tự giải quyết 34](#_bookmark85)

# DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Ký hiệu | Giải thích |
| DB | Data Base |
| MVC | Mô hình Model View Controller |
| TKXD | Thiết kế xây dựng |
| CSDL | Cơ sở dữ liệu |

## LỜI MỞ ĐẦU

### Lý do chọn đề tài

Ngày nay với tốc độ phát triển của khoa học kỹ thuật trên thế giới ngày càng mạnh mẽ. Cuộc cách mạng công nghệ thông tin đã và đang diễn ra trên hầu hết các nước tiên tiến trên thế giới. Cùng với sự phát triển không ngừng của công nghệ thông tin nhu cầu về ứng dụng phần mềm càng được tăng cao. Có thể nói phần mềm ngày nay đóng vai trò hết sức quan trọng trong cuộc sống hàng ngày của mỗi con người từ việc ăn gì, xem gì ở đâu cho đến vấn đề cổ phiếu tăng giá hay giảm, những vấn đề quan trọng của thế giới hay phần mềm giúp con người làm việc nhanh hơn, thậm chí thay con người làm việc.

Nhìn nhận vấn đề đầu tư và phát triển phần mềm ở thế giới nói chung và nước ta nói riêng đang rất phát triển. Đặc biệt là những phần mềm có thể đáp ứng được lượng thông tin cần thiết và được tích hợp trong những chiếc điện thoại thông minh nhỏ gọn có thể dùng bất cứ khi nào. Trong thời đại bận rộn hiện nay càng khó có thể có thời gian đi chọn những quyển báo hay cuốn sách hay để đọc.

Chính vì các lý do trên, em đã lựa chọn đề tài Health of Child là phần mềm cung cấp kiến thức các bệnh thông thường và các rối loạn của trẻ em được mô tả cùng với các nguyên nhân gây bệnh, cách điều trị, các cách tự giúp và tiên lượng bệnh cho tất cả mọi người nhất là những bà mẹ ông bố đang có con nhỏ có thể đọc và tìm hiểu về một vài triệu chứng bệnh thường gặp ở trẻ. Ngoài ra, ứng dụng có các bộ câu hỏi và câu trả lời của những triệu chứng bệnh của em bé để chẩn đoán giúp các bậc cha mẹ và những người nuôi trẻ xác định xem với một triệu chứng hay một nhóm triệu chứng nào thì cần hỏi ý kiến bác sĩ và khi cần tìm ra bệnh càng sớm càng tốt.

## CHƢƠNG 1. GIỚI THIỆU

## Mục tiêu

Dùng Swift trên Xcode xây dựng ứng dụng Health of Child trên IOS giúp người dùng có thể đọc và tìm hiểu về kiến thức của trẻ bao gồm: cơ thể của trẻ, sự phát triển của trẻ, các triệu chứng, nguyên nhân, một vài bệnh thường gặp ở trẻ. Ngoài ra, ứng dụng còn cung cấp bộ câu hỏi, câu trả lời của từng triệu chứng hỗ trợ chẩn đoán bệnh, cách giải quyết bệnh tại nhà và một số phương pháp sơ cứu, chăm sóc cho trẻ.

## Lĩnh vực

Chuyên ngành: công nghệ phần mềm

Chuyên môn: lập trình ứng dụng di động. Sử dụng ngôn ngữ Swift xây dựng ứng dụng Health of Child bằng công cụ Xcode trên nền tảng IOS.

Lĩnh vực liên quan: y tế

## Tính năng

Health of Child là ứng dụng dành cho tất cả mọi người, đặc biệt là các bà mẹ ông bố đang có trẻ nhỏ có thể sử dụng mọi lúc mọi nơi với chiếc điện thoại của mình. Ứng dụng cung cấp thông tin chi tiết về một chứng bệnh đặc biệt bao gồm các triệu chứng, nguyên nhân, khi nào cần đi khám bệnh, cách điều trị …

Trẻ em thường hay bị bệnh hơn người lớn, các bệnh và chấn thương dễ xảy ra nhưng cũng mau khỏi. Trong trường hợp như thế, các bậc cha mẹ luôn muốn biết khi nào thì cần phải đưa bé đi khám bệnh. Ứng dụng này sẽ giúp các bạn phân biệt đước các bệnh lặt vặt với các vấn đề trầm trọng cần phải được sự quan tâm của y tế ngay lập tức. Ứng dụng này không giúp bạn chẩn đoán thay cho bác sĩ, nhưng có thể giúp bạn quyết định cứ để triệu chứng diễn tiến tự nhiên rồi sẽ tự khỏi hoặc tự cho trẻ uống thuốc ở nhà và ngược lại – giúp bạn nhận ra được những dấu hiệu báo động cần phải điều trị gấp. Ngoài ra, bạn có thể tìm lại những thông tin mà bác sĩ đã phổ biến cho bạn nhưng có thể bạn đã quên hoặc chưa nắm rõ lúc bạn đi khám bệnh

## Chức năng chính

* + - Cơ sở dữ liệu: Dữ liệu được lưu trên device bằng công cụ Realm
    - Trang Home: Chứa những đề mục chính: tìm hiểu cơ thể của trẻ, biểu đồ triệu chứng, bệnh và các rối loạn, sơ cứu và chăm sóc trẻ bệnh, lịch sử, hướng dẫn sử dụng
    - Trang Menu:

+ Search Bar: tìm kiếm nhanh các triệu chứng bệnh và các mục khác.

+ Table View: hiển thị các mục của những tiêu đề ở home.

* + - Trang Read: Hiển thị nội dung, thông tin của bệnh, tìm hiểu cơ thể của trẻ, cách sơ cứu và chăm sóc

* + - Trang Chuẩn đoán : cho người dùng chẩn đoán bệnh theo theo bộ câu hỏi và câu trả lời liên tiếp nhau, đưa ra kết luận nguyên nhân, hành động cần làm và tự giải quyết tại nhà.

## Yếu tố công nghệ

* + - Hệ điều hành Mac OS.
    - Phần mềm Xcode.
    - Các máy điện thoại, Ipad có Hệ điều hành IOS.
    - Máy giả lập Iphone , Ipad.
    - Simulator giả lập của Iphone, Ipad của Xcode.

## CHƢƠNG 2. CƠ SỞ LÝ THUYẾT

## PHẦN MỀM HỖ TRỢ

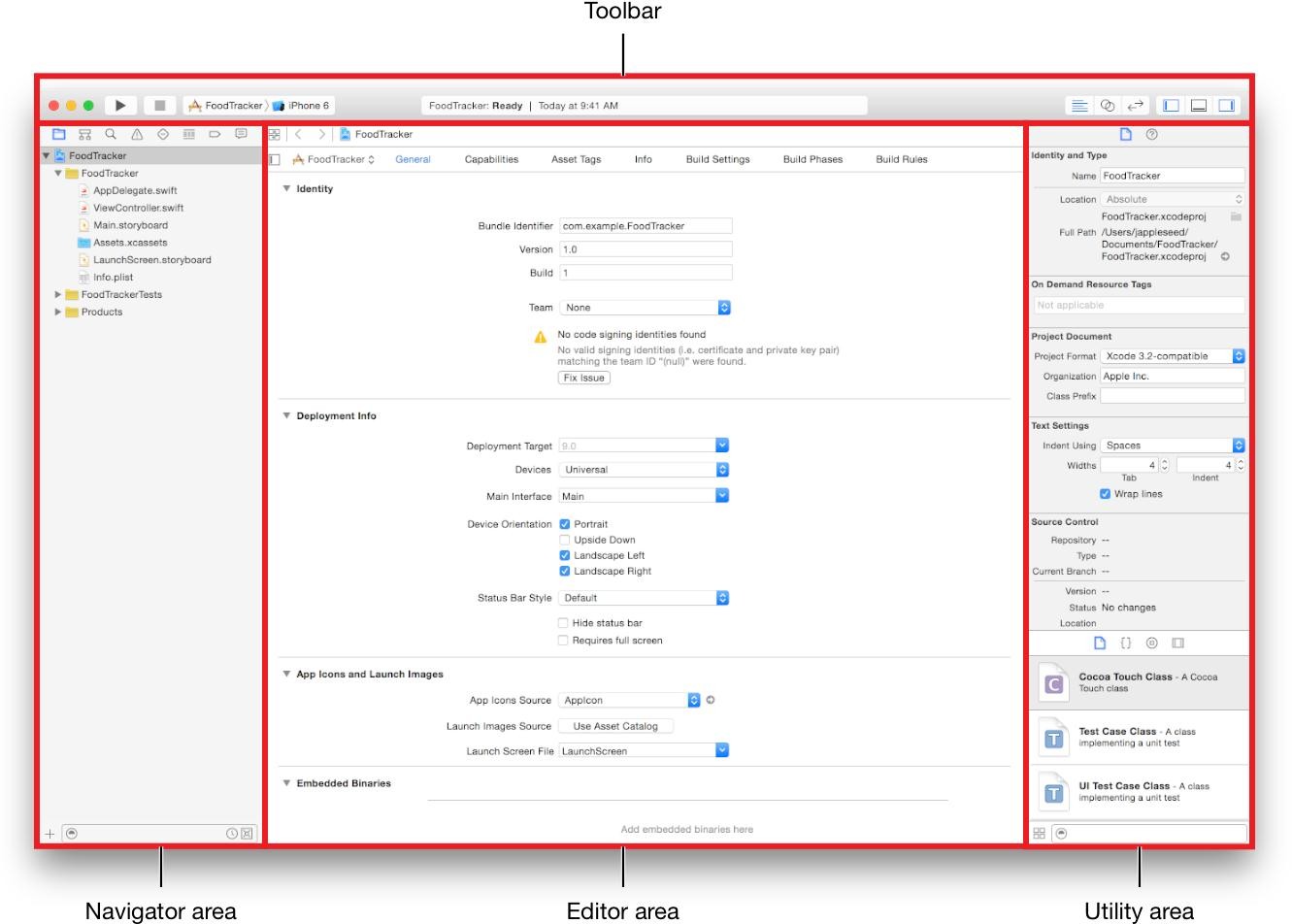
## Cơ bản về Xcode

## Khái niệm Xcode

* + - * + Xcode là công cụ chính để lập trình Iphone/Ipad. Tất cả những gì mới nhất của Apple đều được tích hợp trong Xcode, IOS 6 hay nhưng gì mới nhất của hệ điều hành. Apple rất ưu ái cho các lập trình viên những người đã làm cho hệ sinh thái của Apple trở nên đa dạng.
        + Xcode còn tích hợp cả máy ảo (Simulation) để cho các lập trình viên thử nghiệm các sản phẩm không cần thiết bị thật và môi trường SandBox để kiểm tra các mua bán (in game purchase).
        + Xcode là công cụ dễ dàng và đơn giản nhất để phát triển một phần mềm Iphone hay Ipad.
        + IOS là hệ điều hành chạy trên tất cả các thiết bị iPhonevà iPad. Nó chịu trách nhiệm cho tất cả các hoạt động cơ bản của điện thoại, chẳng hạn như thực hiện cuộc gọi điện thoại, các thao tác vẽ trên màn hình, và chạy các ứng dụng.

