Project

*CECS 590 - Report*

CECS 590 – Mobile Programming - Spring 2015

Viralkumar R. Intwala

7/22/2015

Project

CECS 590 - Report

* **Description :**

**Project Name**: Cocktail

**Application Target**: IPhone

**Project Summary**:

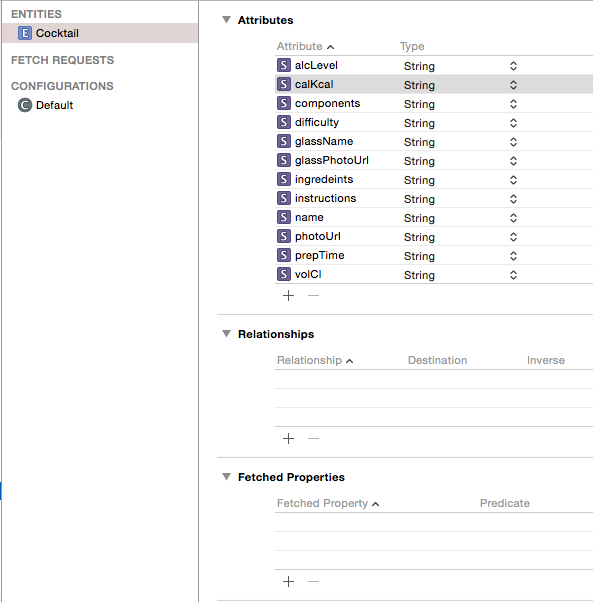
The Application targeted for an IPhone device is a recipe application for cocktails. It gives information about cocktails with their ingredients, nutrition facts and a fun quiz on cocktails.

**Project Functions**:

1. Number of Cocktails with their ingredients, recipe, nutrition facts, a sample photo.
2. A local notification that repeats every Friday at night to get user’s attention on the application.
3. A Random Shaker tab which gives random cocktail on shaking of the device.
4. A fun quiz on cocktails.

* **Logic Employed :**

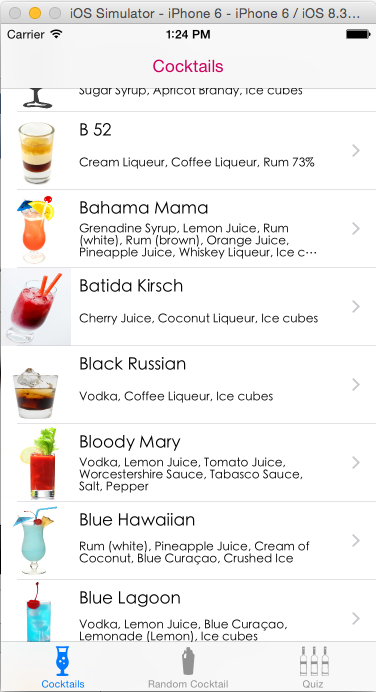
1. **Model Structure**:
2. The first thing needed for this project is the model structure that holds the cocktail information. I.e. The Entity Relationship for the application.
3. Using the Core Data framework this structure is developed.
4. Cocktail is the main and only entity for this application.
5. To form a objective-C class that represents the model, a subclass of the Core Data is created and the model is attached to that class which creates all the properties in the model as NSString objects in the header file.
6. Further, the implementation file also creates dynamic declaration of those attributes.
7. The model has 12 Attributes of String type.
8. Below is a screenshot of the model viewed as a structure:



1. **Local Notification Setup**:
   1. In the AppDelegate implementation setting up a notification as soon as the app starts.
   2. An instance of calendar is taken with the Georgian calendar format and a instance of the current time is taken.
   3. Using that instance, a date object is created such that it holds Friday and time of 20:00 hours.
   4. A local notification instance is initialized and its firedate is set to the date created in the above step.
   5. Further its alert body is set to a String and its set to repeat every week.
2. **Persistence Store Manager**:
   1. An objective-c class is created for the persistence store the application needs to manage and access the data from the database created locally by the application.
   2. It has a class method to initialize the data for all the cocktails from the store and an instance method to get all the cocktails from the store.
   3. The class method calls a private initializer to retrieve all the cocktails from the database and store it in a private NSArray of cocktails. The private initializer creates a context using NSPersistentStoreCoordinator which will be used to create requests for the database.
   4. After the context being created, the class method makes a fetch request to the database to retrieve all the cocktails from the database.
   5. When ran first time, since the database is empty all the cocktail details are added to the database.
   6. The instance method gives a copy of that array to whoever needs it.
3. **Custom Cell view**:
   1. For the table view of this application, a custom cell is need which displays the cocktail name, ingredients and a thumbnail image.
   2. Thus two labels and an image view is added such that the thumbnail is displayed on the left, name label on right top with big fonts and the ingredients label on right bottom with small fonts.
   3. Below is the screenshot of the cell:
   4. 
   5. A custom height is given to cell so it holds the thumbnail properly.
   6. From this NIB file for cell, an objective-c class is created which is subclassed from UITableVIewCell. It has three outlets to hold the three entities of the cell.
4. **Main Views**:
   1. For the main table that displays the cocktails no particular custom view is needed.
   2. For the Detail View a NIB file is created same as the cell.
   3. The detail view has a big main image of a cocktail, Ingredients label and its contents label, Instructions label and its contents, two columns with 3-3 => 6-6 labels to hold the nutrition facts of a cocktail.
   4. For the Random Cocktail view only an image covering the whole view is created which informs user to shake the bottle image.
   5. For the last view which is the Quiz view, two labels to hold the questions for the quiz are created.
   6. And three buttons that hold the answer alternatives are added below the question label.
   7. *Note:* All these views leave space at the top for the navigation controller.
5. **Controllers**:
   1. The main controller for the application which is also the default one is the table view controller for the list of cocktails.
   2. It conforms to UITableViewDelegate
   3. In the init method, tabbar image, tabbar name, navigation title are defined for this view.
   4. In the viewDidLoad method, the Custom cell NIB is registered so that the table view controller can use the custom one instead of the plain cell.
   5. In the viewWillAppear method, we set the scrollview and set the custom height of the cell.
   6. In the cellForRowAtIndex method, all the cocktails are retrieved using store and the entities for the custom cell are setup.
   7. Lastly, the didSelectRowAtIndex method, detail view controller’s cocktail is set to the selected one and the detail view controller is pushed to the navigation stack.
   8. Now, the Detail View Controller is the file owner of the NIB file for its respective view.
   9. Thus, it needs to setup all the images and labels of a cocktail that is selected.
   10. In the application the height of the labels is not fixed, so to make it dynamic all the labels are given height and frame relatively again by using the bounds of the main window.
   11. A maximumHeight is calculated first to constraint the height and further all the expected heights for labels are calculated from the label text and font.
   12. Doing so, always keeps the height maintained for each cocktail data.
   13. And lastly, the labels and images are set to the selected cocktail.
   14. For the Random View Controller, it just needs to conform to the ShakerController.
   15. In the method motionEnded, a shake motion is checked if done or not. If yes then a random cocktail is generated using arc4random and the detail view controller is pushed to the navigation stack with that random cocktail as property.
   16. Lastly for the Quiz view controller, in the viewWillAppear a random cocktail is taken and as the question ingredients are displayed to the view.
   17. Further in the same method the solution is stored and displayed at a random position from the three buttons. Also two random cocktail and their names are taken and displayed at the remaining two positions of the buttons.
   18. Once the answer button is tapped, it is checked with the solution we stored if it matches, success alert is displayed and if not fail alert is displayed.
   19. For the alert, UIAlertView is used. It is initialized and set up with a string message of success or fail accordingly.
6. **Application Setup**:
   1. For the main view of the application AppDelegate needs to be setup such that we have a tab bar view.
   2. There is a need to set up three tabs for the application, thus three instances of 1. CocktailViewController, 2. RandomCocktailViewController and 3. QuizViewController are created.
   3. Now it is necessary for all those controllers to be setup with navigation controller so that traversal can be made to detail controllers.
   4. Three Navigation controllers are setup with root controllers as the three instances created earlier related to each other.
   5. Lastly a tab bar controller is setup with an array containing the three navigation controllers created in above step. And the root controller is added as the cocktailtableviewcontroller.
7. **Localized Strings**:
   1. Every String required for the cocktail data is setup with a string key and it contains the main content as their respective String values.
   2. This allows the data to be separated from the database keys. So ultimately the database holds the key strings to main content strings that belongs to each cocktail.
   3. Also this allows the programmer to add language support to the application in future expansion.

