

APPLICATION NOTE ECE 480

Creating a User Interface for the iPhone mobile application in Xcode

By: Emmanuel Wadieh

Date: 11/13/15

Abstract: iPhone application with the use of Xcode is becoming more common because it provides the user with a more friendly way of interacting with the computer. There are numerous applications available on today's market for different purposes. Most iPhone applications do not give the user complex code to implement from but a user-friendly application consisting of entry text, labels and buttons from which a user can implement the desired application. This application note for the ECE 480 design team 1 class, talks about how to create an iPhone application in Xcode using swift.

Introduction

iPhone application developers usually use Xcode and program in swift when creating an application for iPhones. Swift is a programming language, used by apple developers mainly to develop applications in apple products. Unlike objective C, which is also one of the options available to developers when creating an application in Xcode, Swift is very user friendly and drops legacy conversation which is the use of semi-colons to end lines or parenthesis to surround conditional expressions. This makes programming in swift easier compared to Objective C and also makes the code easier to read by another user. The purpose of the User Interface is to create an application that would allow users to be able to interact with a server and database, which would in turn provide the user with information about the user's baggage whereabouts during a flight.

Implementation Details -Setting Up The iPhone Application In Xcode Using Swift

First and foremost, in order to create an iPhone application using the Swift programming language; the user needs to download the Xcode package. This can be done by going to the Xcode download website. Here is a link to that.

<https://developer.apple.com/xcode/download/>.

This website makes available the latest version of Xcode which is on the market. In this case Xcode 7.2 Beta 3. The user must click the link to the download Xcode on this webpage and install this program on the computer. The installation is very simple. For mac user once the software has been download, drag it into the application folder and follow installation instruction all the way through to finish. For windows user run the “exe” file and also follow the installation instruction all the way through to finish.

After the installation is complete, the user can now create a new project and start with the application creation process.

Every application starts with a new project in Xcode, which is Apples Integrated Development Environment (IDE). To create a new project, double-click on the Xcode icon to open up the program. When the program is opened the user needs to click on Create A New Xcode Project. In the IOS application window the user needs to select Single View Application, which is one of Xcode’s built in template and clicks on next. After this the user must name the project and has a choice to select a language, which is either Objective C or Swift. For this senior design project Swift is selected. This window gives the user a chance to select which device the application should be on. In the case of this project that option was set to universal in order to allow this application to be used by

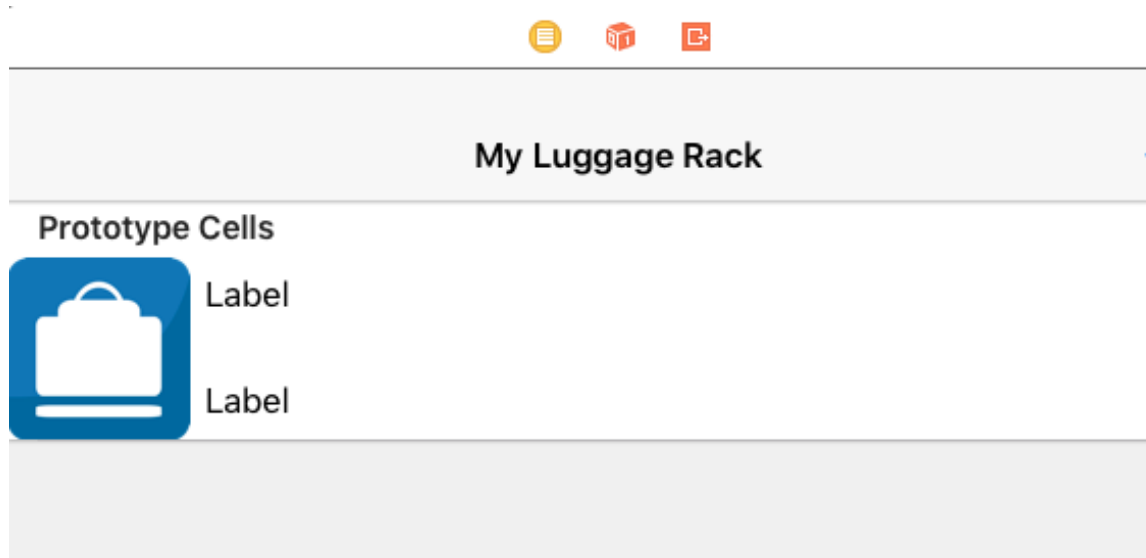
all apple products. Once this is done click next, save the project on a desired location on the computer and click on create.