## Cơ bản về Xcode

Xcode bao gồm mọi thứ cần để tạo ra một ứng dụng. Nó không chỉ tổ chức các tập tin để tạo ra một ứng dụng mà còn cung cấp một trình biên dịch compiler, một trình gỡ lỗi debugger mạnh mẽ và các thành phần (component) khác cho phép xây dựng và chạy ứng dụng.



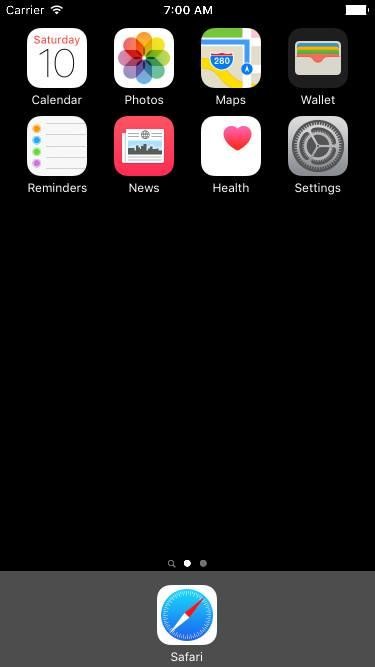
### Hình 1: Màn hình 1 project mới đƣợc khởi tạo

Chạy Simulator

Bởi vì chúng ta sẽ phát triển ứng dụng dự án dựa trên một mẫu template có sẵn nên Xcode sẽ giúp chúng ta cấu hình môi trường cơ bản cho ứng dụng. Mặc dù không cần phải viết bất kỳ đoạn mã nào vẫn có thể tạo và chạy 1 ứng dụng mẫu Single View Application mà không cần phải cấu hình thêm nữa.

Để xây dựng và chạy ứng dụng chúng ta sẽ sử dụng chương trình mô phỏng simulator có trong Xcode. Simulator cung cấp một cái nhìn trực quan về các ứng dụng sẽ hoạt động như thế nào nếu nó được chạy trên một thiết bị thực.

Simulator có thể mô hình hóa nhiều loại thiết bị (device) khác nhau như iPad, iPhone và với kích thước màn hình khác nhau của các device này. Do đó nó có thể mô phỏng các ứng dụng chạy như thế nào trên tất cả các thiết bị khác nhau.

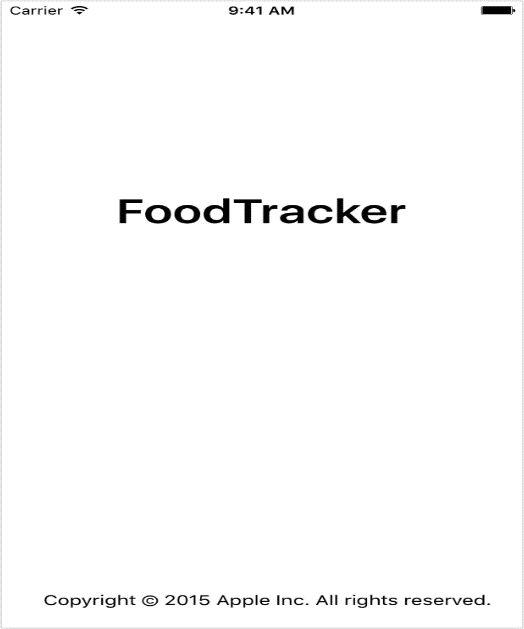


### Hình 2: Màn hình Simulator khi khởi động lên

Xcode hiển thị các thông báo về quá trình build dự án trong khung activity viewer nằm ở giữa thanh công cụ.

Sau khi kết thúc việc build dự án, trình mô phỏng simulator sẽ được tự động khởi động, quá trình này có thể mất một vài phút.

Simulator sẽ mở ứng dụng trong chế độ của iPhone, điều này được chúng ta cấu hình ở các bước phía trên. Trên màn hình mô phỏng thiết bị iPhone, trình mô phỏng simulator sẽ khởi chạy ứng dụng. Trước khi ứng dụng kết thúc việc khởi động, màn hình hiển thị tên ứng dụng được đặt trước trong một thời gian ngắn.

### Hình 3: Name Project trên màn hình Simulator

Kết thúc bước trên đã hoàn tất việc tạo một ứng dụng theo kiểu Single View Application sử dụng Xcode. Sơ qua cơ bản về XCODE sẽ hiểu được các thành phần khác nhau của Xcode và có thể sẵn sàng tiếp cận những kiến thức phức tạp hơn sau này.

Cuối cùng để thoát Simulator chọn Simulator > Quit Simulator (hoặc nhấn Command + Q) để tắt ứng dụng.

## Cách sử dụng Xcode

Mở Xcode lên -> New Project -> Chọn Single View Application -> Trong cửa sổ mới hiện ra, bạn để ý vào những thông tin sau:

* + - * + Product Name (Tên ứng dụng, có thể viết chữ hoa, chữ thường, chữ Trung Quốc, Nhật Bản, ... thoải mái)
        + Organization Name (Tên công ty, nếu bạn là cá nhân, có thể để tên bạn)
        + Company Identifier (Mã chứng thực công ty, hoặc có thể là cá nhân. Vì đây là bài Test đầu tiên, cho nên chỗ này, bạn có thể điền là "com.education")
        + Class Prefix (Tiền tố đứng trước tên của Class, bạn có thể điền hoặc không, nếu tò mò, bạn hãy thử và xem có gì thay đổi hay không sau khi đọc và thực hành xong bài này)
        + Devices (Thiết bị mà bạn lập trình ứng dụng, Universal nghĩa là cả 2 thiết bị iPhone và iPad, ở đây bạn chọn iPhone nhé, vì sao, để bạn build cho dễ dàng, và gọn màn hình, không cần cuộn chuột để xem nội dung, và bài viết này yêu cầu cũng rất đơn giản)

Cuối cùng ở 3 dấu tích, bạn chọn Use Storyboards và Use Automatic Reference Counting nhé.

Tiếp theo chọn Next, chọn nơi lưu Project này.

## Cơ bản về Swift

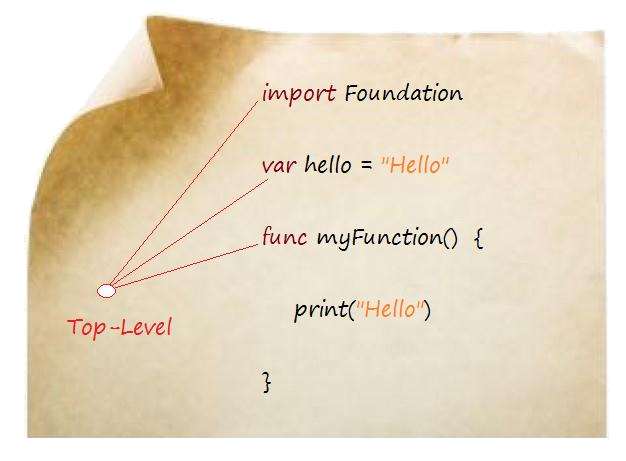
## Khái niệm Swift

Swift là ngôn ngữ mới nhất hiện tại của Apple bao gồm Swift 2 và Swift 3.

Đượcphát triển lên từ Objective C và có tính hướng đối tượng.

Để bắt đầu, Chúng ta cần phải sử dụng hệ điều hành Mac OS (trên máy Mac hoặc Hackintosh) và dĩ nhiên là phải cài ứng dụng Xcode.

## Các đặc điểm cơ bản trong Swift

Một dòng lệnh hoặc biểu thức không nằm trong một hàm, khối lệnh hoặc một class có nghĩa là nó nằm ở *Top-Level*

### Hình 4: Mô hình Top-Level

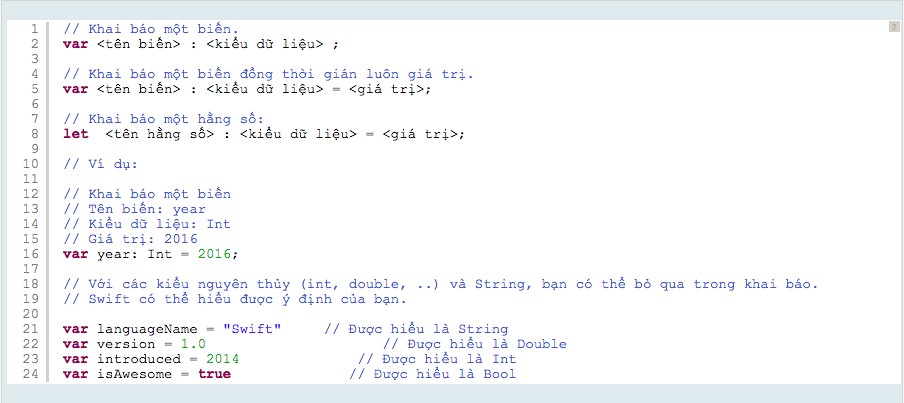
Top- Level là nơi khai báo sử dụng các thư viện, khai báo biến, hằng số, hàm, lớp. Theo mặc định các biến và hằng số được khai báo ở mức Top-Level của file nguồn này có thể được sử dụng trong các file nguồn khác trong cùng một module (project).