* **Screenshots :**

**Default View of the Application: Three Tabs and Cocktails Showcase**

****

**Selecting a Cocktail to View Recipe:**

****

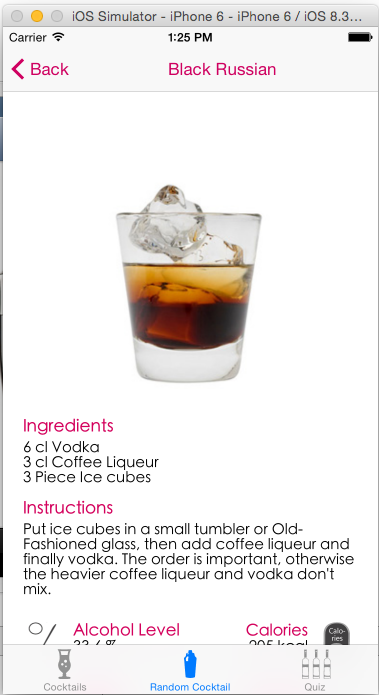
**Detail View Contd.**

****

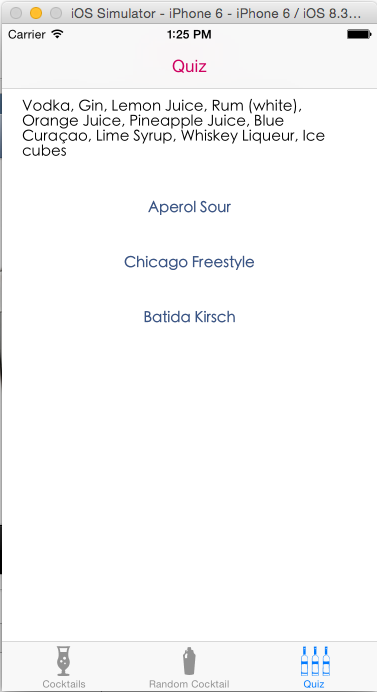
**Random Cocktail Tab: (Shake Motion Gesture)**

****

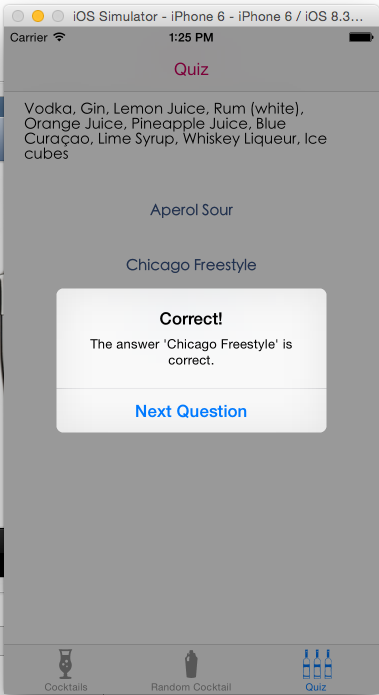
**Shaking the device will show random cocktail recipe:**

****

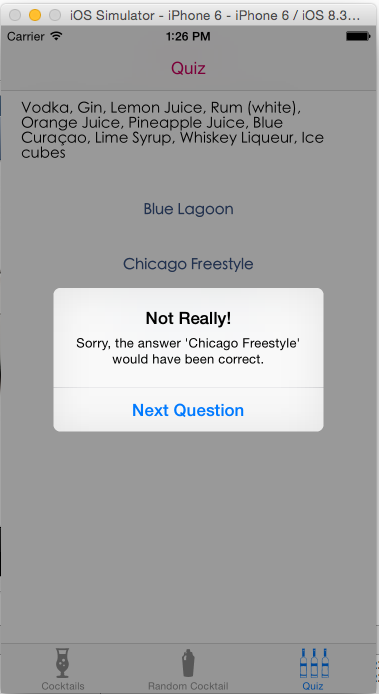
**Quiz Tab:**

****

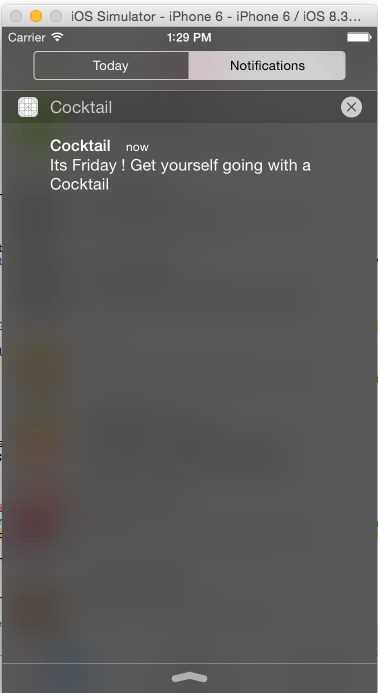
**Tapping answer: (Correct)**

****

**Tapping answer: (Incorrect)**

****

**Notification on Friday Night:**

****

* **Conclusions :**

1. All the basic Functions needed and as stated in the introduction are completed and the application is working fine on IPhone device.
2. There is a potential need in the future for the application to be compatible with IPad Devices too.
3. A programmer can easily add support for multiple languages with localization.
4. Can easily add more Cocktails to the database.