**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**NIÊN LUẬN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**



**Đề tài**

**XÂY DỰNG ỨNG DỤNG THI TRẮC NGHIỆM TRÊN IOS BẰNG NGÔN NGỮ SWIFT**

Người hướng dẫn:Sinh viên thực hiện:

**TS/Ths Trần Công Án Nguyễn Khắc Sâm Thương**

MSSV: **B1410602**

Khoá: **40**

***Hoà An, 5/2018***

# LỜI CẢM ƠN

Trong suốt thời gian từ khi bắt đầu thực hiện đề tài niên luận đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ từ thầy cô và bạn bè.

Em xin cảm ơn thầy **Trần Công Án** – giảng viên hướng dẫn trong suốt quá trình thực hiện đề tài, đã cho em cơ hội được tiếp cận và tìm hiểu những vấn đề thực tế để có thể áp dụng kĩ năng vào thực tiễn, giúp em có thêm nhiều kinh nghiệm giải quyết vấn đề và kiến thức về công nghệ mà đề tài nghiên cứu. Đây là những nền tảng vô cùng cần thiết cho em khi nghiên cứu luận văn và sau khi ra trường.

Xin cảm ơn những người bạn ở lớp **Công Nghệ Thông Tin K40 – Khoa Phát Triển Nông Thôn – Trường Đại Học Cần Thơ**, đã cung cấp nhiều thông tin, tài liệu hữu ích, hỗ trợ khi em gặp khó khăn, giúp em hoàn thành đề tài niên luận này.

Và cuối cùng, xin gửi lời cảm ơn đến gia đình, những người đã tạo điều kiện, giúp đỡ em trong khoảng thời gian nghiên cứu và luôn luôn động viên, khích lệ tinh thần em trong suốt quá trình thực hiện đề tài niên luận.

Xin chân thành cảm ơn!

MụC LỤC

[LỜI CẢM ƠN 1](#_Toc514050201)

[GIỚI THIỆU 3](#_Toc514050202)

[I. Đặt vấn đề 3](#_Toc514050203)

[I. Mục tiêu đề tài 3](#_Toc514050204)

[II. Phạm vi nghiên cứu 4](#_Toc514050205)

[III. Nội dung nghiên cứu 4](#_Toc514050206)

[NỘI DUNG 5](#_Toc514050207)

[I. Cơ sở dữ liệu của ứng dụng 5](#_Toc514050208)

[**1.** **Sơ đồ Use case:** 5](#_Toc514050209)

[**2.** **Sơ đồ các đối tượng và quan hệ giữa các đối tượng** 5](#_Toc514050210)

[**3.** **Thiết lập và cài đặt cơ sở dữ liệu** 6](#_Toc514050211)

[**4.** **Cài RMD cho ứng dụng, nhập dữ liệu vào cơ sở dữ liệu** 8](#_Toc514050212)

[II. Phân tích ứng dụng 12](#_Toc514050213)

[**1.** **Sơ đồ Main.storyboard** 12](#_Toc514050214)

[**2.** **Phân tích các View Controllers** 13](#_Toc514050215)

[III.Kết quả thực hiện và kiểm thử 31](#_Toc514050216)

[IV. Kết luận 35](#_Toc514050217)

[**1.** **Nhận xét về ứng dụng** 35](#_Toc514050218)

[**2.** **Hướng cải thiện và phát triển cho ứng dụng** 35](#_Toc514050219)

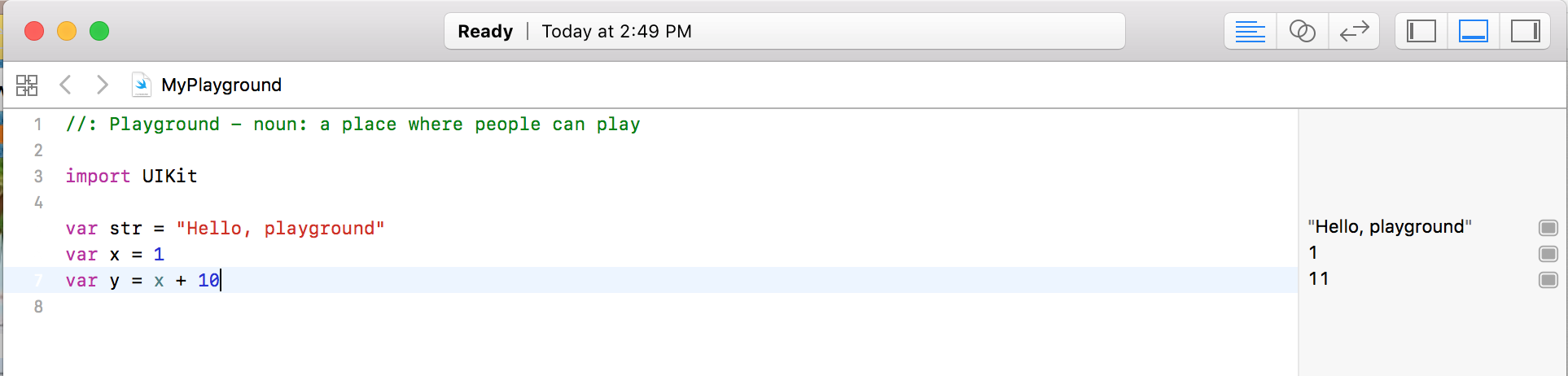
[TÀI LIỆU THAM KHẢO 36](#_Toc514050220)

# GIỚI THIỆU

1. **Đặt vấn đề**

Ngoài thời gian học ở lớp, tự học là cách củng cố và tiếp thu kiến thức hiệu quả. Với sự phát triển mạnh mẽ của smart phone và các thiết bị di dộng, việc tự học trở nên thuận tiện, thú vị hơn. Người học có thể chủ động tận dụng thời gian rảnh để học, giảm rất nhiều sự phụ thuộc vào sách vở hay cơ sở vật chất.

**Swift** là một ngôn ngữ lập trình hướng đối tượng dành cho việc phát triển iOS và OS X, được giới thiệu bởi Apple tại hội nghị WWDC năm 2014. Swift được thiết kế là một ngôn ngữ có khả năng phòng chống lỗi cao, được biên dịch bởi trình biên dịch LLVM. Một điểm thú vị của Swift được Apple trình diễn ngay tại WWDC 2014 là khả năng hiển thị kết quả theo thời gian thực. Khi viết xong bất kì đoạn mã nào, kết quả sẽ hiển thị ngay lập tức ở X Code Runtime, giúp lập trình viên theo dõi dễ dàng và tiến hành sửa lỗi. Tháng 3 năm 2017, chưa đầy 3 năm sau khi chính thức ra mắt, Swift đã đứng đầu trong bảng xếp hạng TIOBE hàng tháng dành cho các ngôn ngữ lập trình phổ biến nhất.

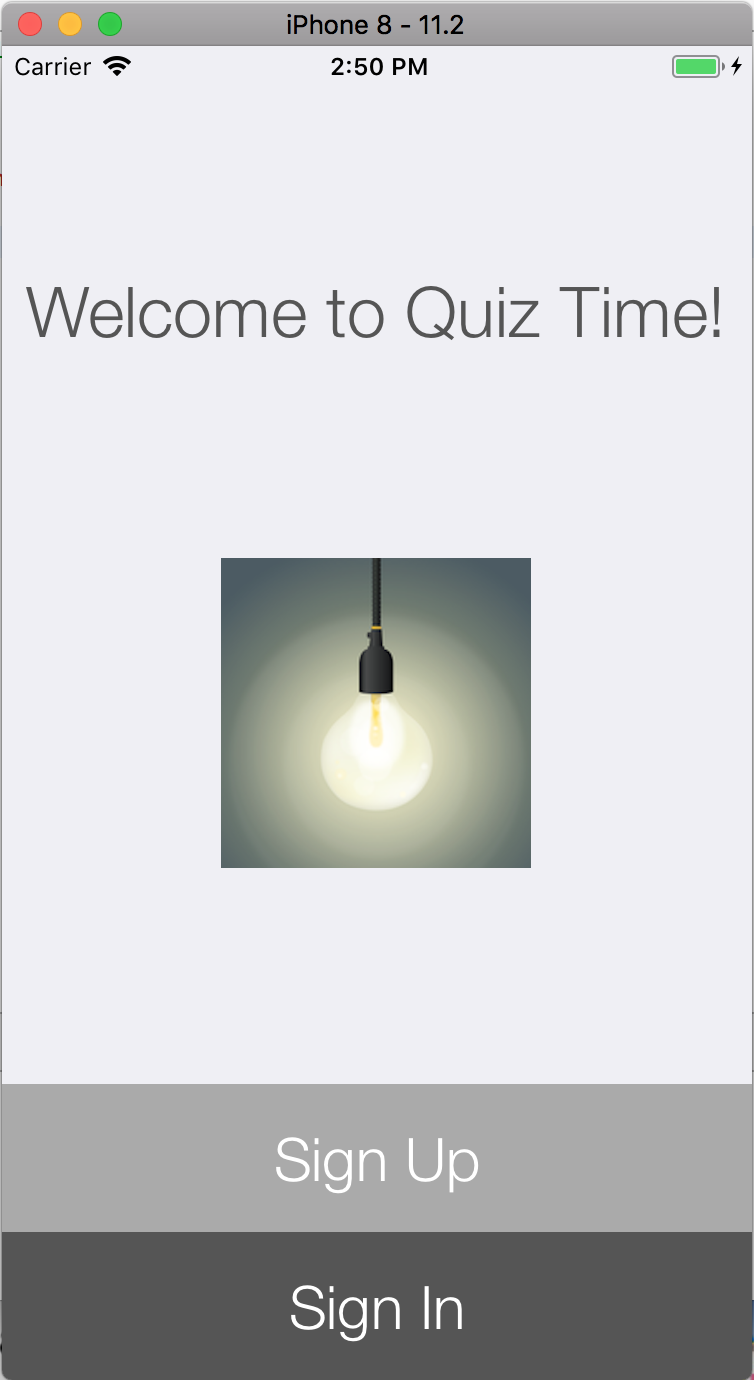


1. **Mục tiêu đề tài**

Với sự tiện lợi của ứng dụng di động và phát triển mạnh mẽ của Swift, em đã nghiên cứu và **xây dựng ứng dụng thi trắc nghiệm trên iOS bằng ngôn ngữ lập trình Swift**. Bên cạnh đó, ứng dụng sẽ được xây dựng dựa trên một cơ sở dữ liệu mới cho thiết bị di động: **Realm Mobile Database** (RMD). Realm cung cấp **Realm Browser** để theo dõi database dễ dàng và thuận tiện. Tốc độ truy vấn nhanh, đa nền tảng và được cấp miễn phí, đó là những lý do em chọn RMD cho ứng dụng này.

1. **Phạm vi nghiên cứu**

Đối với ứng dụng thi trắc nghiệm **QuizTime**, phạm vi nghiên cứu của đề tài là ứng dụng iOS được xây dựng bằng Swift, IDE là Xcode 9. Ứng dụng sử dụng offline database từ Realm Mobile Database cùng với sự hỗ trợ của Realm Browser để quản lý database.