Bạn không thể viết một biểu thức ở Top-Level, chẳng hạn một biểu thức để gọi hàm. Vì vậy gọi hàm print("Hello") để in ra dòng chữ "Hello" ở Top-Level là không hợp lệ, trừ khi bạn làm việc đó ở file nguồn có tên main.swift (Đây là trường hợp ngoại lệ).

* + - * Các kiểu dữ liệu trong Swift

|  |  |  |
| --- | --- | --- |
| **Kiểu dữ liệu** | **Độ rộng** |  |
| Int8 | 1 byte | -127 tới 127 |
| UInt8 | 1 byte | 0 tới 255 |
| Int32 | 4 bytes | -2147483648 tới 2147483647 |
| UInt32 | 4 bytes | 0 tới 4294967295 |
| Int64 | 8 bytes | -9223372036854775808 tới -  9223372036854775807 |
| UInt64 | 8 bytes | 0 tới 18446744073709551615 |
| Float | 4 bytes | 1.2E-38 tới 3.4E+38 (~6 digits) |
| Double | 8 bytes | 2.3E-38 tới 1.7E+308 (~16 digits) |

* + - * Biến và khai báo: Biến trong swift được khai báo như sau



### Hình 5: Ví dụ về biến và khai báo biến trong Swift

* + - * Các câu lệnh rẽ nhánh trong Swift

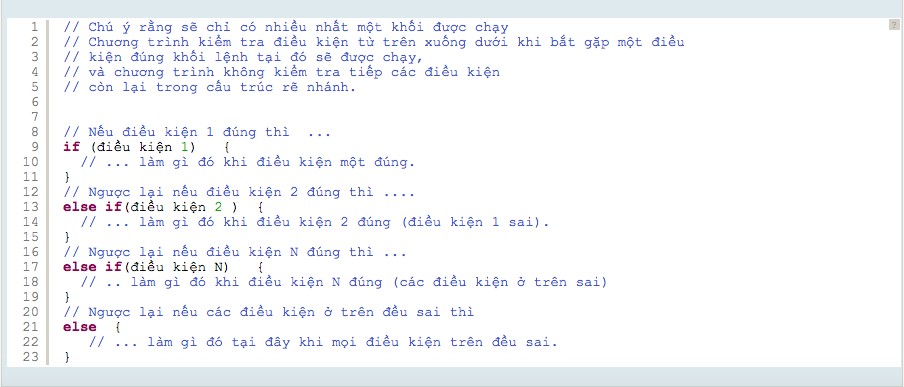
**-** Câu lệnh If-else: **if** là một câu lệnh kiểm tra một điều kiện gì đó trong **Swift**.

Chẳng hạn: Nếu **a > b** thì làm gì đó .... Các toán tử so sánh thông dụng:



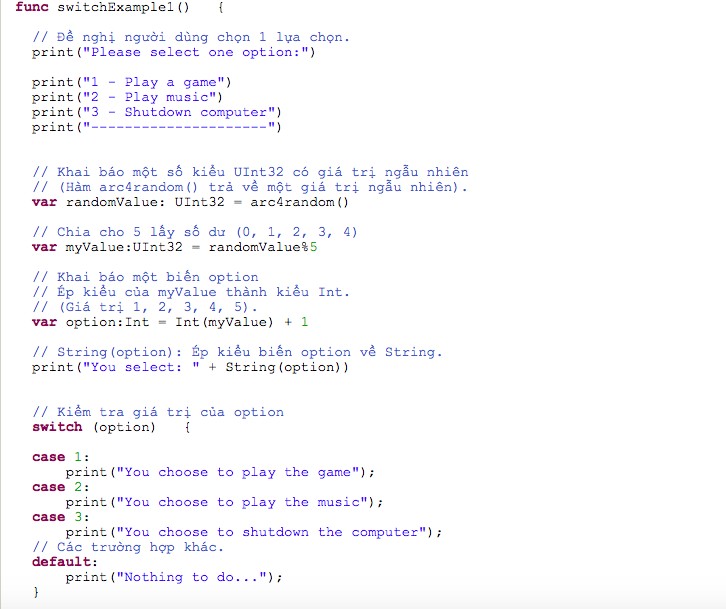
### Hình 6: Bảng toán tử trong Swift

Cú pháp của If – else



### Hình 7: Ví dụ If-Else trong Swift

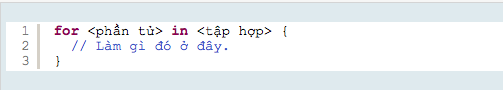
* Câu lệnh Switch – Case:



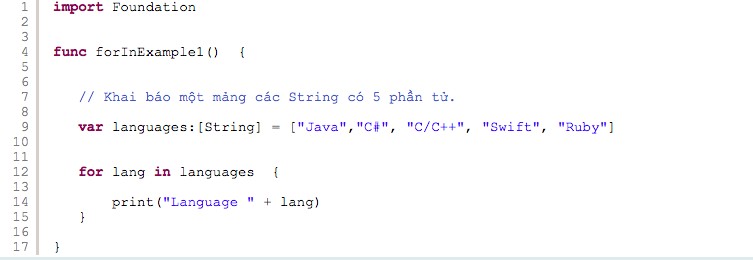
### Hình 8: Ví dụ Switch – Case trong swift

Bình thường đối với các ngôn ngữ khác như java, C#, C++ .. Sau mỗi trường hợp case phải có “break” còn đối với swift đã tự động break sau khi nó thực thi các lệnh trong Case

* + Vòng lặp trong Swift
* Vòng lặp for dung để duyệt trên một tập hợp hoặc một mảng

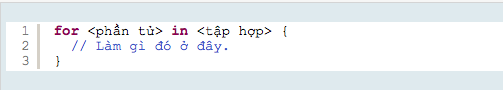


### Hình 9: Cú pháp for trong Swift



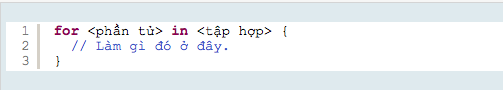
**Hình 10: Ví dụ vòng lặp for trong Swift**

* Vòng lặp While



### Hình 11: Cú pháp While trong Swift

* Vòng lặp repeat-while

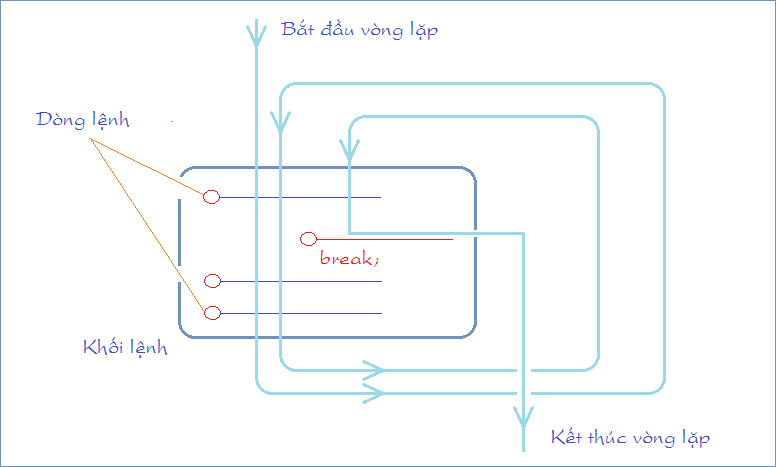


### Hình 12: Cú pháp repeat-while trong Swift

* + Lệnh break trong vòng lặp

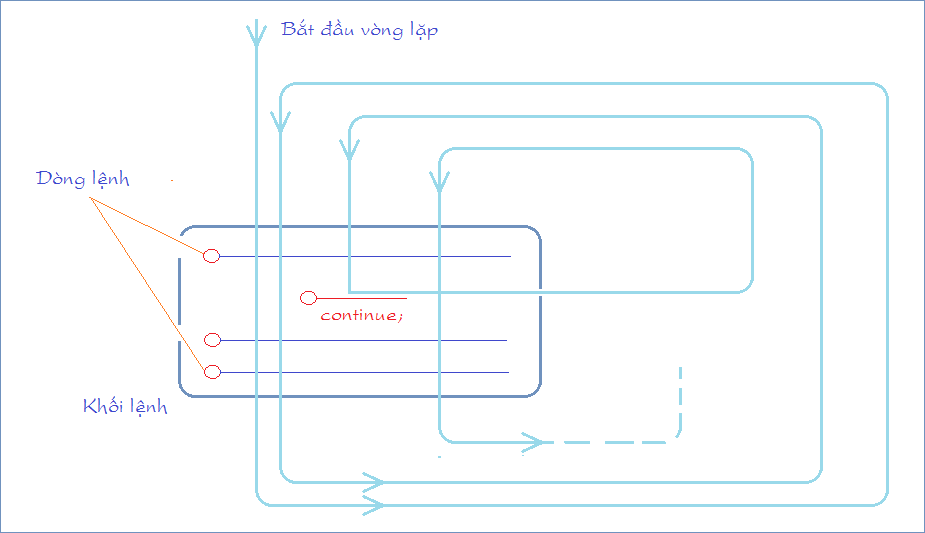
Break là một lênh nó có thể nằm trong một khối lệnh của một vòng lặp.

