iOS: Exam

Introduction

When you are done, send an archive (right click -> compress) containing all projects (one project per folder) to emmanuel.perez@3ie.fr with the subject "[iOS] submission $LOGIN".

Your file need to be named "`ios2013_ing1_$LOGIN.zip`".

Clean your projects before sending them.

Your file must be submitted before 12h30. (You'll lose points if your submission is late)

All the exercises are to be done with the following parameters:

- Storyboard
- ARC
- no autolayout

Part1: Welcome

The goal is create an application that displays a greeting message to the user.

Create your "Welcome" app by using the "Single View Application" template.

Use a TextField to let the user input its name, and then when the user clicks on a "Greetings" button, display a "Welcome $USER" message in a Label.

The keyboard must disappear when the user is done typing its name.

Part 2: WineCellar

The goal is to create an app to manage a collection of wine bottles.

Create a "WineCellar" application by using the "Tabbed Application" template.

Part 2.1: Project foundations

First step: WineBottle

Create a "WineBottle" class with the following fields

- name [string] (for example Saint-Emilion)
- region [string] (for example Bordeaux)
- year [int] (for example 1982)
- country [for example France]

Create a "- `initWithName:AndRegion:AndYear`" method to initialize your object.

Create a "- `getFullDescription`" method that returns a string containing all the fields from your object. Use the following format :

$NAME - $REGION - $YEAR - $COUNTRY

Create a "`getShortDescription`" method that returns a string containing the following fields from your WineBottle. Use the following format :

$NAME - $YEAR

Second step: Randomize

Create a "+ `getRandomWineForCountry:`" method that returns a randomized bottle (except for the country).

Third step: Console

In your appDelegate.m, in the "`application:didFinishLaunchingWithOptions:`" method, create an array of 20 bottles for the "France" and display them with "NSLog" using the "`getFullDescription`" method.

Part 2.2: Storyboarding

First step: TabBar

Use the firstViewController from the app template to display a "welcome, enjoy this app" message to the user.

Remove the SecondViewController from the tabBar.

Second step: List

Add a UITableViewController named "BottlesTVC" and use it to display a list of 15 bottles for the Country "France" in the second tabBar item (call it "Cellar").

Don't bother trying to use the WineBottles from your appDelegate, create new ones in your tableViewController in the "`viewDidLoad`" method.

Challenge: use TableViewCell of "Subtitle" type.

Third step: Modification

Let the user delete and move rows.

Part 2.3: Sections

The "French" bottles were displayed in the first section of your tableView.

Use the second section to display a list of 10 random "Italian" wines.

Give a title to each section by using the "header" from the tableView (tip: go take a look at the UITableViewDelegate Protocol Reference).

Challenge: do not let the user move a row from one section to another.

Part 2.4: DetailView

When a user clicks on a row, push a view to display the value of the fields from the selected row.

Name your new class "WineBottleDetailVC".

Part 2.5: Edition

Let the user edit the values in the WineBottleVC (except for the country that we don't want the user to modify because it's linked to the section).

When you go back to the list, the values displayed must be up to date.