## Nội dung nghiên cứu

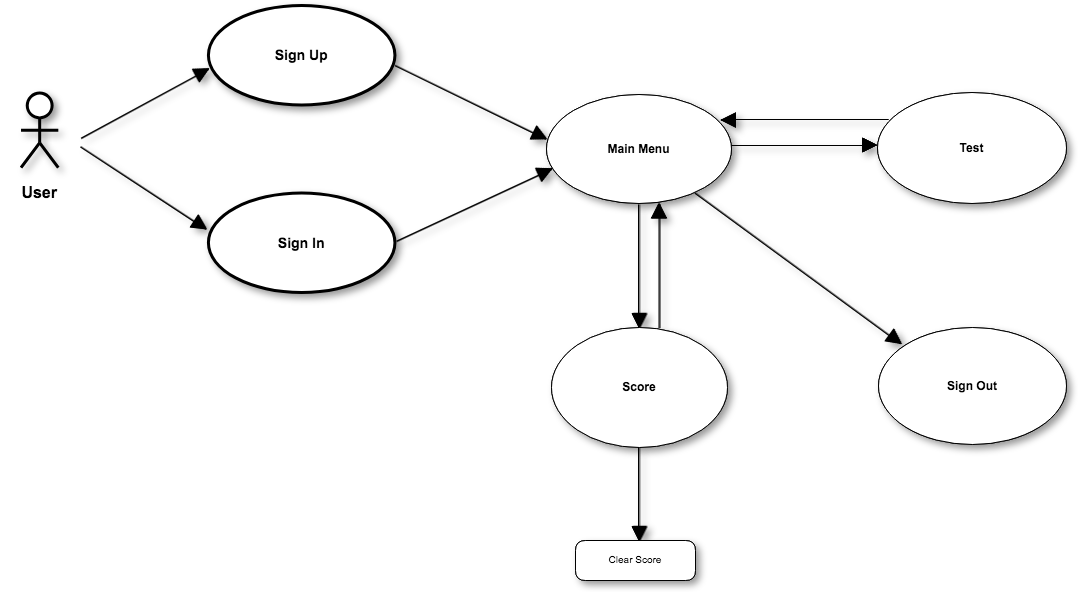
- Tìm hiểu ngôn ngữ lập trình Swift

- Cách xây dựng một ứng dụng iOS bằng Xcode

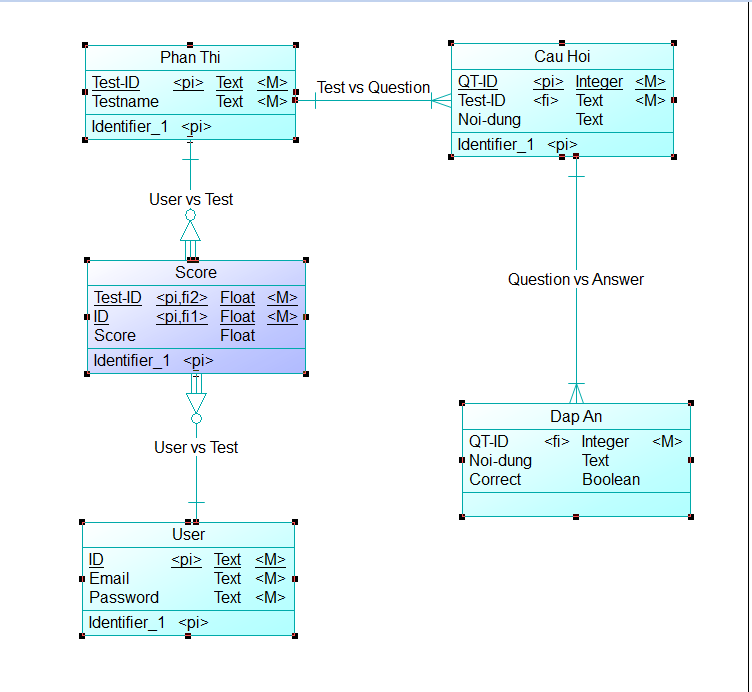
- Tìm hiểu Realm Mobile Database

# NỘI DUNG

1. **Cơ sở dữ liệu của ứng dụng**
2. **Sơ đồ Use case:**

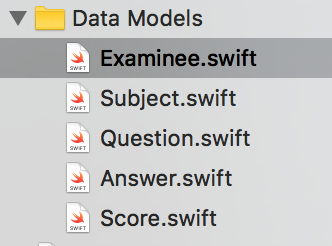
****

1. **Sơ đồ các đối tượng và quan hệ giữa các đối tượng**

****

1. **Thiết lập và cài đặt cơ sở dữ liệu**

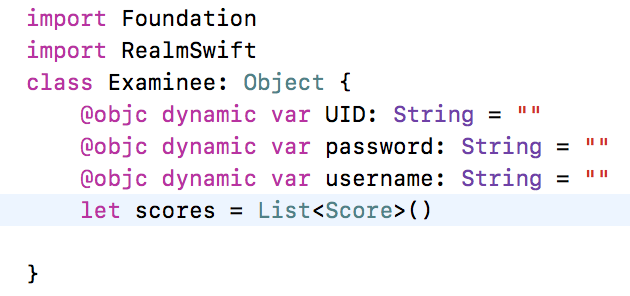
Mỗi đối tượng sẽ được tạo bằng một class trong source code.



1. **Đối tượng User:**

Các thuộc tính:

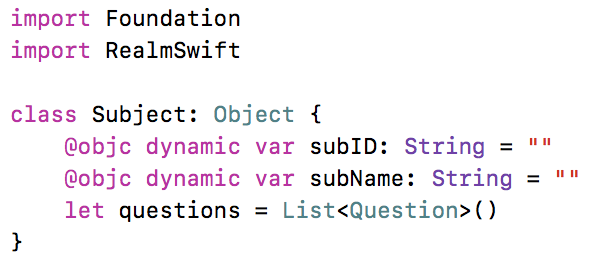
* **UID:** mã người dùng
* **username:** tên người dùng
* **password :** mật khẩu
* **scores:** danh sách điểm là một danh sách các đối tượng **Score**



* 1. **Đối tượng Subject:**

Các thuộc tính:

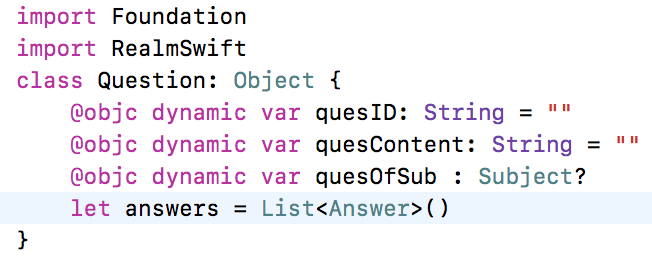
* **subID:** mã môn học
* **subName:** tên môn học
* **questions:** danh sách các câu hỏi là danh sách các đối tượng **Question**

****

* 1. **Đối tượng Question:**

Các thuộc tính:

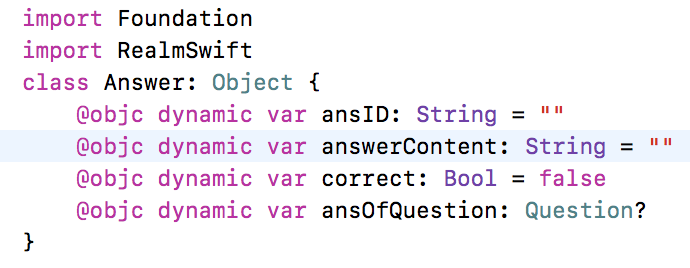
* **quesID**: mã câu hỏi
* **quesContent**: nội dung câu hỏi
* **quesOfSub:** câu hỏi thuộc môn học
* **answers:** danh sách các đáp án là danh sách các đối tượng **Answer**

****

* 1. **Đối tượng Answer:**

Các thuộc tính:

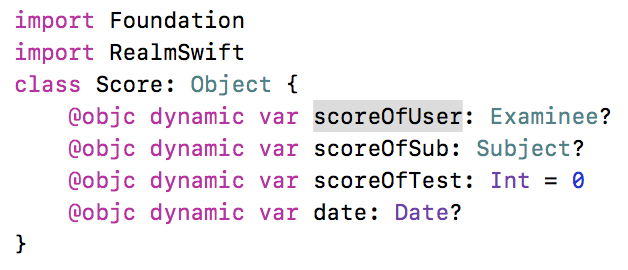
* **ansID**: mã đáp án
* **answerContent**: nội dung đáp áp
* **correct**: tính đúng sai
* **ansOfQuestion**: đáp án thuộc câu hỏi

****

* 1. **Đối tượng Score:**

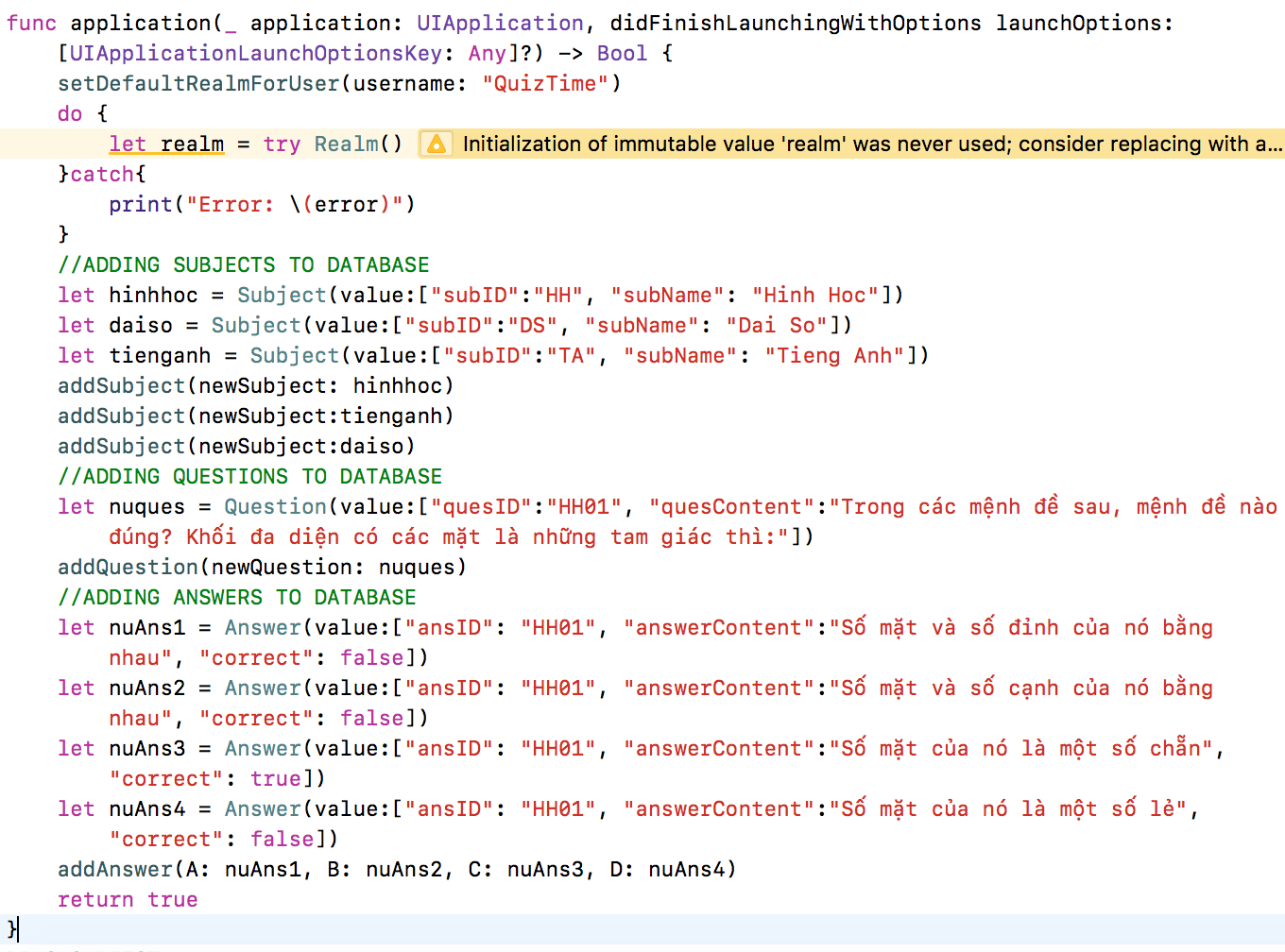
Các thuộc tính:

* **scoreOfUser**: điểm của người dùng
* **scoreOfSub:** điểm của môn học
* **scoreOfTest:** số điểm của bài kiểm tra
* **date:** thời gian hoàn thành bài kiểm tra

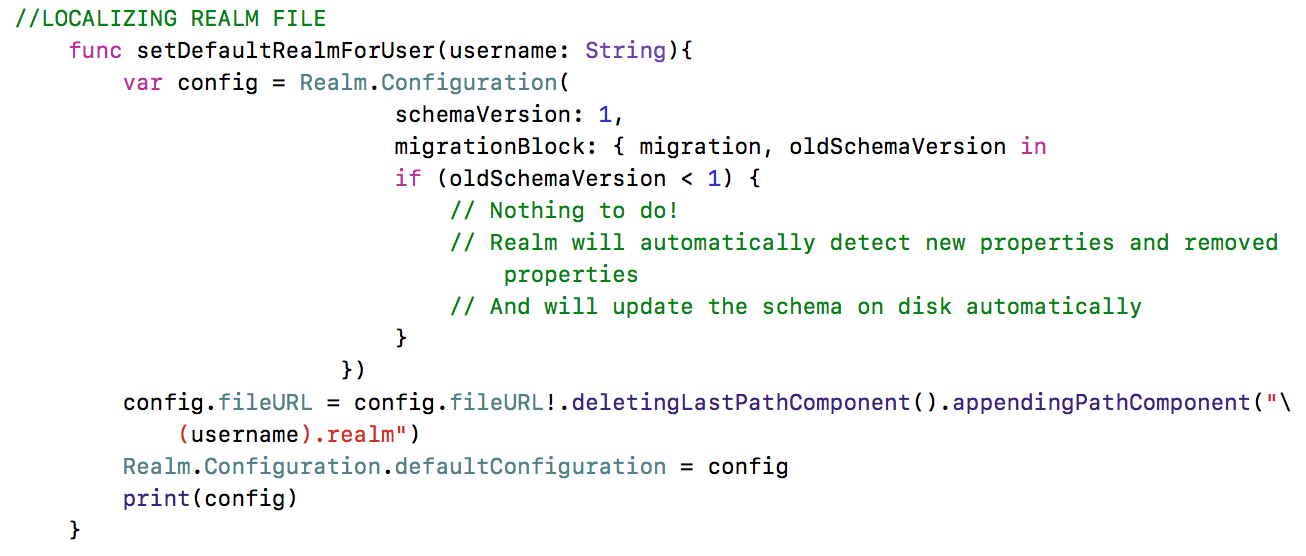


1. **Cài RMD cho ứng dụng, nhập dữ liệu vào cơ sở dữ liệu**

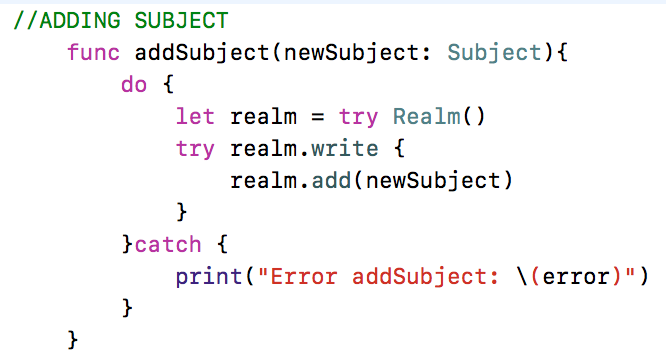
Sau khi cài đặt thư viện **Realm** và **RealmSwift** bằng **Cocoa Pod**, tại **AppDelegate.swift** (mặc định khi tạo project trong Xcode) trong **func application()**, cấu hình nơi lưu file **.realm** và thêm dữ liệu.