Đây là lệnh kết thúc vòng lặp vô điều kiện



**Hình 13: Mô hình Break trong vòng lặp**

* + Lệnh Continue trong vòng lặp

**Continue** là một lệnh, nó có thể nằm trong một vòng lặp, khi bắt gặp lệnh continue chương trình sẽ bỏ qua các dòng lệnh trong khối phía dưới của continue và bắt đầu một vòng lặp mới.

### Hình 14: Mô hình Continue trong vòng lặp

* Ngoài ra, còn có một số khái niệm phổ biến của Swift
* Mảng: Mảng của **Swift** về bản chất là một **Struct**, nó khác với một mảng trong các ngôn ngữ khác như **Java**, **C#**,...Mảng của **Java**, **C#** có số phần tử cố định và không thể thêm hoặc bớt số phần tử, trong khi đó mảng của **Swift** lại làm được điều này.

Vì nó là một struct nên trong mảng có thể khai báo phương thức

* Chuổi và một số thuộc tính khác

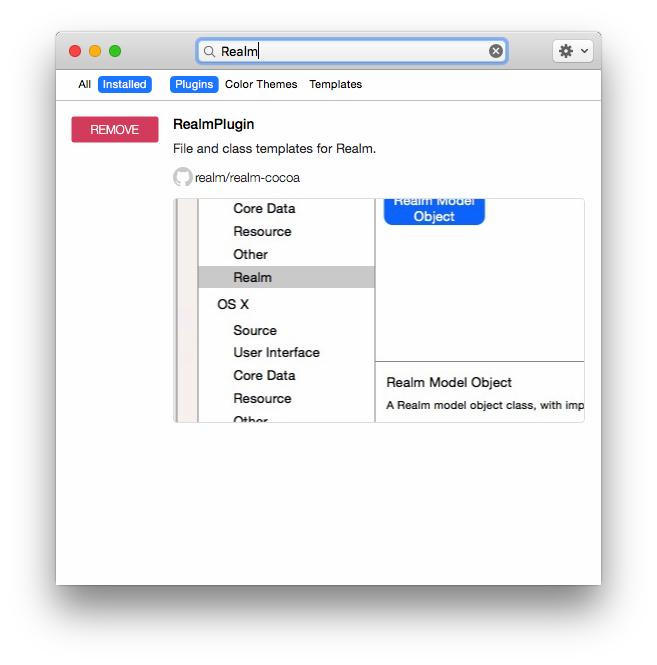
## Cơ bản về Realm và RealmSwift

## Khái niệm Realm và RealmSwift

* **Realm** là một DB sử dụng core C++ với tham vọng thay thế SQLite với các ưu điểm về tốc độ và dễ sử dụng
* **RealmSwift** là một Framework hỗ trợ việc thao tác với Realm DB với ngôn ngữ Swift, cũng giống như CoreData vậy

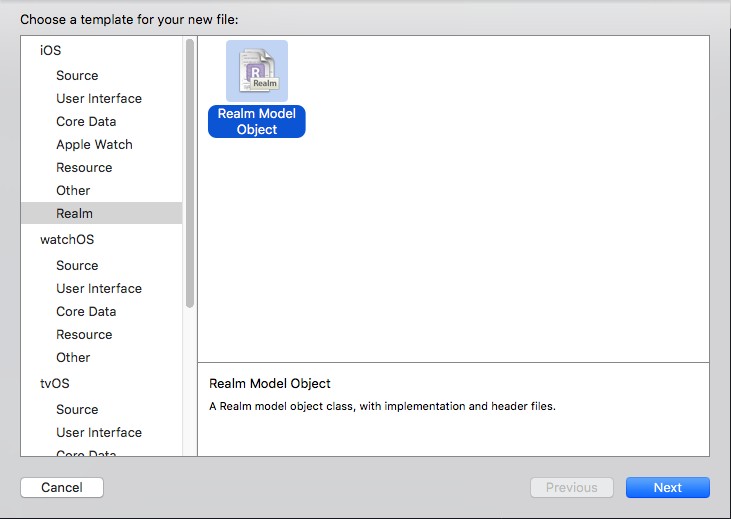
## Một số công cụ hỗ trợ

**- Plugin cho Xcode:** giúp tiết kiệm thời gian cho việc tạo một Realm Object subclass



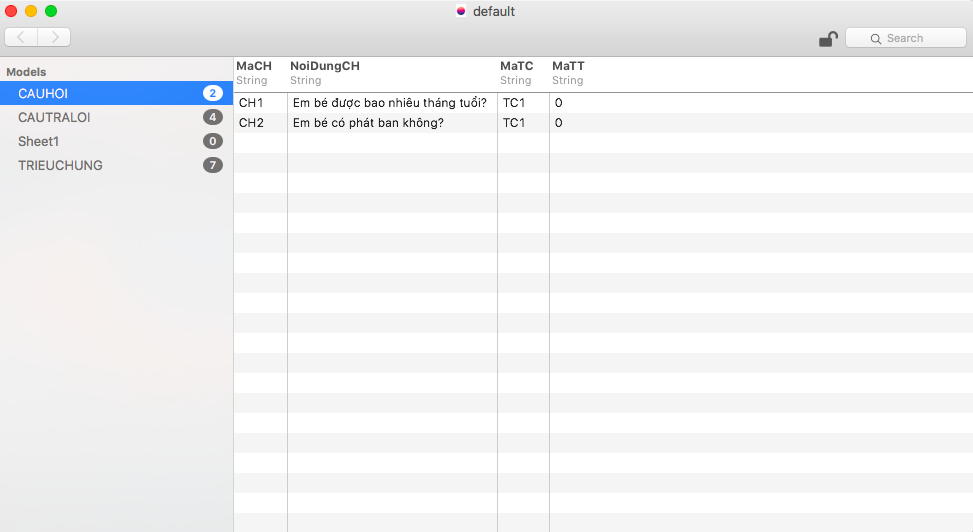
### Hình 15: Realm Plugin trong Xcode

Sau khi cài đặt Plugin, restart Xcode thì ta sẽ được



### Hình 16: Realm File trong Xcode

**- Realm Browser:** Đây là phần mềm hiển thị giao diện trục quan, dễ dàng sửa đổi, tạo mới DB.

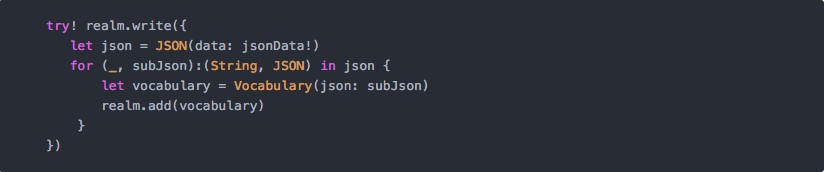


### Hình 17: Realm Browser trong Xcode

## Cách sử dụng Realm và RealmSwift

* + - * **Setup Realm :** Cách add Realm và RealmSwift rất đơn giản, chỉ cần download fameword Realm về và add bằng cách kéo vào project

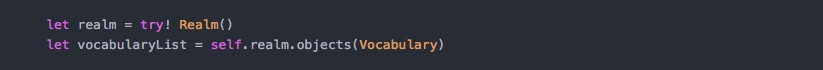
### Thao tác cơ bản với DB:

* Ghi dữ liệu vào Realm, chúng ta sẽ đọc từ file JSON để import initialize data

### Hình 18: Ghi dữ liệu vào Realm

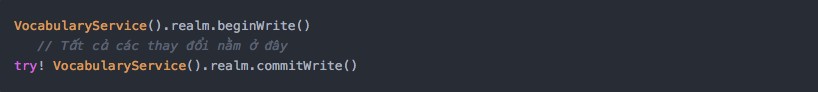
* Hoặc có thể copy từ 1 file Realm

### Hình 19: Ghi dữ liệu vào Realm từ 1 file Realm

* Đọc data

### Hình 20: Đọc dữ liệu vào Realm

* Edit, update, xoá dữ liệu



### Hình 21: Thêm sửa xóa dữ liệu vào Realm

## CHƢƠNG 3. PHÂN TÍCH THIẾT KẾ XÂY DỰNG ỨNG DỤNG HEALTH OF CHILD

## PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU

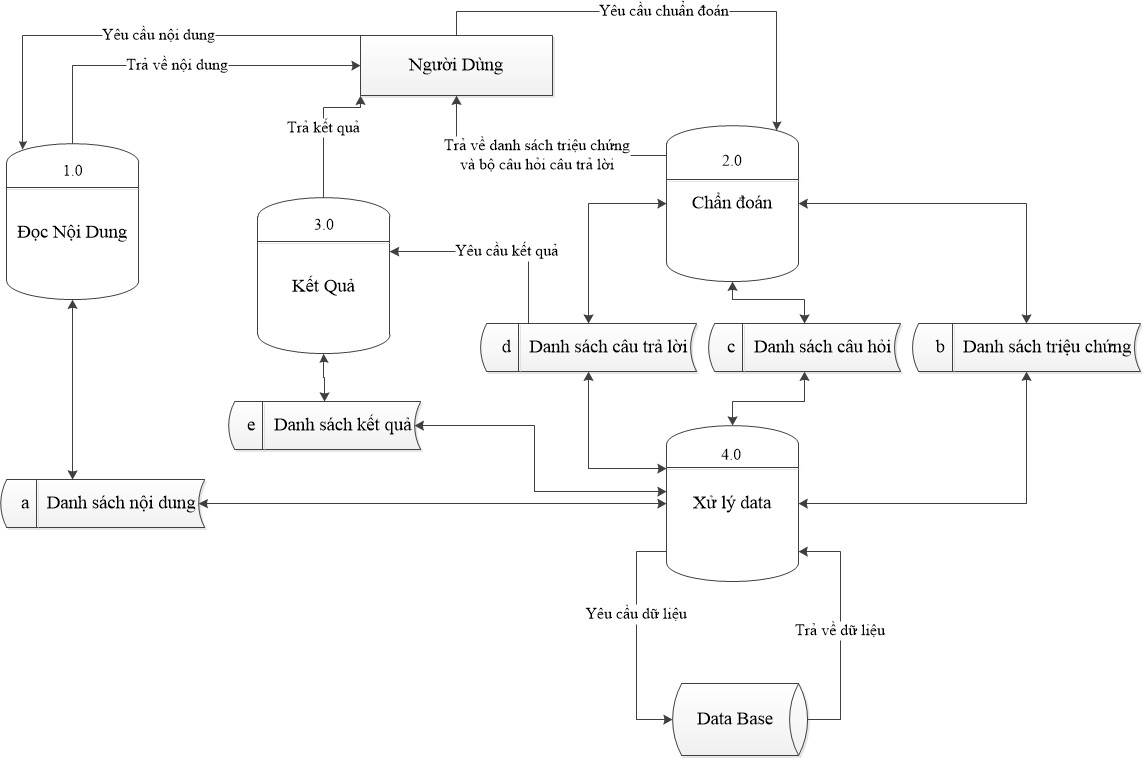
## Mô hình ngữ cảnh

****

**Hình 22: Mô hình ngữ cảnh**

## Mô hình DFD các mức

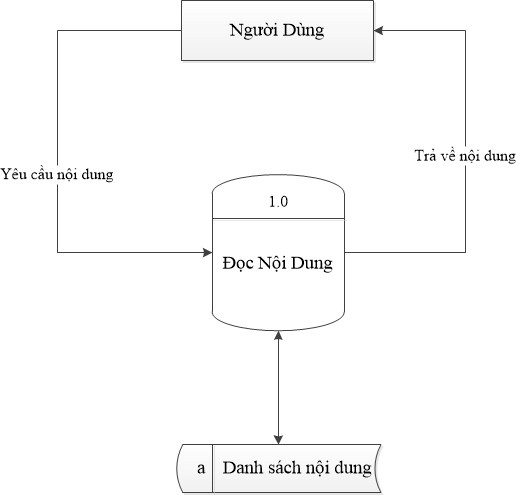
## Mô hình DFD mức 0

****

**Hình 23: Mô hình DFD mức 0**

## Mô hình DFD mức 1:

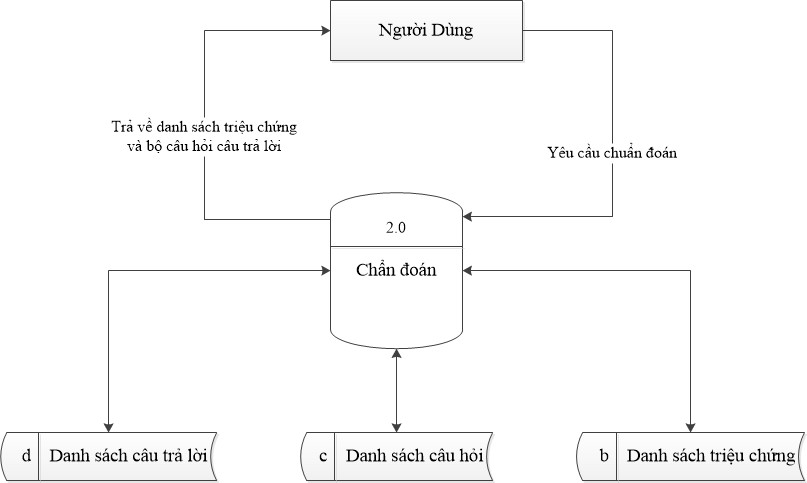
DFD nội dung



### Hình 24: Mô hình DFD mức 1

## Mô hình DFD mức 2

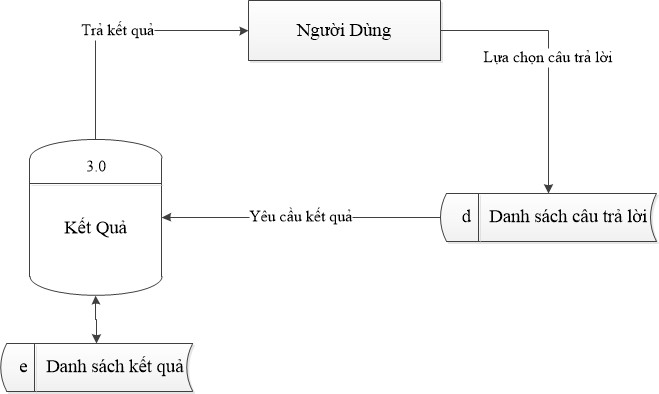
DFD chuẩn đoán



### Hình 25: Mô hình DFD mức 2

## Mô hình DFD mức 3

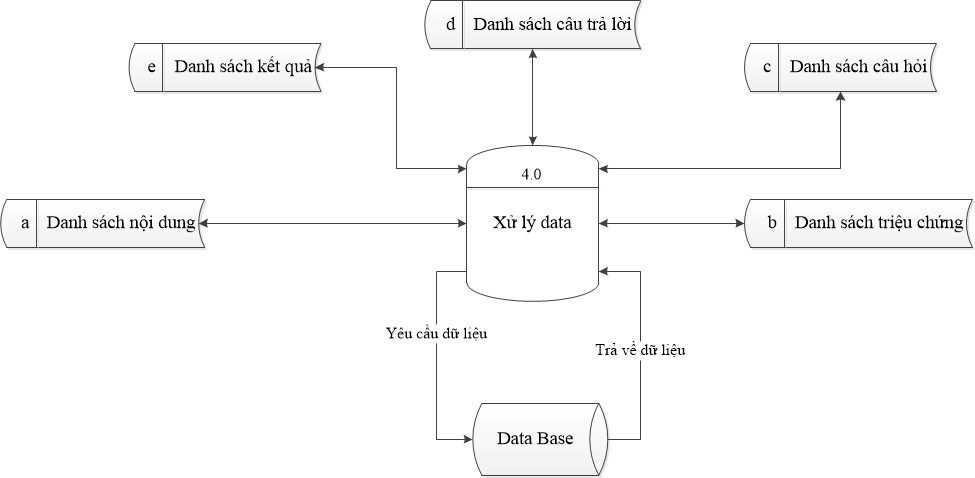
DFD kết quả



### Hình 26: Mô hình DFD mức 3

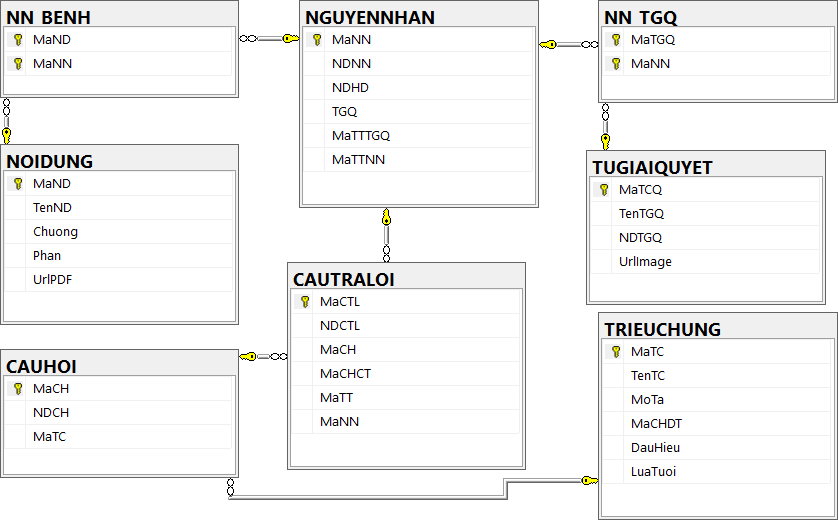
## Mô hình DFD mức 4

DFD Data Base



### Hình 27: Mô hình DFD mức 4

* + 1. **Lƣợc đồ Diagram**

****

**Hình 28: Lƣợc đồ Diagram**

* + 1. **Lƣợc đồ CSDL**
       - Bảng TRIEUCHUNG chứa tất cả các triệu chứng của trẻ
* MaTC: Khoá chính, mã của triệu chứng có dạng “TC1,TC2..TCn”
* TenTC: Tên của triệu chứng
* MoTa: Một đoạn ngắn mô tả về triệu chứng
* MaCHDT: Mã câu hỏi bắt đầu đầu tiên của từng triệu chứng
* DauHieu: Dấu hiệu cảnh báo nguy hiểm của triệu chứng (có thể có hoặc không, nếu không có sẽ mang giá trị NULL)
* LuaTuoi: Có 3 lứa tuổi được đánh dấu:
  + 1: Trẻ em dưới 1 tuổi
  + 2: Trẻ em trên 1 tuổi
  + 3: Trẻ em các lứa tuổi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TRIEUCHUNG | | | | | |
| MaTC | TenTC | MoTa | MaCHDT | DauHieu | LuaTuoi |