Implementation Details - Creating The iPhone Application

Xcode opens the project to the project setting views. The files that create the application are listed in the project navigator on the left side of the window. To begin designing the application the user must select the main.storyboard file. This is a way to visually edit the application user interface. After user interface element has been added to the storyboard a code is linked to them that provides functionality. The object library contains all the user interface elements that can be added to the storyboard. This application consists of several features such as pictures and button, entry text and label widgets. When using a picture make sure it has the “png” file extension. Using other picture file extension might not work and hence not recommended. This application contains five main features.

These features are navigator controller scene, my luggage rack scene, navigation controller scene, travel scene and lastly a service table view controller. First and Foremost the user needs to add a navigation controller to the storyboard. A navigation controller scene manages navigation through a hierarchy of views in IOS programming. This is done by dragging the navigation controller, which is located amongst a list of useful functions on the bottom right part of the screen. The first navigation controller scene is the storyboard entry point. This is the start of the hierarchy like structure. Also when dragged unto the screen the first navigation controller is linked to my luggage rack. In the luggage rack scene, a lot of new features are added such as labels and pictures. In order to add a picture, download the desired picture and save it into your “assets.xcassets

folder”. After this is done, drag the Image View widget onto the screen. This widget display a single image, or an animation described by an array of images. The next step is to drag the saved or desired picture into Image View widget. This helps create a picture on the screen. Below is an example



The second step is to add labels. Labels are added by simply dragging a label text widget onto the mainstoryboard. The user has a chance to rename them. This can be done in a couple of ways. The easiest way is to double click on the label delete its initial content and fill the space in with what the user desires. Also an add button feature is added to the top write of this view controller in order to allow the user to add more luggage. This is done, by also simply dragging the button widget onto the screen. The user has an option to change the name and this is also done the same way, as the user would change the name for the label. Also the view controller can be renamed. This is effective so that the user knows which views are available and can also distinct between each view. This is

done by clicking the top panel of the controller and filling in the name field to the right of the screen. The next step is to create another navigation controller that would link and control the travel information. The travel information also has one more added feature, which is the entry text widget. The entry text widget allows the user to enter text in the required field. Also two more buttons are added to this section of the code, which is the search, and cancel buttons. Below is an example of the main storyboard.

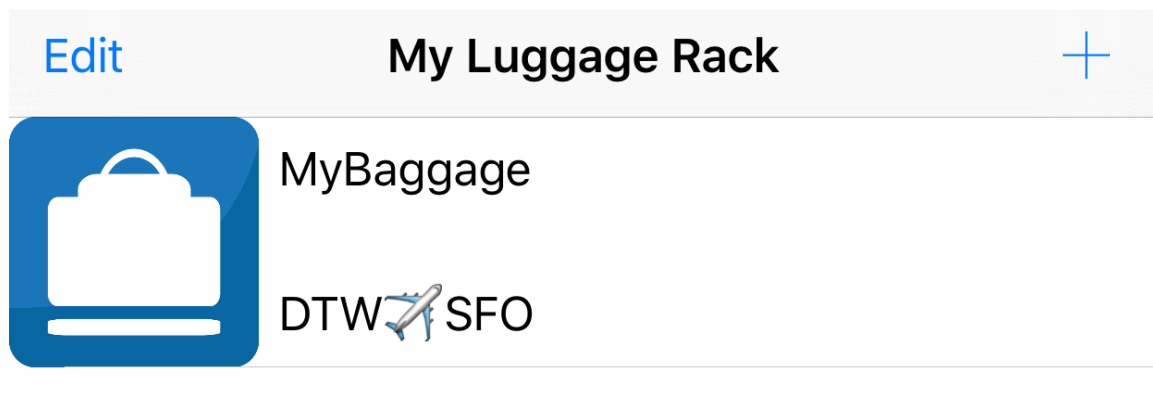


Cancel	Travel Information	Search
Passenger Name		
Enter the name of baggage owner		
Flight No.		

The entry text widget is also added in a similar fashion as labels, button and view or navigation controllers, which is to drag them onto the desired part of the screen. Entry fields are made for Passenger Name and Flight No. In this senior design project. Also in a similar fashion the top header of the view controller is renamed to Travel information the same way my luggage rack was renamed. Which is to drag the navigation bar title onto the view controller click on it and to the right of the screen, the user can fill out the desired name. The final step is to add a table view prototype content. This is done by also dragging onto the screen.

Results

In completion, the mobile application's main functionality is to allow users to enter their name and flight number in the required text field. This would then go through the database to find the luggage and check to see if it has arrived or not arrived at its desired destination. The user also has the feature to add more luggages this allows the user to search for multiple luggages. The user can also delete luggage if the list is also too much or if the luggage has already been found and has no further use in the system. This is an example of how the mobile application looks when it is run.



When the "+" button is clicked, the user is sent to this page

Cancel

Travel Infomation

Search

Passenger Name

Enter the name of baggage owner

Flight No.

This page allows the user to enter details such as flight information and flight number.