* **Hàm cấu hình cho Realm**



* **Hàm thêm Subject**

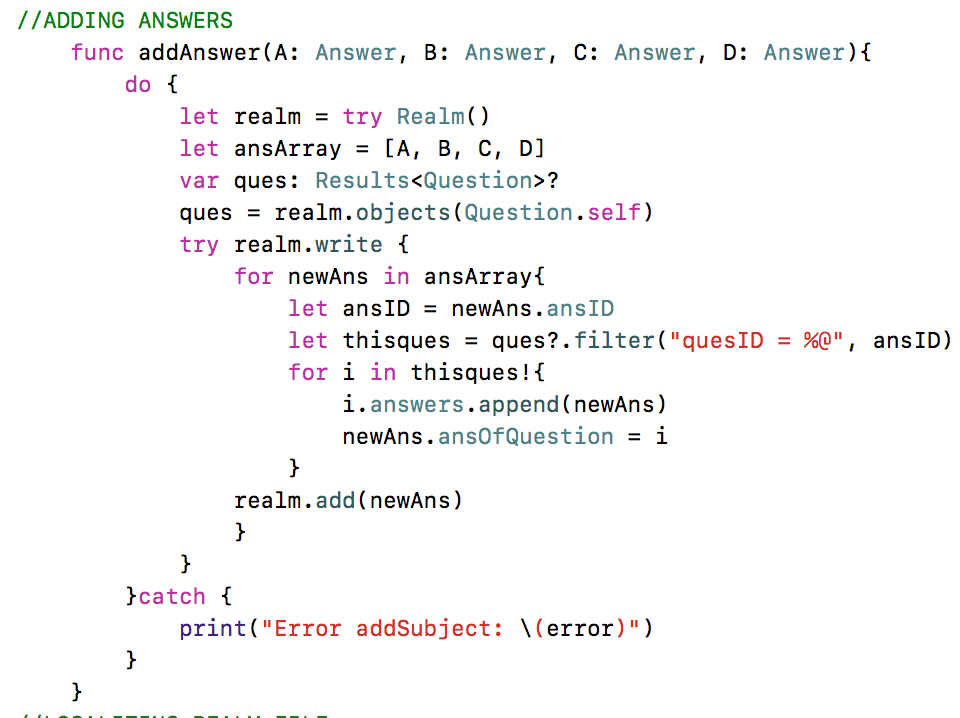


* **Hàm thêm Question**





* **Hàm thêm Answer**

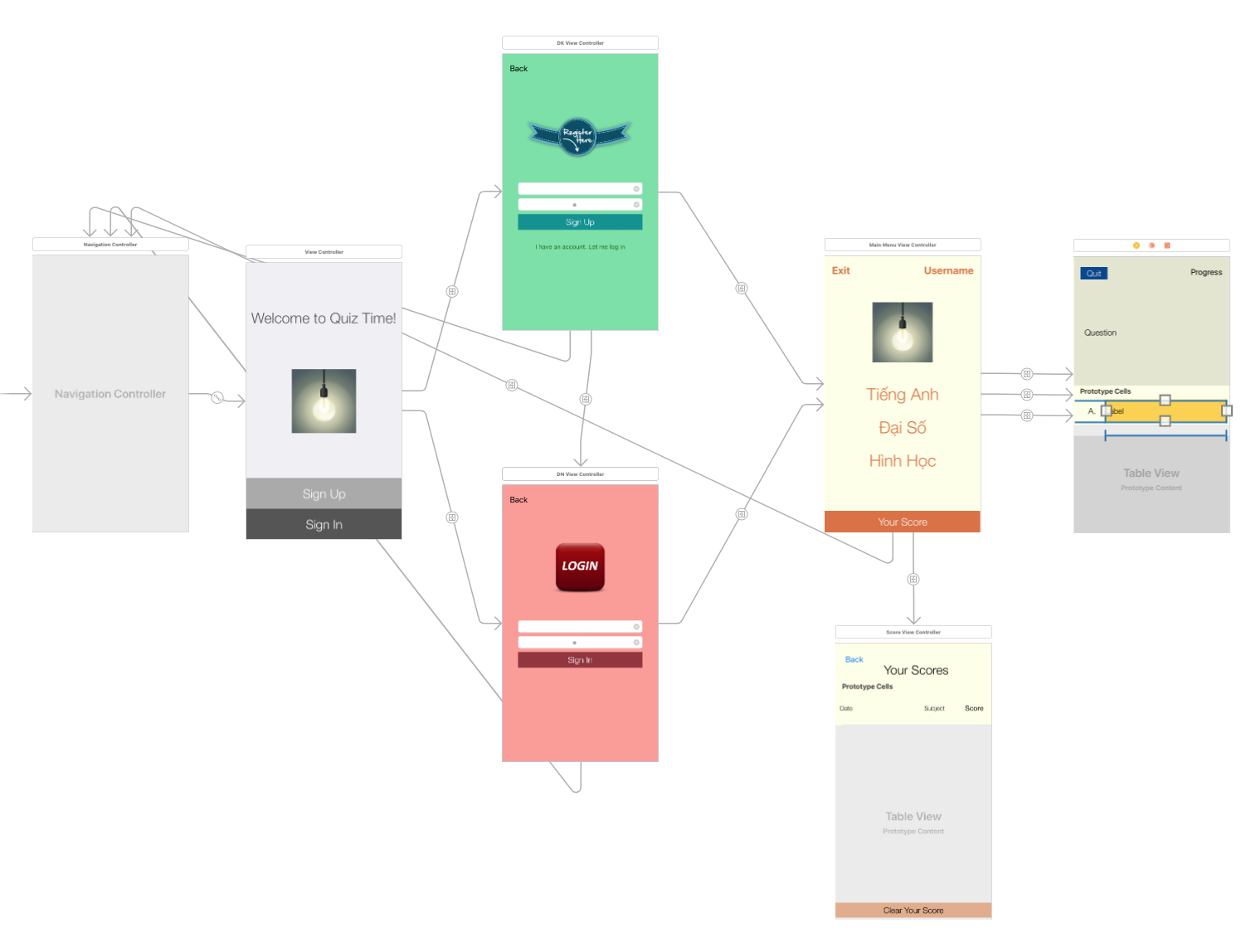


## Phân tích ứng dụng

1. **Sơ đồ Main.storyboard**

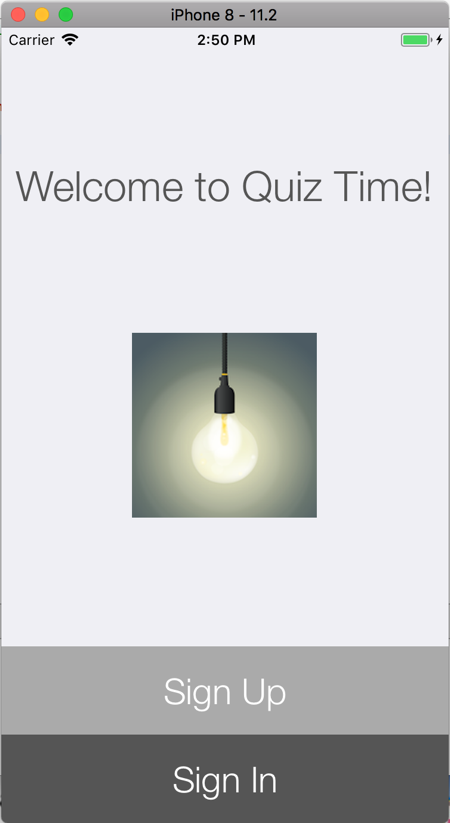
Gồm:

* **View Controller:** màn hình khởi tạo
* **DK View Controller**: màn hình đăng ký
* **DN View Controller**: màn hình đăng nhập
* **Main Menu View Controller**: màn hình menu
* **Score View Controller**: màn hình điểm của người dùng
* **Test View Controller**: màn hình bài kiểm tra

****

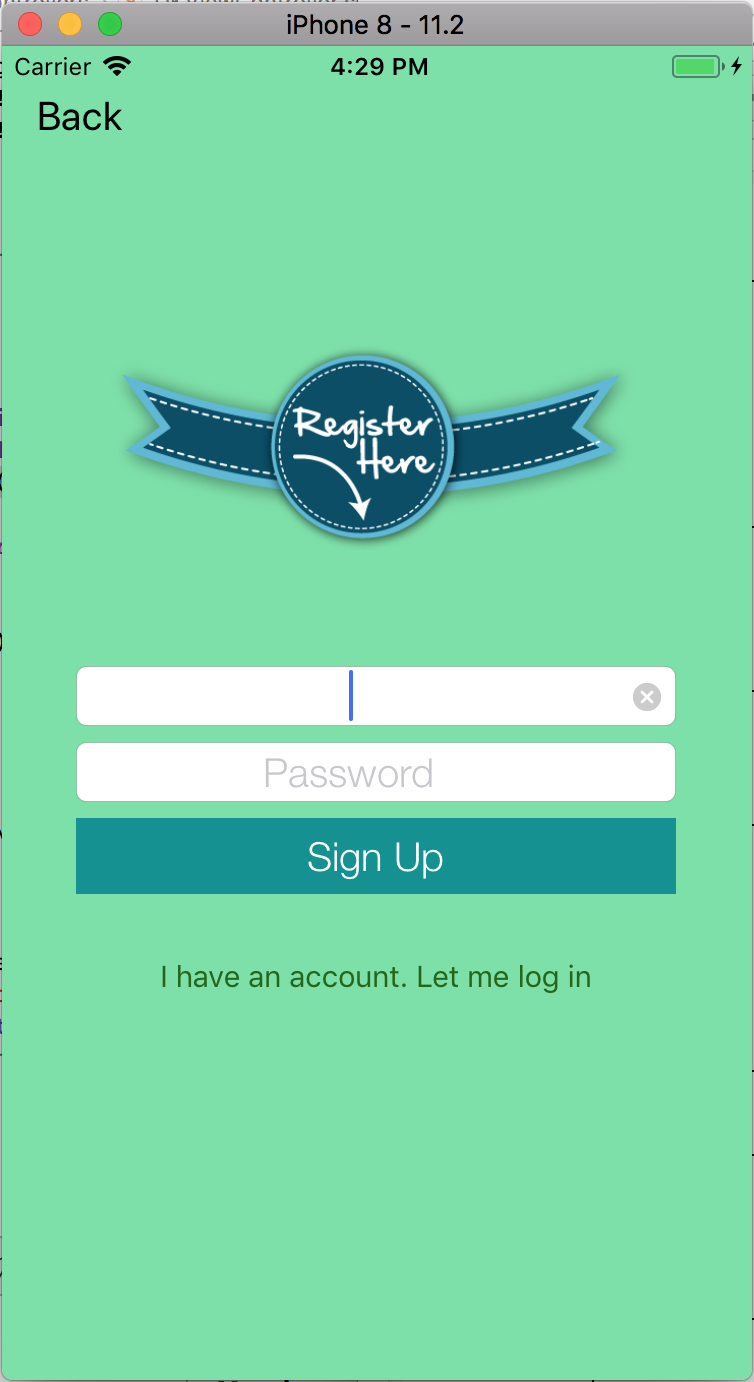
1. **Phân tích các View Controllers**
   1. **View Controller**

|  |  |
| --- | --- |
| **Input** | -Nút **Sign Up**  -Nút **Sign In** |
| **Xử lý** | -Luồng **Sign Up**: đến **DK View Controller**  -Luồng **Sign In**: đến **DN View Controller** |
| **Output** | -Segue từ nút **Sign Up**  -Segue từ nút **Sign In** |



* 1. **DK View Controller (DKViewController.swift)**

|  |  |
| --- | --- |
| **Input** | * **username.text**: tên đăng ký nhập từ username text field * **password.text**: mật khẩu đăng ký từ password text field * Nút **Back** * Nút **Sign Up** * **I have an account. Let me log in** |
| **Xử lý** | -Luồng **Sign Up**: hàm **dkButtonPressed()**  -Luồng **Back**: trở về **View Controller**  -Luồng **I have an account. Let me log in**: chuyển đến **DN View Controller** |
| **Output** | Tên người dùng được gán vào **thisUsername** để chuyển đến **Main Menu View Controller** |

****

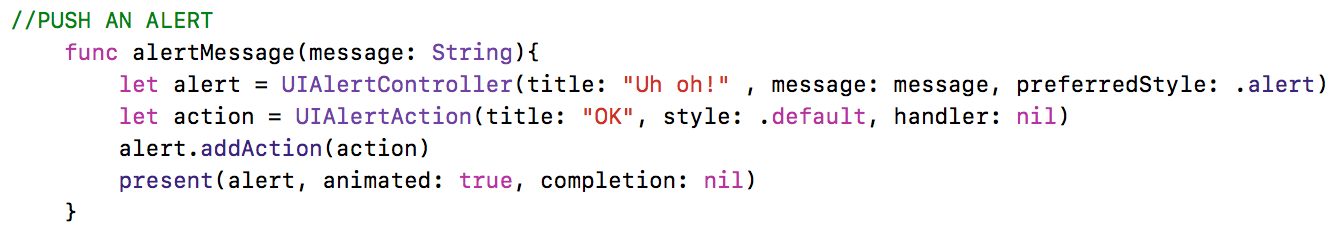
**-Hàm dkButtonPressed()**

Khi người dùng để trống 2 trường hay 1 trong 2 trường bị để trống và bấm **Sign Up** sẽ hiện thông báo yêu cầu người dùng nhập lại. Nếu tên đăng ký đã có, app sẽ hiện thông báo yêu cầu người dùng nhập tên khác.