* Bảng CAUHOI chứa tất cả các câu hỏi của triệu chứng
* MaCH: khoá chính, mã của câu hỏi, có dạng “CH1, CH2, … CHn”
* NDCH: nội dung của câu hỏi
* MaTC: khoá ngoại, mã của triệu chứng

|  |  |  |
| --- | --- | --- |
| CAUHOI | | |
| MaCH | NDCH | MaTC |

* Bảng CAUTRALOI chứa tất cả câu trả lời
* MaCTL: khoá chính, mã câu trả lời, có dạng “CTL1, CTL2, … CTLn”
* NDCTL: nội dung câu trả lời
* MaCH: khoá ngoại, mã câu hỏi
* MaCHCT: mã nội dung chuyển tiếp, có thể chuyển tiếp sang câu hỏi tiếp theo hoặc không
* MaNN: khoá ngoại, mã nguyên nhân
* MaTT: mã trạng thái, gồm có 3 trạng thái
  + 0: Không chuyển tiếp
  + 1: Chuyển tiếp tới kết luận
  + 2: Chuyển tới một triệu chứng khác

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CAUTRALOI | | | | | |
| MaCTL | NDCTL | MaCH | MaCHCT | MaTT | MaNN |

* Bảng NGUYENNHAN chứa nội dung nguyên nhân, hành động của triệu chứng
* MaNN: mã nguyên nhâ hành động, có dạng “NN1, NN2, … NNn”
* NDNN: nội dung nguyên nhân
* NDHD: nội dung hành động cần làm
* TGQ: nội dung tự giải quyết ở nhà
* MaTTNN: chứa mã chuyển tiếp của nguyên nhân và kết quả, có thể chuyển tiếp qua 1 bệnh, có thể có giả trị NULL
* MaTTTGQ: chứa mã chuyển tiếp qua một tự giải quyết khác, có thể có giá trị NULL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NGUYENNHAN | | | | | |
| MaNN | NDNN | NDHD | TGQ | MaTTTGQ | MaTTNN |

* Bảng TUGIAIQUYET chứa nội dung tự giải quyết các bệnh tại nhà
* MaTGQ: mã tự giải quyết, có dạng “TGQ1, TGQ2, … TGQ3”
* TenTGQ: tên của tự giải quyết
* NDTGQ: nội dung của tự giải quyết
* UrlImage: tên file PDF

|  |  |  |  |
| --- | --- | --- | --- |
| TUGIAIQUYET | | | |
| MaTGQ | TenTGQ | NDTGQ | UrlImage |

* Bảng NOIDUNG: chứa tất cả nội dung kiến thức về bệnh
* MaND: Khóa chính, chứa Mã của nội dung, có dạng “ND1, ND2…NDn”
* TenND: tựa đề của nội dung
* Chuong: Có 3 chương, tương ứng là 1,3,4
* Phan: trong chương có từng phần nhỏ, số phần tùy thuộc vào số lượng mỗi chương
* UrlPDF: link file PDF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NOIDUNG | | | | |
| MaND | TenND | Chuong | Phan | UrlPDF |

* Bảng NN\_BENH: là bảng được sinh ra do kết nối nhiều nhiều của 2 bảng NGUYENNHAN và NOIDUNG

|  |  |
| --- | --- |
| NN\_BENH | |
| MaND | MaNN |

* Bảng NN\_TGQ: là bảng được sinh ra do kết nối nhiều nhiều của 2 bảng TUGIAIQUYET và NOIDUNG

|  |  |
| --- | --- |
| NN\_TGQ | |
| MaTGQ | MaNN |

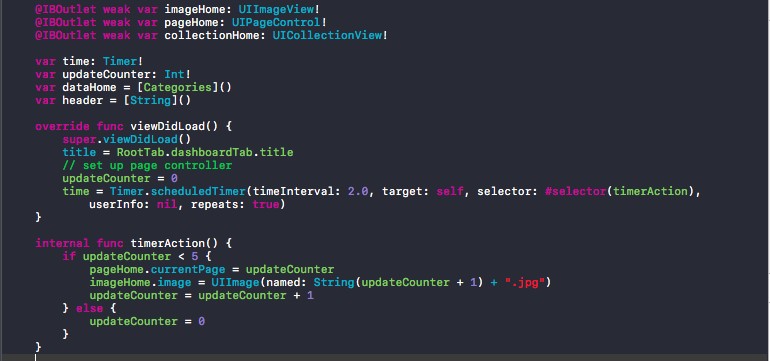
## XÂY DỰNG ỨNG DỤNG HEALTH OF CHILD

Ứng dụng được xây dựng theo mô hình MVC

## Màn hình Home

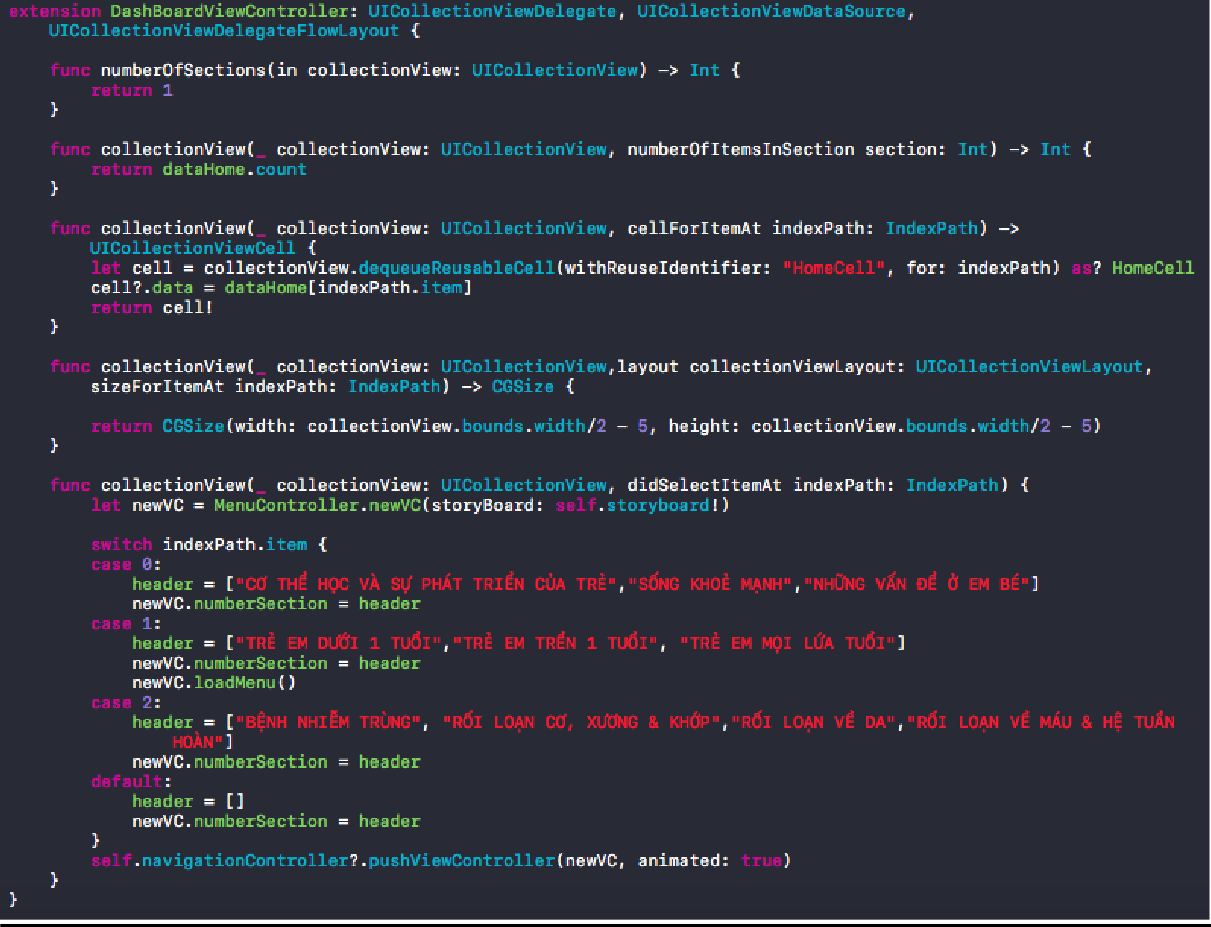
Sử dụng UICollectionView để hiển thị nội dung từng danh mục kèm theo UIPageController và UIImageView để hiển thị giao diện

* + - * UIImageView: Hiển thị một mảng hình ảnh được gán mặc định theo tên hình ảnh
      * UIPageController: Set tự động chuyển hình ảnh sau 2s.



### Hình 29: Source Code Giao Diện Màn hình Home

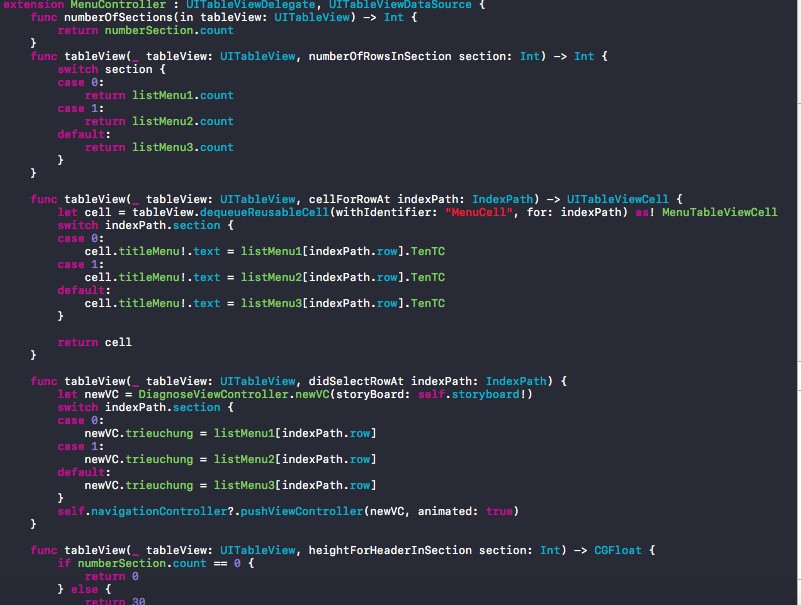
* + - * UICollectionView: hiển thị các danh mục và action chọn



### Hình 30: Source Code Danh mục Màn hình Home

* + 1. **Màn hình Menu**

Dùng UITableView để quản lý Menu, tuỳ vào DB đọc lên mà ta cho hiển thị lên tableView. Khi người dung click vào từng row của table thì sẽ được chuyển trang sang view mới tương ứng.