-**Hàm alertMessage()**

Tham số truyền vào là 1 chuỗi, khi gọi hàm sẽ hiện thông báo kèm với message là chuỗi truyền vào hàm.



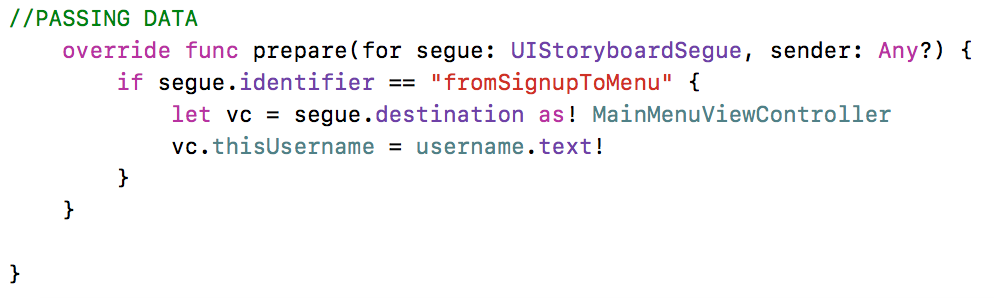
-**Hàm addUser()**

Tham số truyền vào là đối tượng **Examinee**, hàm sẽ thêm **Examinee** được truyền vào cơ sở dữ liệu



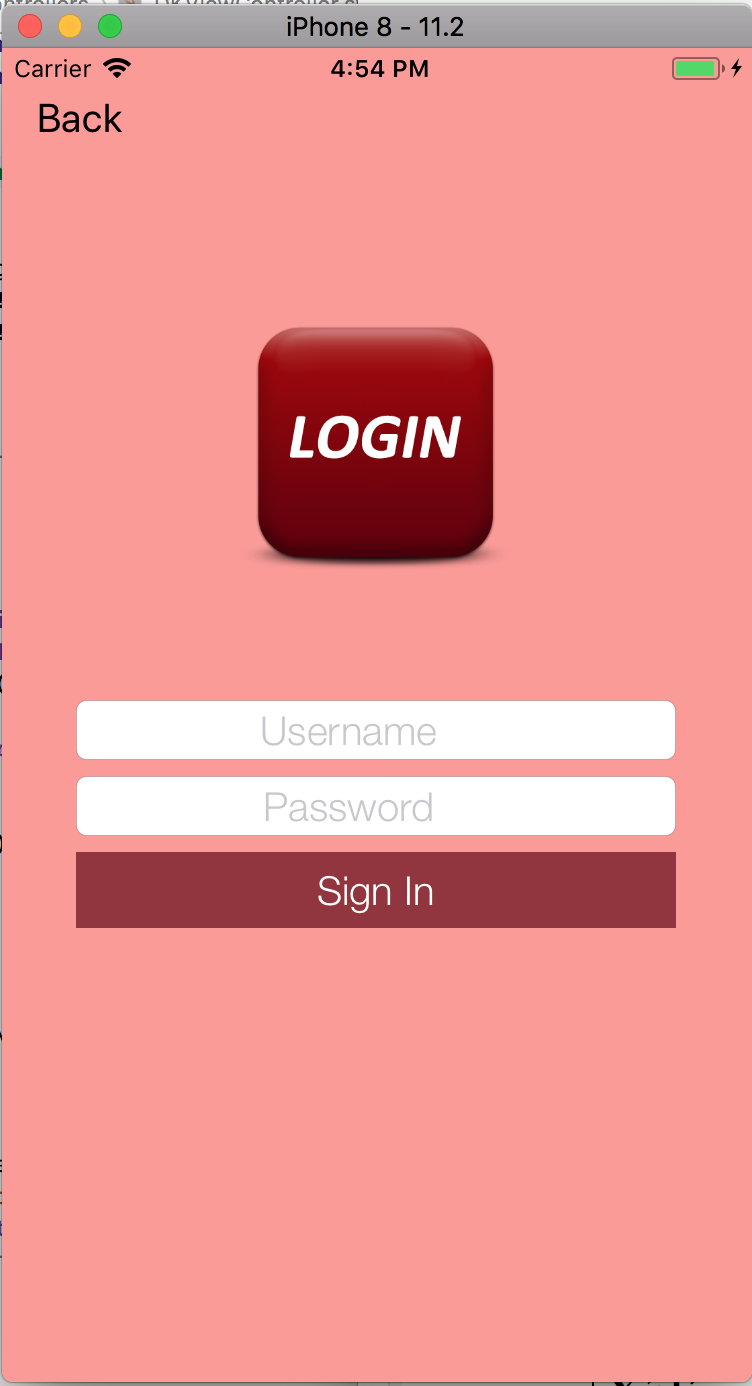
-**Hàm prepare()**

Là hàm có sẵn, sau khi kiểm tra tài khoản đăng ký hợp lệ sẽ chuyển dữ liệu ở trường **username** vào biến **thisUsername**, biến này được khai báo ở **MainMenuViewController.swift**

****

* 1. **DN View Controller (DNViewController.swift)**

|  |  |
| --- | --- |
| **Input** | * **username.text**: tên đăng nhập từ username text field * **password.text**: mật khẩu đăng nhập từ password text field * Nút **Back** * Nút **Sign In** |
| **Xử lý** | -Luồng **Sign In**: hàm **signinButtonPressed()**  -Luồng **Back**: trở về **View Controller** |
| **Output** | Tên người dùng được gán vào **thisUsername** để chuyển đến **Main Menu View Controller** |

****

**-Hàm signinButtonPressed()**

Hàm kiểm tra nếu 2 trường hay 1 trong 2 trường để trống sẽ hiện thông báo yêu cầu người dùng nhập lại. Sau khi nhận được dữ liệu từ 2 text fields, hàm sẽ truy vấn đối tượng **Examinee** có **username** trùng với tên đăng nhập, kiểm tra nếu **password** không khớp sẽ yêu cầu người dùng nhập lại.



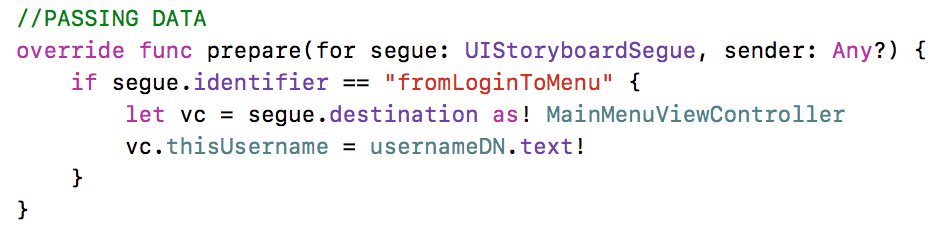
**-Hàm alertMessage()**

Hàm sẽ đưa thông báo cùng với message truyền vào, sau đó xoá 2 trường username và password để người dùng nhập lại khi bấm vào nút **OK** trên thông báo.



**-Hàm prepare()**

Hàm truyền tên đăng nhập vào **thisUsername**, biến này được khai báo ở **MainMenuViewController.swift**

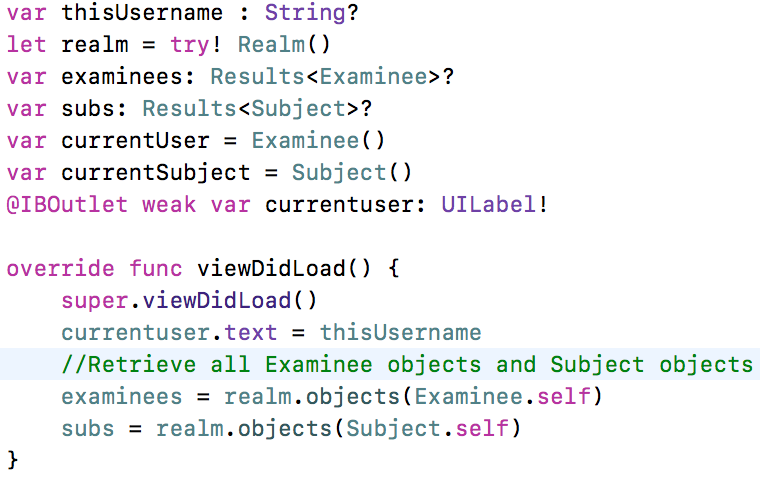


* 1. **Main Menu View Controller (MainMenuViewController.swift)**

|  |  |
| --- | --- |
| **Input** | -Nút **Exit**  -username truyền vào biến **thisUsername**  -Nút **Tiếng Anh**  -Nút **Đại Số**  -Nút **Hình Học**  -Nút **Your Score** |
| **Xử lý** | -Luồng **Exit**: hàm **exitButtonPressed()**  -Luồng **Tiếng Anh**: hàm **TAButtonPressed()**  -Luồng **Đại Số**: hàm **DSButtonPressed()**  -Luồng **Hình Học**: hàm **HHButtonPressed()**  -Luồng **Your Score**: chuyển đến **Score View Controller** |
| **Output** | -Truyền **subID** của môn học đã chọn vào **thisSub** và **username** của người dùng hiện tại vào **thisUser**, 2 biến **thisSub** và **thisUser** được khai báo ở **TestViewController.swift**  -Truyền tên của người dùng hiện tại vào **thisUsername** được khai báo ở **ScoreViewController.swift** |

****

**Khai báo các biến:**

****

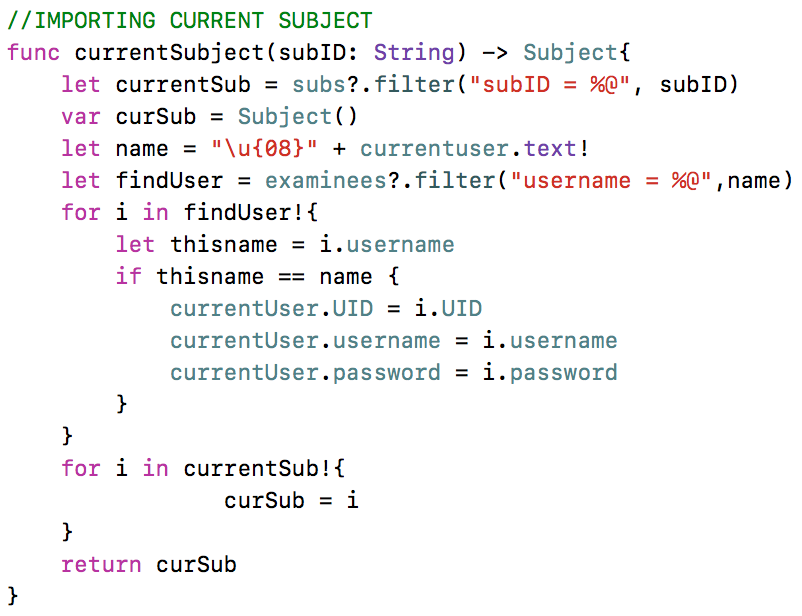
**-Hàm TAButtonPressed(), DSButtonPressed(), HHButtonPressed()**

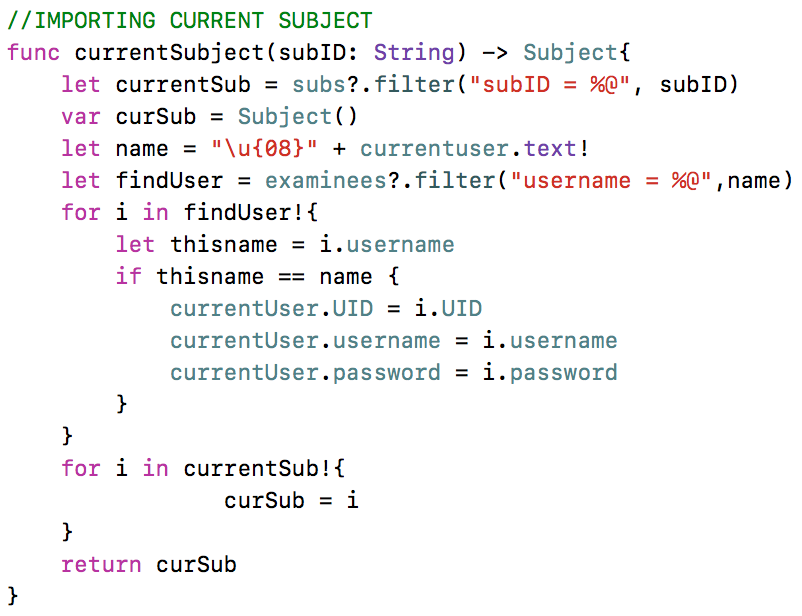
Khi các nút tương ứng với tên các môn học được ấn, môn học hiện tại sẽ đẽ truy vấn thông qua hàm **currentSubject(subID: String)** và truyền vào biến **currentSubject**



Hàm **currentSubject()**

Hàm sẽ truy vấn Subject dựa trên subID truyền vào, người dùng hiện tại dựa vào dữ liệu trong khung currentuser





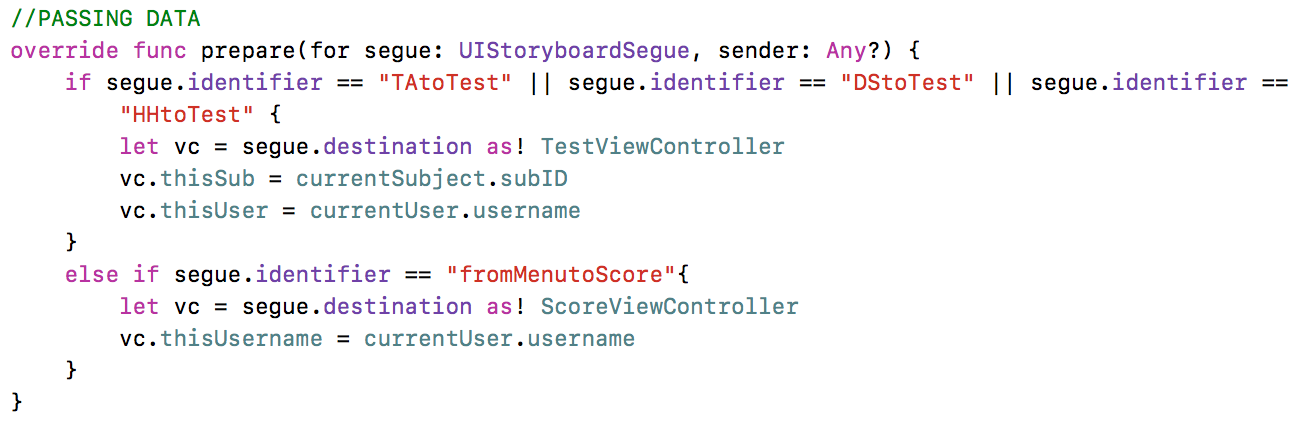
**-Hàm exitButtonPressed()**

Khi ấn **Exit**, thông báo sẽ hỏi người dúng có muốn thoát hay không, nếu có chọn **Yes** và chuyển sang **View Controller**, nếu không chọn **No**



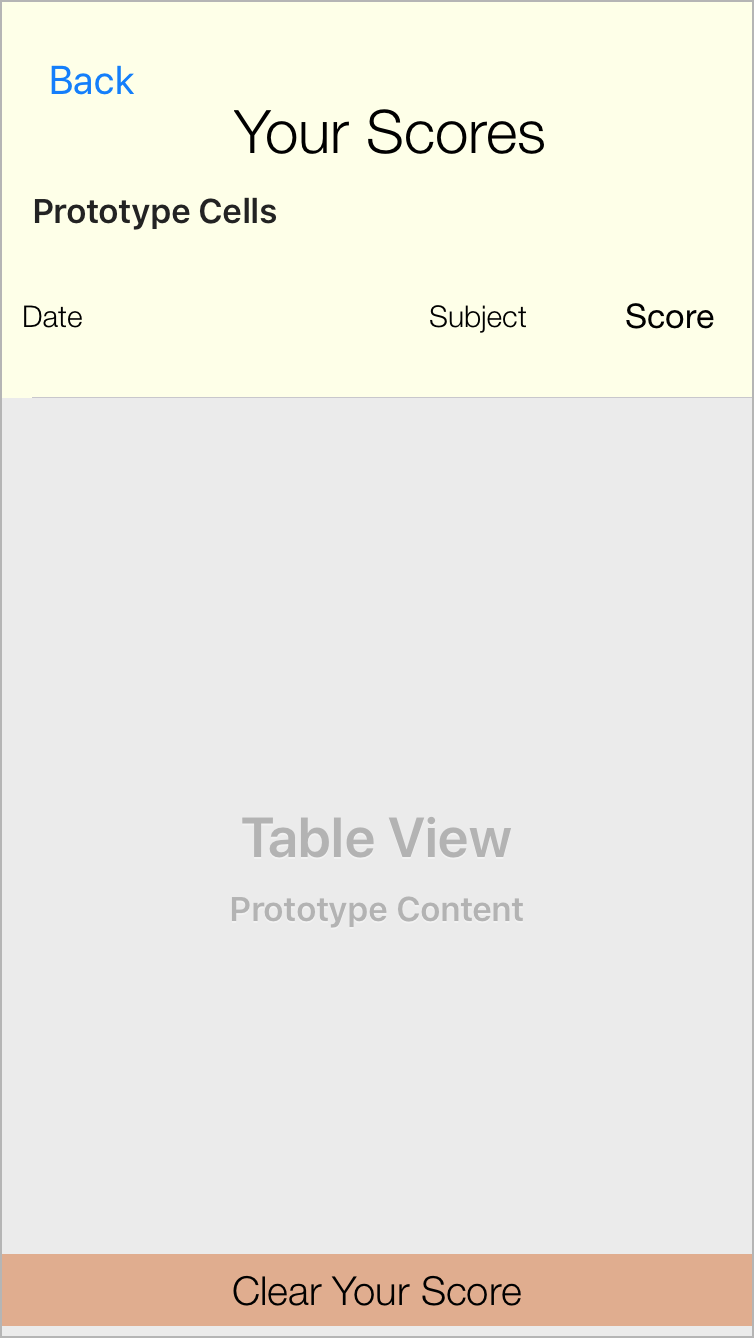
**-Hàm prepare()**

Nếu luồng là các nút chọn môn học thì hàm sẽ truyền **subID** của môn học hiện tại vào **thisSub** và tên người dùng hiện tại vào **thisUser** rồi chuyển qua **Test View Controller**. Nếu ấn nút **Your Score**, hàm sẽ truyền tên người dùng hiện tại vào **thisUsername** rồi chuyển sang **Score View Controller.**

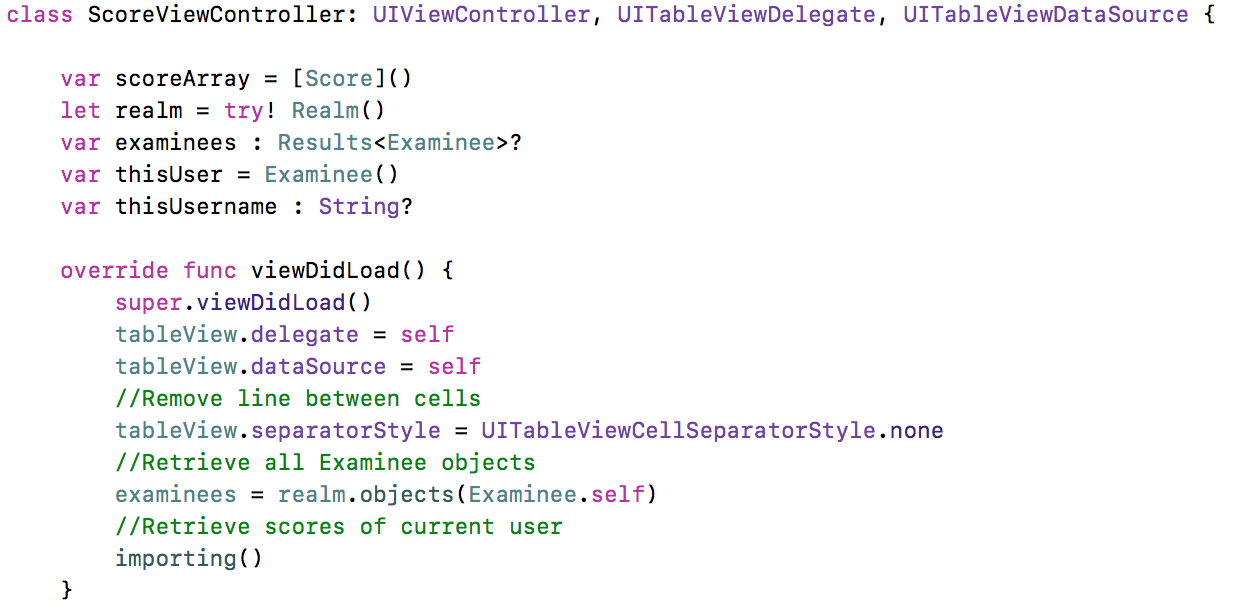


* 1. **Score View Controller (ScoreViewController.swift)**

|  |  |
| --- | --- |
| **Input** | -Nút **Back**  -Tên người dùng hiện tại được truyền vào biến **thisUsername**  -Nút **Clear Your Score** |
| **Xử lý** | -Luồng **Back**: hàm **backButtonPressed()**  -**Tableview** (hiển thị score của người dùng theo từng hàng): các hàm xử lý dữ liệu và hiển thị lên tableview  -Luồng **Clear Your Score**: **clearButtonPressed()** |
| **Output** | -Nếu ấn **Clear Your Score**, toàn bộ điểm lưu trong database của người dùng hiện tại sẽ bị xoá |

****

**Khai báo biến và các lệnh để sử dụng tableview:**

****

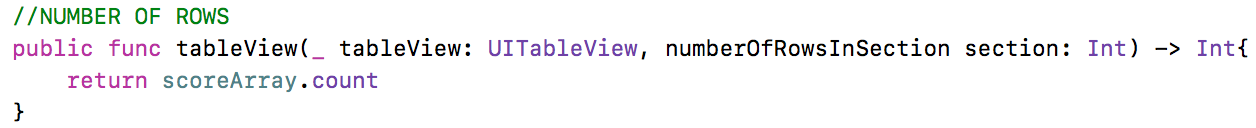
**-Hàm importing()**

Hàm sẽ dùng giá trị được lưu trong biến **thisUsername** để truy vấn người dùng hiện tại trong cơ sở dữ liệu. Điểm của người dùng sẽ được lưu trong mảng **scoreArray**.

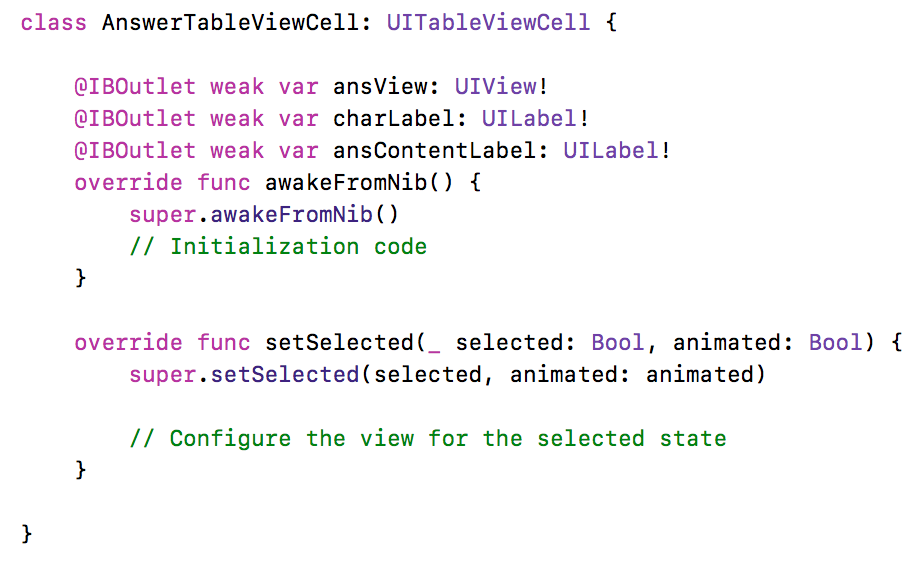


-**Hàm lấy số hàng và dữ liệu cho cell của tableview**

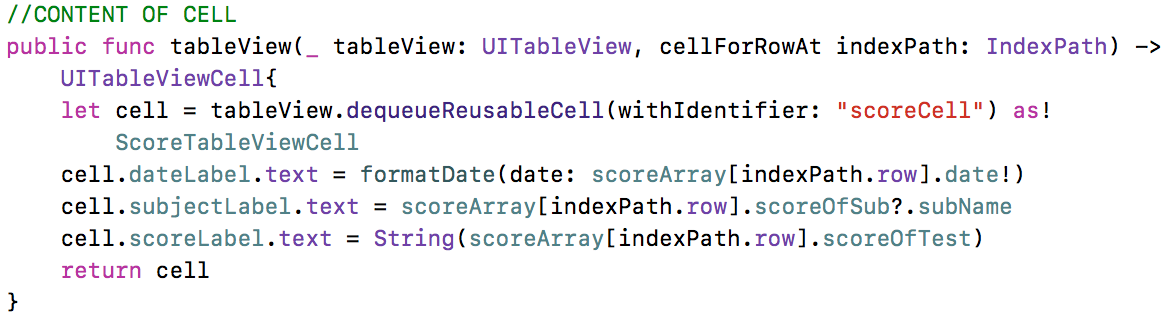
Số hàng sẽ là số phần tử trong mảng **scoreArray**



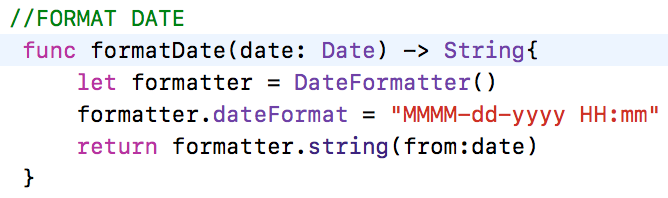
Cell được cấu hình trong **ScoreTableViewCell.swift** gồm 3 text fields: **dateLabel** hiển thị thời gian hoàn thành bài kiểm tra, **subjectLabel** hiển thị tên môn học, **scoreLabel** hiển thị điểm của bài kiểm tra.