### Hình 31: Source Code Danh Dách nội dung triệu chứng màn hình Menu

## Màn hình Chuẩn đoán

Màn hình chuẩn đoán chia làm 3 UIViewController, bao gồm:

* + - * View mô tả bệnh: mô tả bệnh và bắt đầu vào chuẩn đoán
      * View chuẩn đoán: hiển thị câu hỏi và câu trả lời theo thao tác người dung
      * View kết quả: hiển thị kết quả cho người dung Và tất cả đều được quản lý bởi tableView

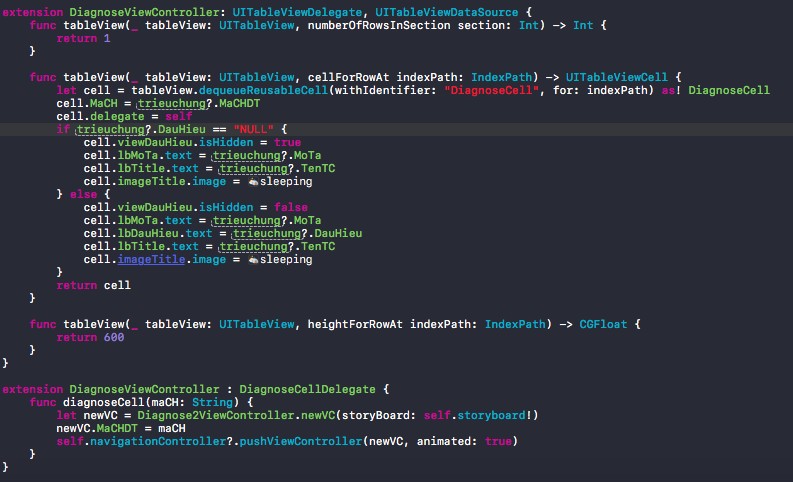
**Mô tả bệnh**: Đối với button chuyển trang, dung Delegate để xác định đang đúng ở triệu chứng nào

Set Delegate cho button:



### Hình 32: Mô hình Delegate

Ở ViewController ta set layout và gọi delegate ra thực thi



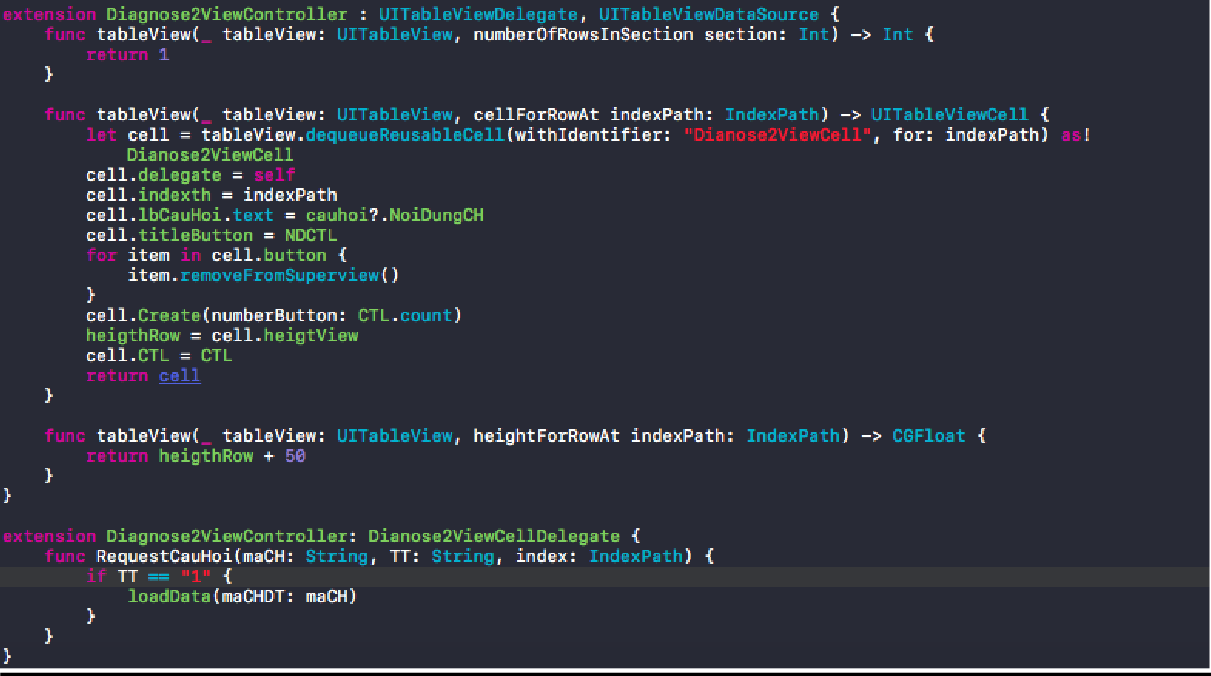
### Hình 33: Gọi và thực thi Delegate

**Chuẩn đoán bệnh:** Khi người dung bấm vào button “Bắt Đầu” sẽ hiển thị lên bộ câu hỏi và câu trả lời, những câu trả lời cũng được sử dụng Delegate để xác định xem câu trả lời đang là câu nào

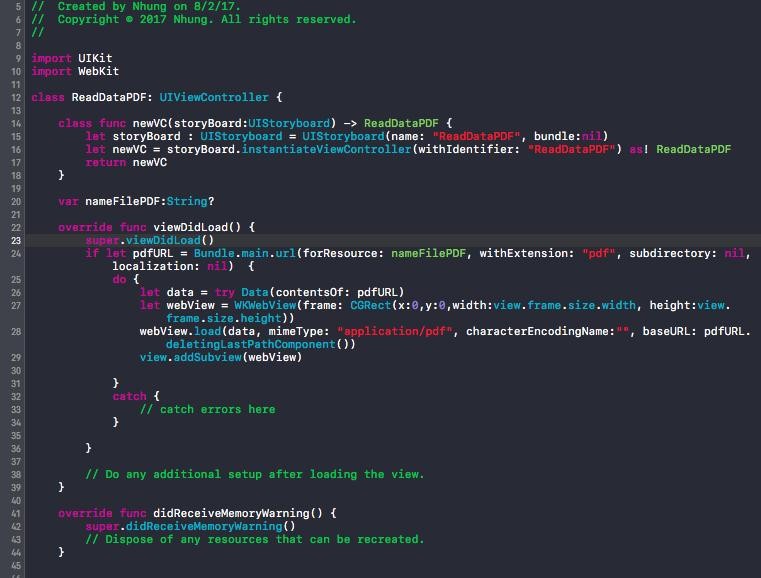
Delegate câu trả lời:



### Hình 34: Delegate Câu hỏi và câu trả lời



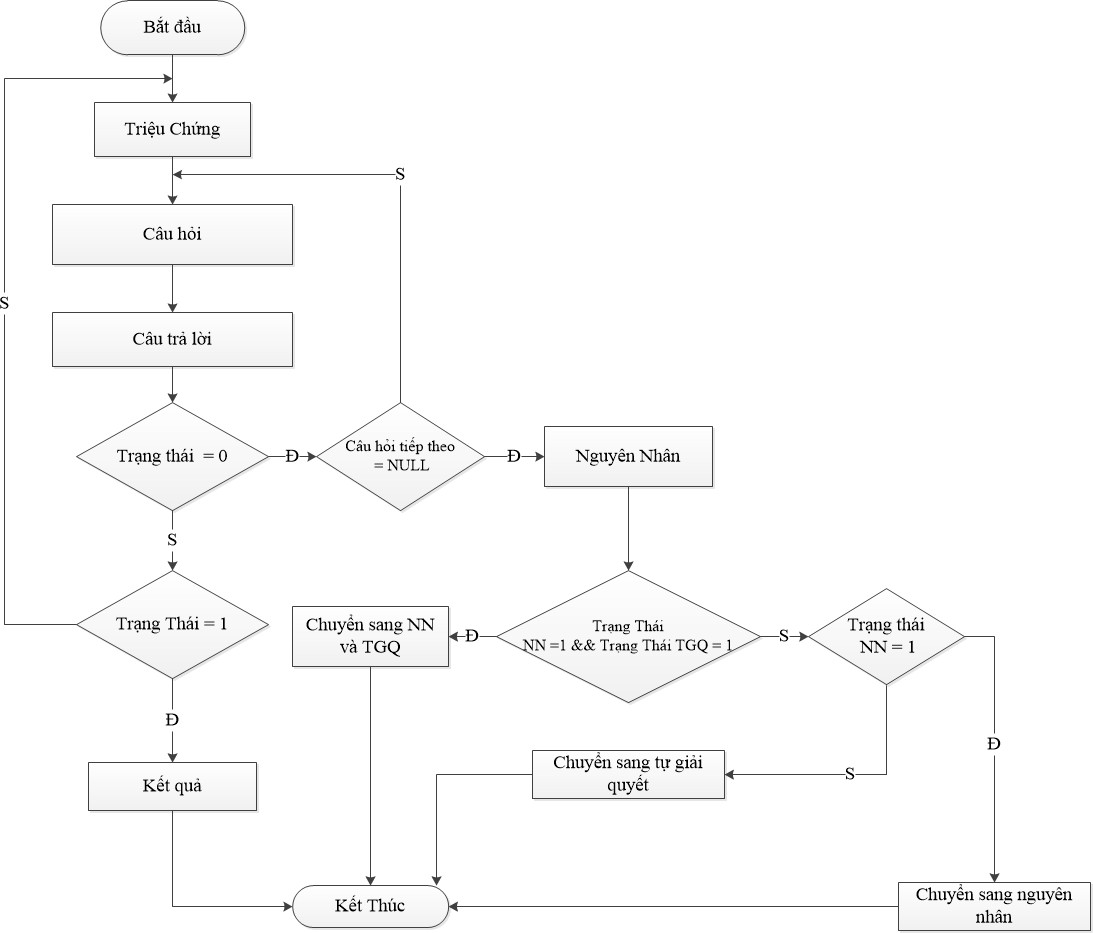
**Hình 35: Xử lý câu hỏi và câu trả lời**

* + 1. **Màn hình Read** Khi người dùng nhập vào những bệnh hoặc triệu chứng thì nội dung cần hiển thị được hiển thị theo file PDF.