Hàm trả về nội dung từng cell của tableview

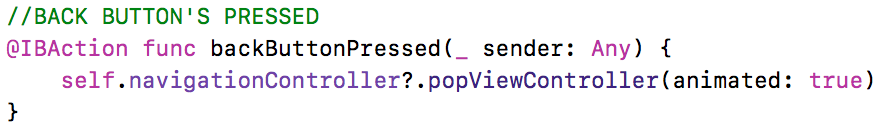


Hàm định dạng ngày tháng cho **dateLabel**



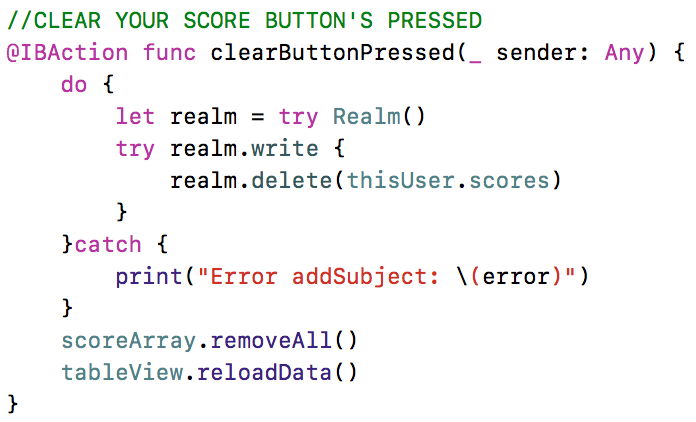
-**Hàm backButtonPressed()**

Chuyển về **Main Menu View Controller**



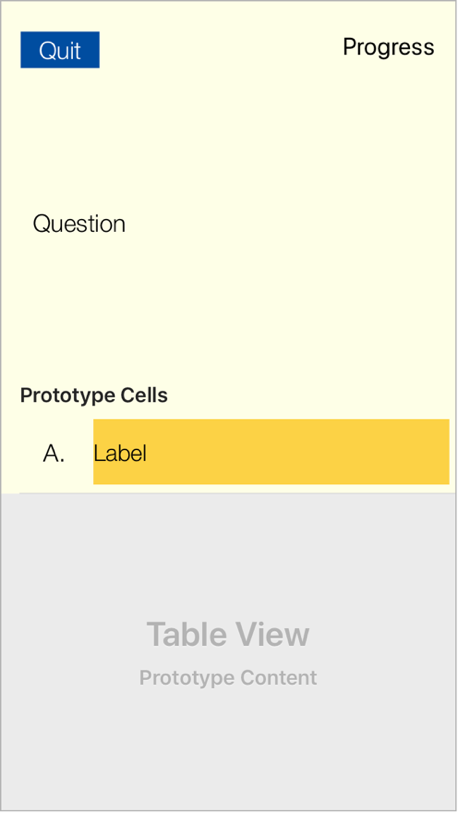
**-Hàm clearButtonPressed()**

Hàm sẽ xoá tất cả điểm lưu trong cơ sở dữ liệu của người dùng hiện tại, xoá tất cả phần tử của mảng **scoreArray** để reload lại tableview.



* 1. **Test View Controller (TestViewController.swift)**

|  |  |
| --- | --- |
| **Input** | -Nút **Quit**  -Tổng số câu hỏi, thứ tự câu hỏi hiện tại  -**subID** của môn học hiện tại được truyền vào **thisSub**  -Tên người dùng hiện tại được truyền vào **thisUser**  -Nội dung câu hỏi, các đáp án của câu hỏi  -Số điểm sau bài kiểm tra |
| **Xử lý** | -Luồng **Quit**: hàm **quitButtonPressed()**  -**Tableview** (hiển thị đáp án): các hàm xử lý cell và dữ liệu sau khi ấn vào cell  - Nội dung câu hỏi, các đáp án của câu hỏi: hàm nextQuestion()  -Số điểm hiện tại và sau bài kiểm tra: hàm checkAnswer() |
| **Output** | Số điểm của người dùng hiện tại ở môn học đã chọn làm kiểm tra. |

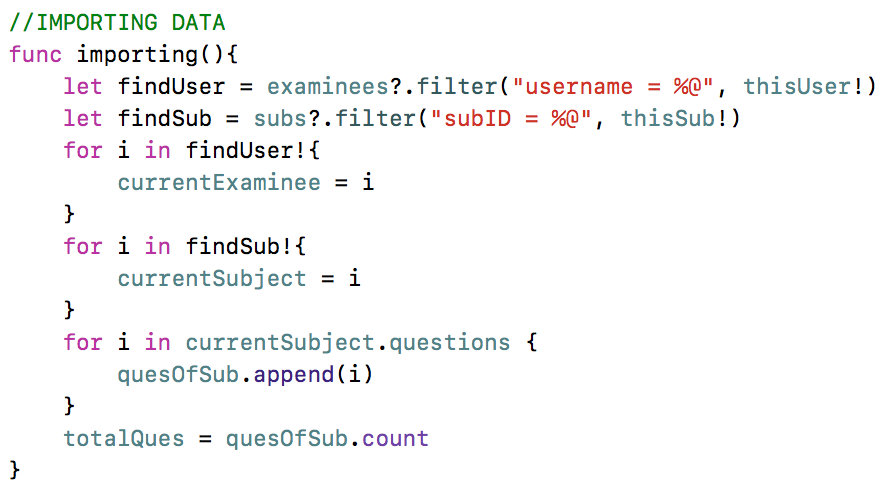
****

**-Khai báo biến, tableview và các label:**



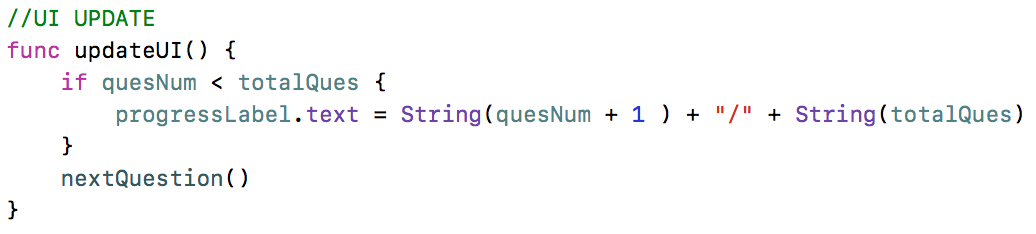
**-Hàm importing()**

Hàm truy vấn người dùng và môn học từ 2 biến **thisSub** và **thisUser**. Sau khi truy vấn môn học, hàm sẽ lưu danh sách câu hỏi vào mảng **quesOfSub** và gán giá trị cho biến **totalQues** là số lượng phần tử của mảng.



**-Hàm updateUI()**

Hàm cập nhật thứ tự câu hỏi hiện tại để hiển thị lên **progressLabel**



**-Hàm nextQuestion()**

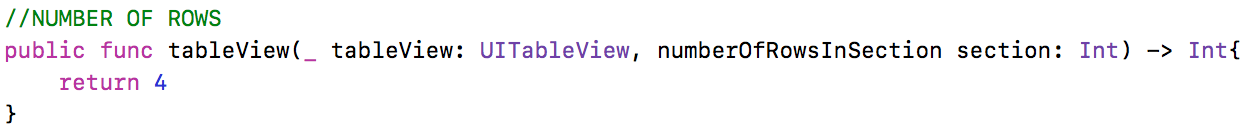
Hàm sẽ kiểm tra nếu chưa hết câu hỏi sẽ hiển thị nội dung câu hỏi kế tiếp lên **quesLabel** rồi cập nhật lại danh sách đáp án của câu hỏi ở mảng **ansArray**.

Nếu câu hỏi hiện tại là câu hỏi cuối cùng, hàm sẽ thêm điểm vào cơ sở dữ liệu và hiện thông báo đã hoàn thành bài kiểm tra.

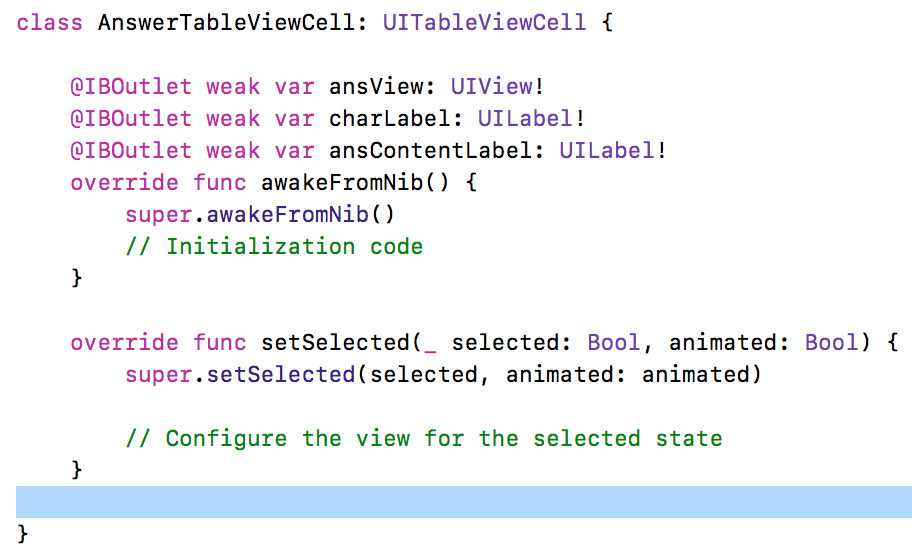


**-Các hàm xử lý cell của tableview:**

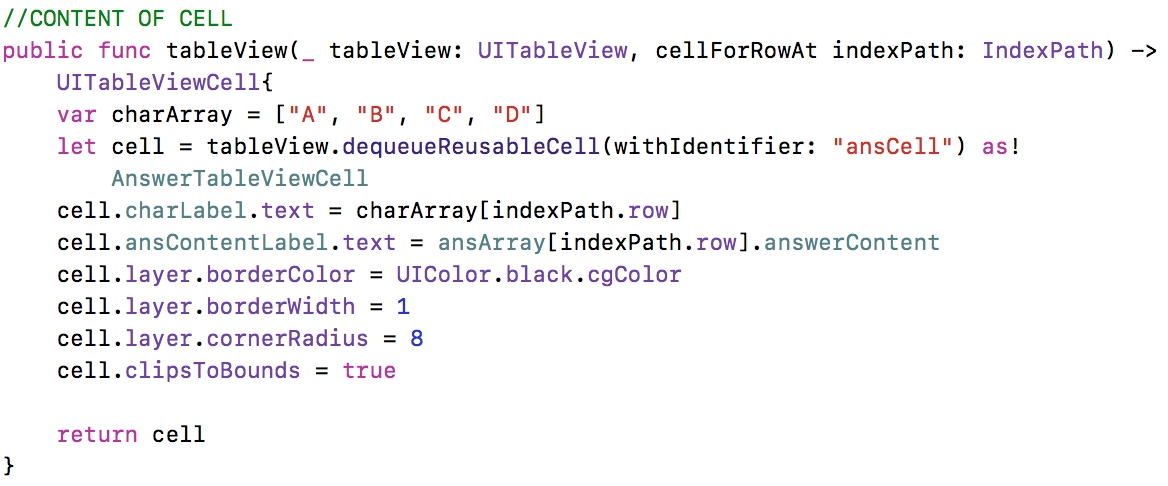
Hàm trả về số cell của tableview



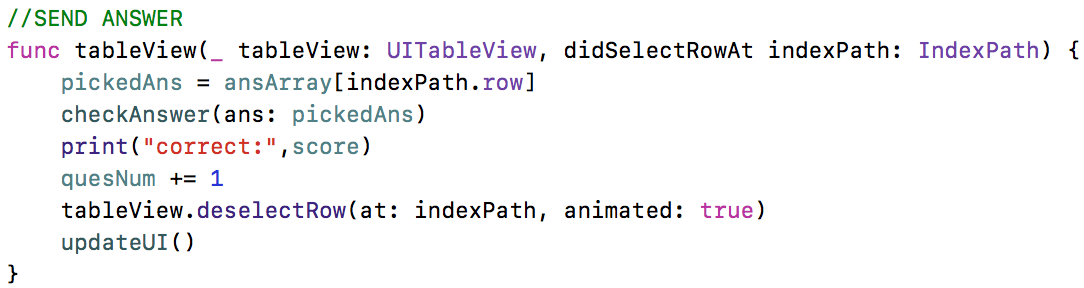
Cell đáp án được cấu hình trong **AnswerTableViewCell.swift** gồm **charLabel** hiển thị kí tự A B C D cho đáp án, **ansContentLabel** hiển thị nội dung đáp án.



Hàm gán dữ liệu và định dạng cho cell

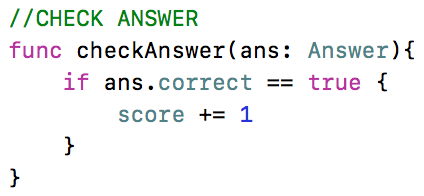


Hàm xử lý khi ấn vào cell



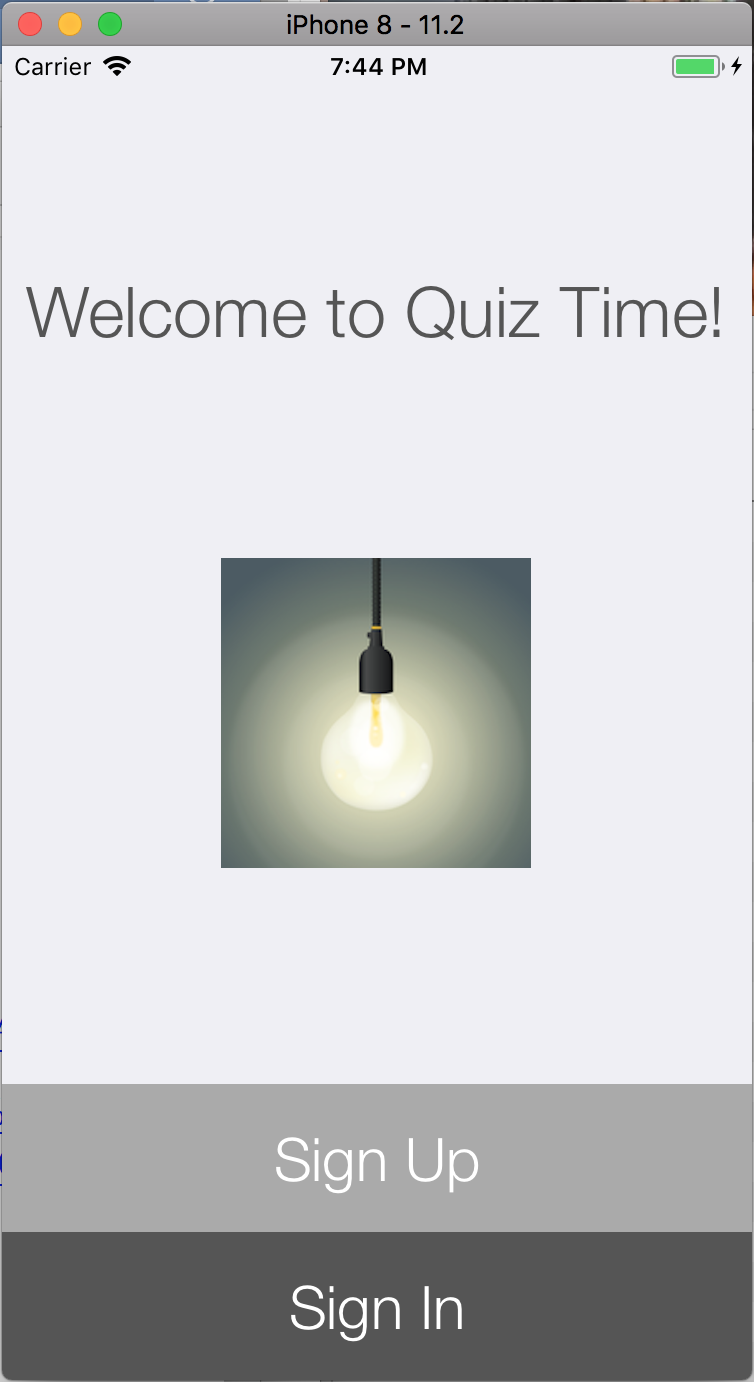
**-Hàm checkAnswer()**

Hàm sẽ kiểm tra dựa vào thuộc tính **correct** của đối tượng **Answer** truyền vào, nếu đúng thì biến **score** tăng lên 1.

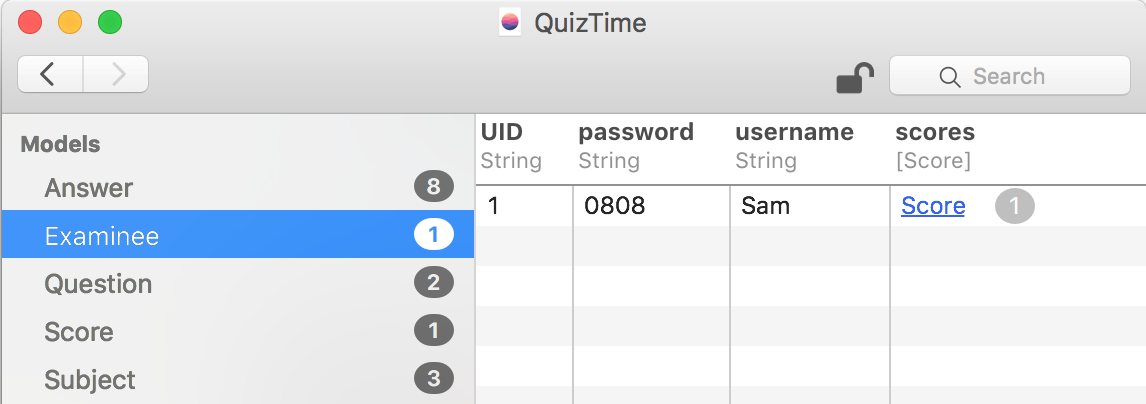


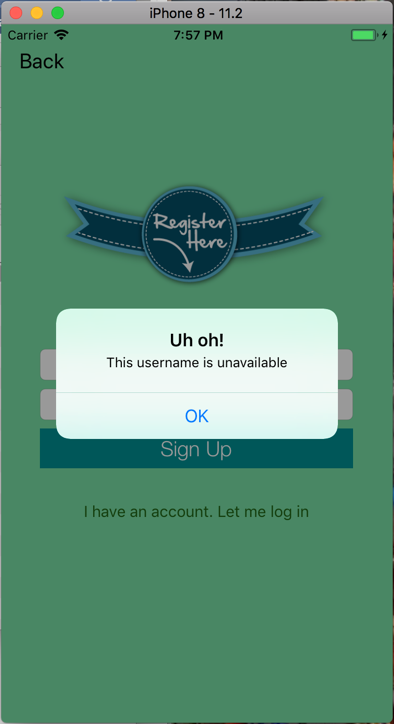
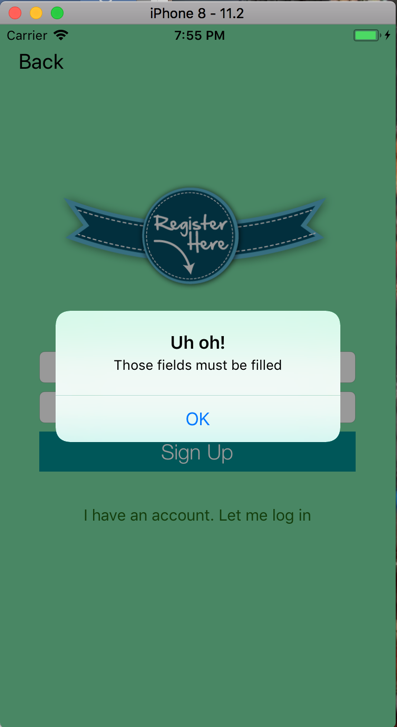
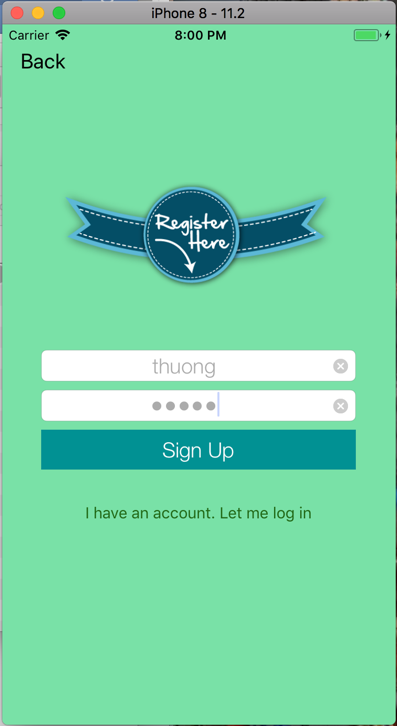
**III.Kết quả thực hiện và kiểm thử**