### Hình 36: Cách đọc file PDF

* + 1. **Màn hình Cause** Màn hình show nguyên nhân và hành động có thể về chuẩn đoán bệnh thường gặp. Dùng delegate để cho biết chuyển tiếp là bệnh có thể hoặc tự giải quyết tại nhà.

## 3.3 Thuật toán chuẩn đoán

****

### Hình 37: Thuật toán chuẩn đoán

## CHƢƠNG 4. GIỚI THIỆU ỨNG DỤNG HEALTH OF CHILD

## Giới thiệu ứng dụng Health Of Child

Ứng dụng chẩn đoán bệnh ở trẻ được đặt tên gọi là Heigth of Child. Là phần mềm dành cho tất cả mọi người, nhất là những bậc bố mẹ có con nhỏ có thể tìm hiểu, bổ sung kiến thức chăm sóc trẻ và chẩn đoán một vài bệnh thường gặp ở trẻ

## Các chức năng của ứng dụng

* + - Màn hình Home:



### Hình 38: Màn hình Home

* + - Màn hình Menu: Khi người dung nhập vào 1 trong những danh mục ở trên Home thì sẽ xuất hiện màn hình này

Menu tìm hiểu cơ thể trẻ Menu các triệu chứng của trẻ



### Hình 39: Màn hình Menu

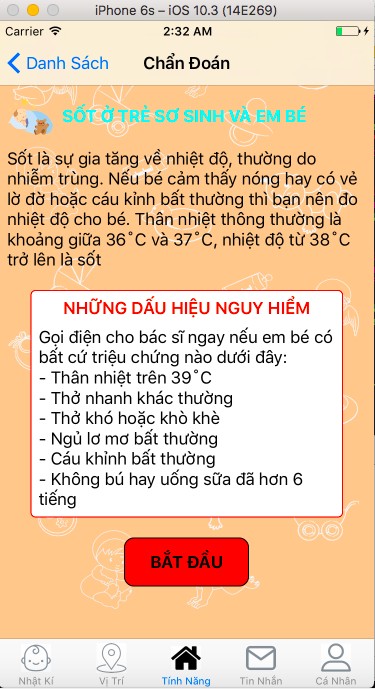
* + - Thông tin cơ thể trẻ: Ở Menu tìm hiểu cơ thể trẻ, khi người dùng click vào thông tin ở trên sẽ hiển thị màn hình đọc



### Hình 40: Màn hình đọc nội dung

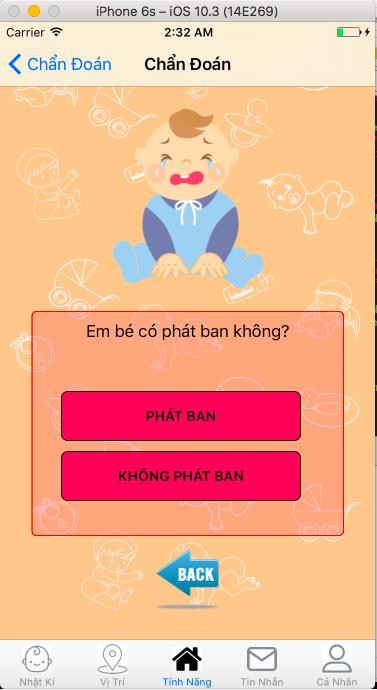
* + - Chẩn đoán bệnh: Trong Menu các triệu chứng bệnh, khi click vào một trong những bệnh trên sẽ show các màn hình cho phép chuẩn đoán bệnh của trẻ

Màn hình khi bắt đầu chuẩn đoán:



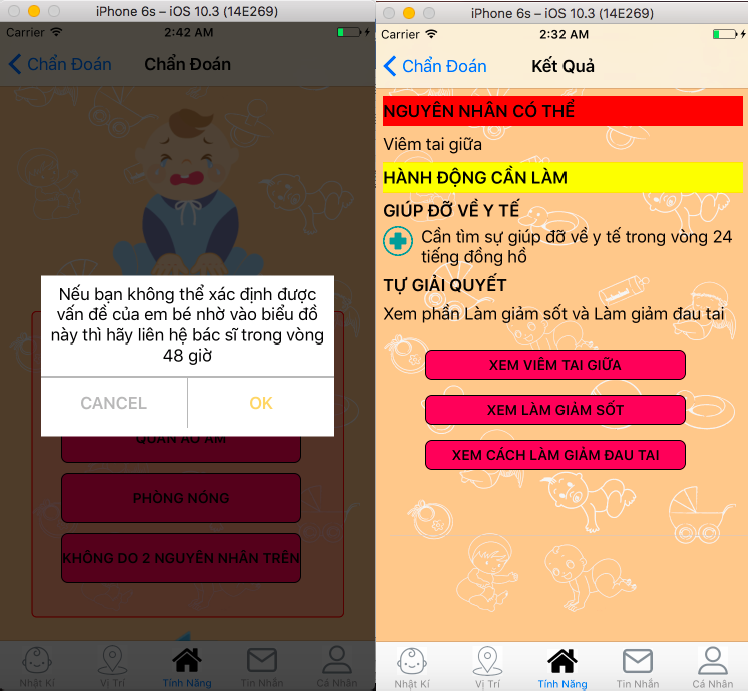
### Hình 41: Màn hình mô tả triệu chứng

Khi bấm bắt đầu sẽ show ra lần lượt những câu hỏi và câu trả lời đi kèm

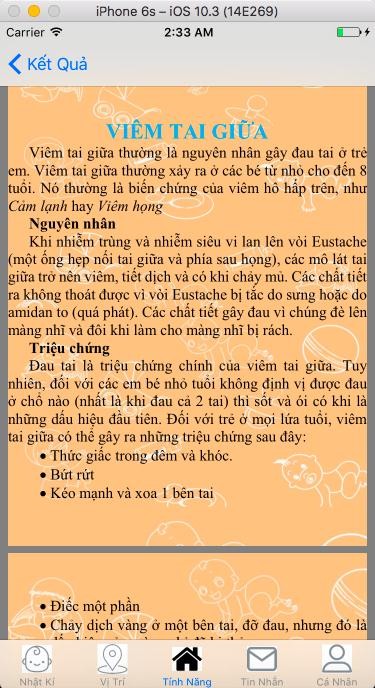


### Hình 42: Màn hình chuẩn đoán

Kết quả sẽ show ra kết quả hoặc nguyên nhân và tự giải quyết



### Hình 43: Màn hình kết quả



**Hình 44: Màn hình nguyên nhân và tự giải quyết**

## CHƢƠNG 5. KẾT LUẬN VÀ HƢỚNG PHÁT TRIỂN

## Kết luận

- Kết quả đạt được

+ Lý thuyết:

* + - Sử dụng thành thạo công cụ XCODE và một số công cụ hỗ trợ khác trên IOS như Simulator, Terminal, …
    - Nắm rõ ngôn ngữ Swift trong lập trình di động
    - Sử dụng được Realm để lưu DB ở device

+ Thực hành: tạo được ứng dụng Health of Child với đầy đủ các chức năng đề

ra.

* Chứa những bài đọc bổ ích về sức khoẻ trẻ em cho người dùng
* Chẩn đoán bệnh và cách triều trị cho trẻ
* Truy xuất, xử lí dữ liệu giữa ứng dụng với database bằng Realm

- Hạn chế:

+ Giao diện ứng dụng chưa bắt mắt người dung.

+ Chưa sử dụng được hết triệt để các chức năng và hỗ trợ Xcode và Realm.

+ DB chưa được nhập liệu đầy đủ

## Hƣớng phát triển

**5.2.1 Hƣớng khắc phục**

+ Mở rộng thêm một số chức năng về user như cho người dùng đăng nhập lưu lại lịch sử người dùng.

+ Tối ưu truy xuất dữ liệu.

+ Liên kết thêm với các mạng xã hội như FaceBook, Gmail API, Zalo API và Twitter API để chia sẽ dữ liệu với nhau

## Hƣớng mở rộng

+Thêm chức năng Phòng chat, call hoặc gọi video giữa người dùng vs bác sĩ tư vấn trực tiếp

+Tạo thêm trang xã hội để người dung có thể trao đổi thông tin với nhau.

## DANH MỤC TÀI LIỆU THAM KHẢO:

### Tiếng Việt:

* + - 1. Giáo trình Lập trình iOS trung tâm tin học KHTN .

### Tiếng Anh:

1. [iOS 7 Application Development in 24 Hours](http://www.amazon.com/Application-Development-Hours-Yourself-Edition/dp/0672337061/) .

### Website:

1. <https://developer.apple.com/>
2. <http://www.appcoda.com/tag/objective-c/>
3. <https://www.codeschool.com/courses/try-objective-c>
4. <https://github.com/>
5. <http://stackoverflow.com/>
6. [https://tinhte.vn](https://tinhte.vn/)