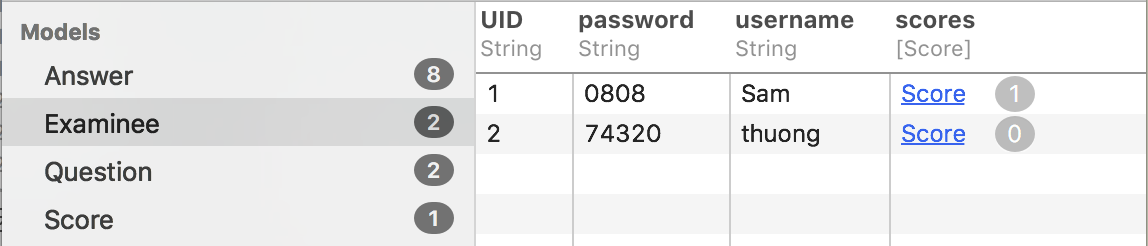
Màn hình khởi tạo của ứng dụng



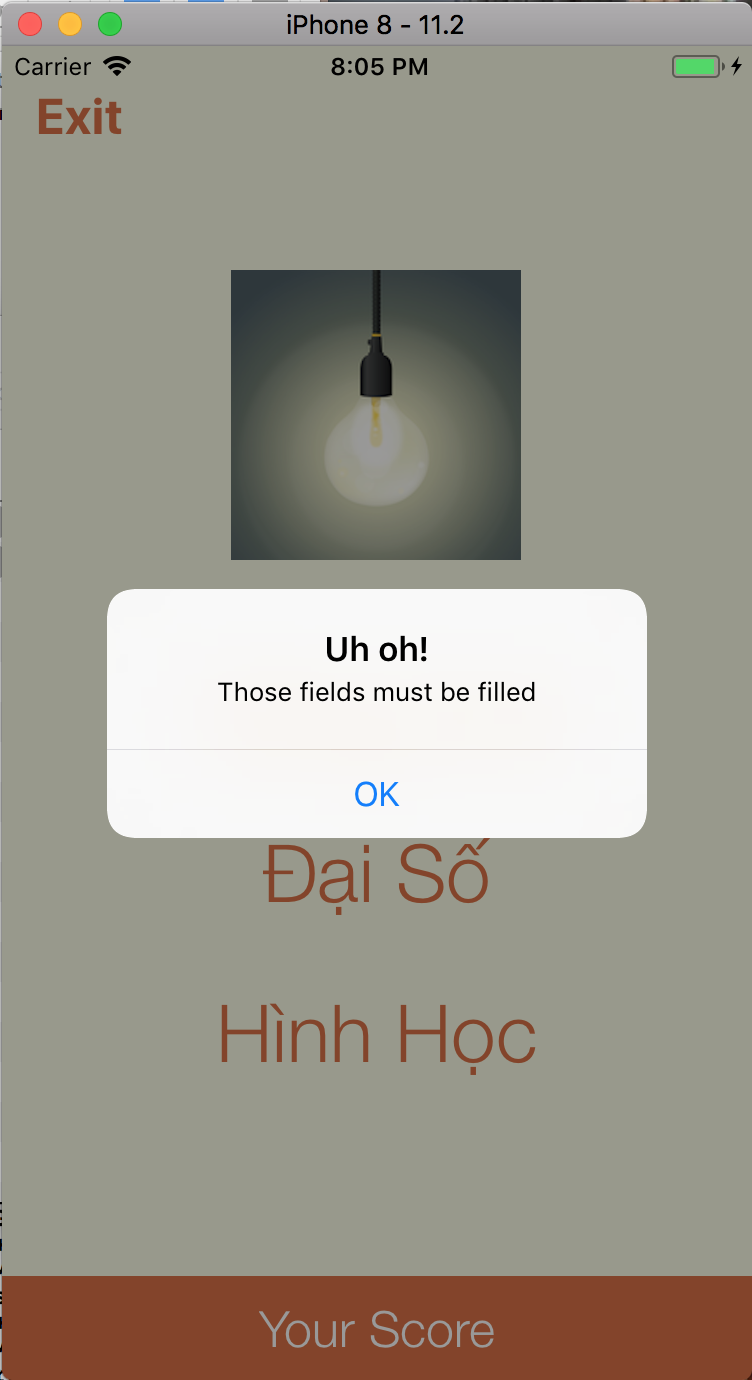
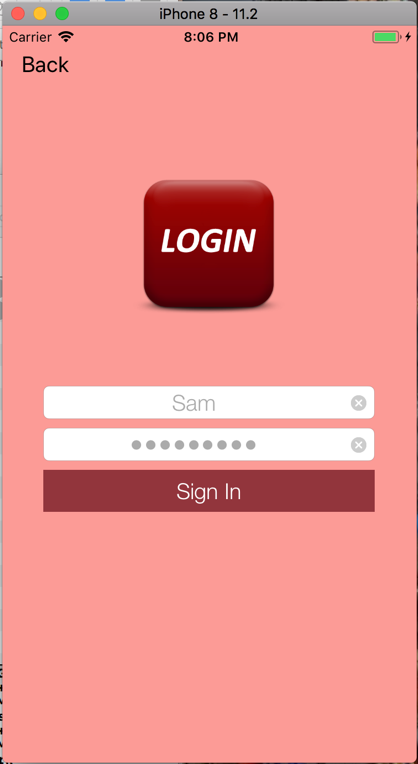
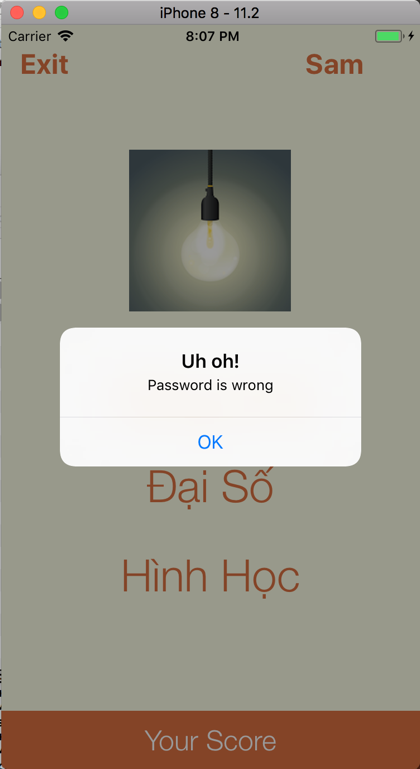
Ấn vào **Sign Up** rồi thử các trường hợp: để trống các trường, nhập tên trùng với tên đã có sẵn trong cơ sở dữ liệu với tên có sẵn là Sam, đăng ký thành công.

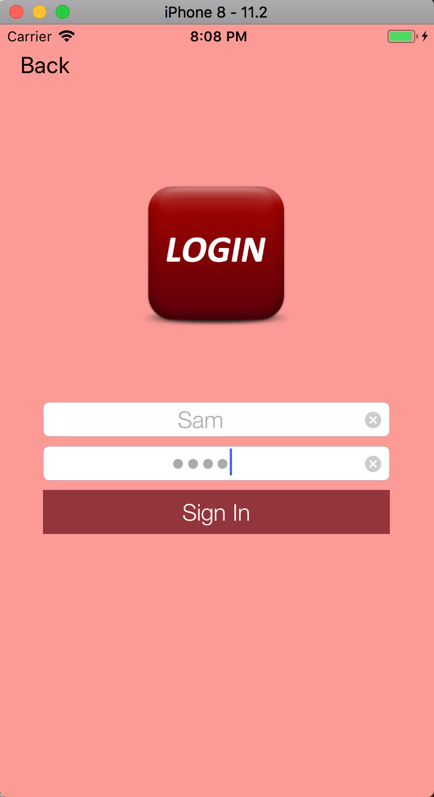




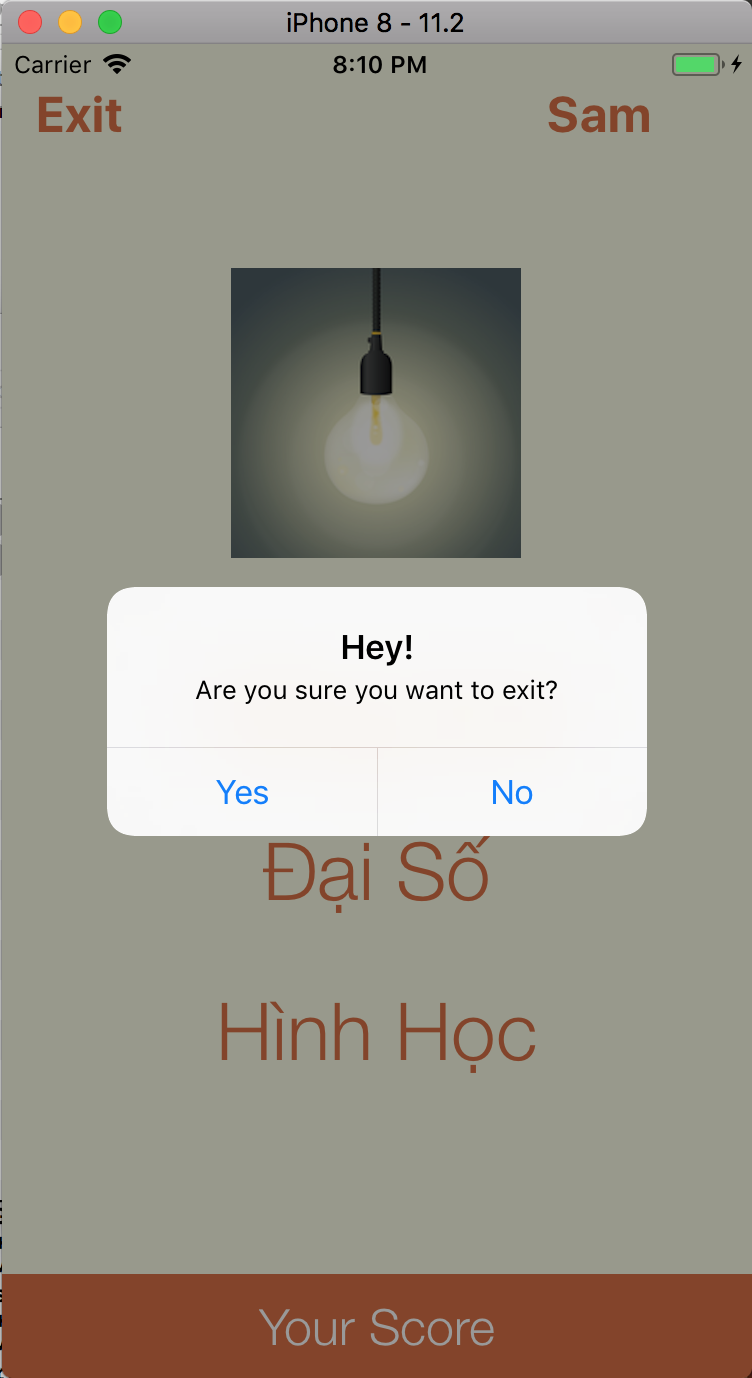


Trở lại màn hình khởi tạo, ấn vào **Sign In** và thử các trường hợp: để trống các trường, nhập sai mật khẩu, đăng nhập thành công.

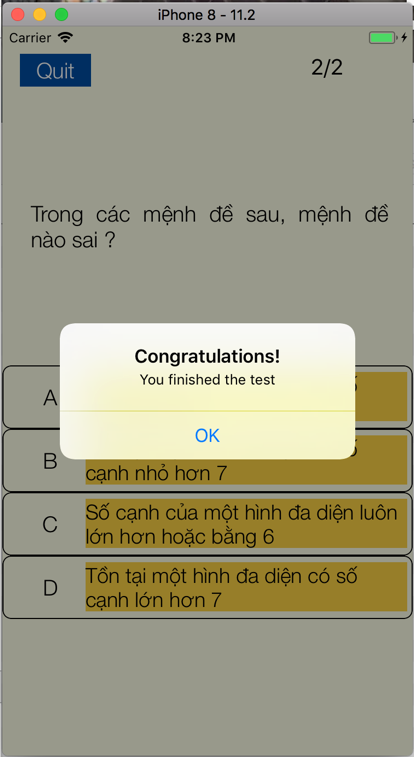


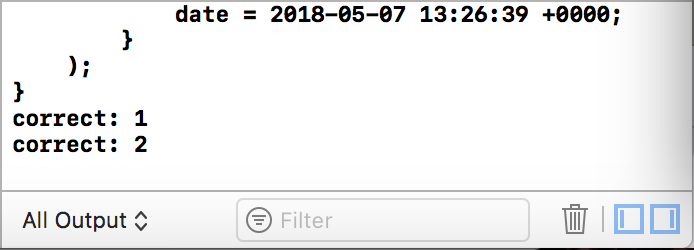


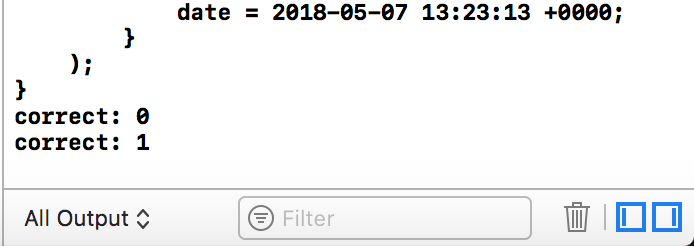
Tại **Main Menu View Controller**, ấn vào **Exit**

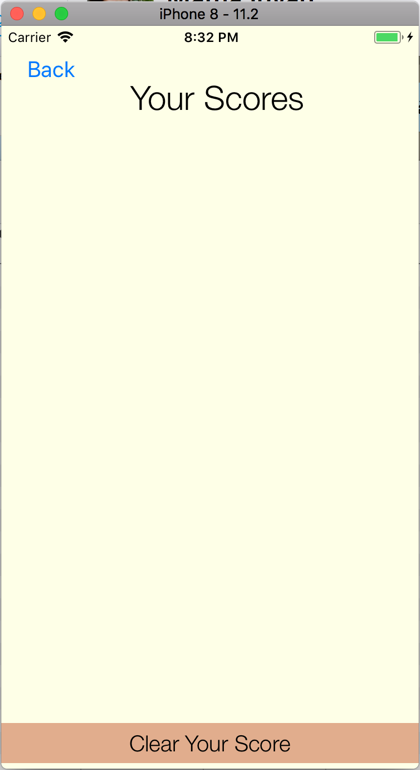


Ấn vào **Hình Học**, trong cơ sở dữ liệu Hình Học có 2 câu hỏi. Câu thứ nhất đáp án đúng là C, câu thứ 2 đáp án đúng là A. Câu thứ nhất chọn đáp án sai và câu thứ 2 chọn đáp án đúng. Sau đó thử lại lần nữa chọn 2 câu đều đúng. Vào **Your Score** để xem kết quả. Tiếp theo ấn **Clear Your Score**.









1. **Kết luận**
2. **Nhận xét về ứng dụng**

* Realm tuy dễ dùng và tiện theo dõi nhưng thêm dữ liệu bằng code khá mất thời gian và bất tiện.
* Thuộc tính UID của đối tượng Examinee bị thừa vì không dùng để truy vấn cũng không được dùng để dữ liệu không bị lặp. Realm rất dễ bị lỗi khi thêm hoặc bớt thuộc tính, phải xoá tập tin .realm và tạo lại cơ sở dữ liệu khác, nghĩa là phải xây dựng lại toàn bộ dữ liệu, rất mất công sức và thời gian.
* DN View Controller bị lỗi nên khi hiện thông báo, alert lại hiển thị trên View Controller tiếp theo (Main Menu View Controller), sau khi ấn vào OK mới trở về.

1. **Hướng cải thiện và phát triển cho ứng dụng**

* Tài khoản đăng ký sẽ được cấp ID, sau đó sẽ đăng nhập bằng ID được cấp.
* Tạo người dùng admin, sau đó xây dựng các View Controller cho phép admin thêm bớt dữ liệu trong cơ sở dữ liệu để cải thiện nhược điểm của Realm.
* Tìm hiểu cách để đưa dữ liệu dạng ảnh, video vào cơ sở dữ liệu để các câu hỏi phong phú hơn.
* Bên cạnh việc lưu điểm, ứng dụng sẽ lưu bài làm của người dùng.

# TÀI LIỆU THAM KHẢO

<https://www.udemy.com/ios-11-app-development-bootcamp/learn/v4/overview>

<https://stackoverflow.com/>

<https://vi.wikipedia.org/>

<https://realm.io/>

<http://vietjack.com/>

<https://www.youtube.com/channel/UCK3l4hRgG-5GmgaN4NC-XFg>

https://www.youtube.com/user/sergeykargopolov