实验编号： 11 **四川师大《IOS》实验报告 2018** 年 **11** 月 **21** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 多MVC \_

姓名： 郭周倩 学号： 2016110413 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验 十一 \_\_\_\_**多MVC **\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 掌握多MVC的设计以及实现；
3. 完成苹果官网的FoodTracker全部内容；
4. 实验要求
5. 认真填写实验报告，要求附加部分运行界面和主要代码；
6. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
7. 实验内容
8. 实现一个简单的多mvc程序
   1. 控制器之间正向传参
   2. 控制器之间反向传参
9. 完成苹果官网的FoodTracker Demo（多MVC）
   1. TableView（自定制Cell）；
   2. NavigationController；
   3. Modal Controller;
   4. ImagePickerController;

说明：苹果官网Demo网址如下

<https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/index.html>

1. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
2. 实现一个简单的多mvc程序
   1. 控制器之间正向传参
   2. 控制器之间反向传参

* 程序代码：

一共有三种方法：

这个只写第三种：

第一个界面代码

  override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

        if segue.identifier == "Second" {

            let secVC = segue.destination as! SecondViewController

            secVC.no = tfNo.text!

            secVC.name = tfName.text!

        }

    }

    @IBAction func backRoot(segue:UIStoryboardSegue){

        let secVC = segue.source as! SecondViewController

        tfNo.text = secVC.no

        tfName.text = secVC.name

    }

第二个界面代码

  override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

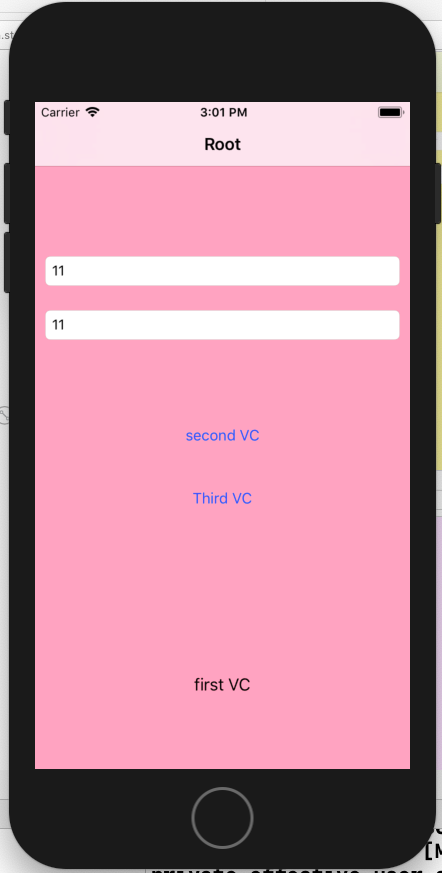
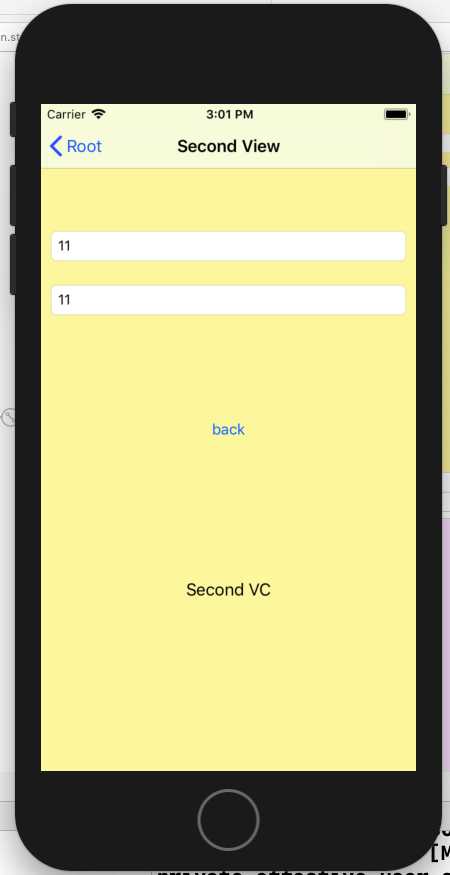
        no = tfNo.text!

        name = tfName.text!

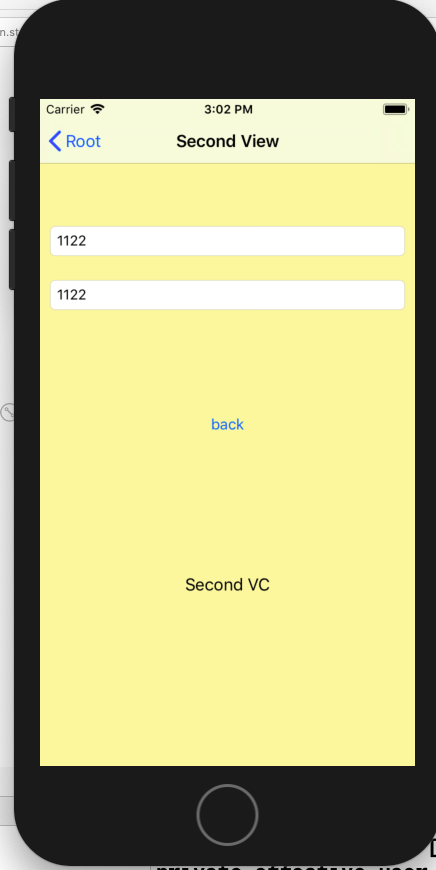
    }

* 运行结果：

正向传值：

反向传值：

1. 完成苹果官网的FoodTracker Demo（多MVC）
   1. TableView（自定制Cell）；
   2. NavigationController；
   3. Modal Controller;
   4. ImagePickerController;

说明：苹果官网Demo网址如下

<https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/index.html>

* 程序代码：

import UIKit

import os.log

class MealViewController: UIViewController, UITextFieldDelegate, UIImagePickerControllerDelegate, UINavigationControllerDelegate {

    @IBOutlet weak var nameTextField: UITextField!

    @IBOutlet weak var photoImageView: UIImageView!

    @IBOutlet weak var ratingControl: RatingControl!

@IBOutlet weak var saveButton: UIBarButtonItem!

var meal: Meal?

    override func viewDidLoad() {

        super.viewDidLoad()

        // Handle the text field’s user input through delegate callbacks.

        nameTextField.delegate = self

        // Set up views if editing an existing Meal.

        if let meal = meal {

            navigationItem.title = meal.name

            nameTextField.text = meal.name

            photoImageView.image = meal.photo

            ratingControl.rating = meal.rating

        }

        updateSaveButtonState()

    }

    func textFieldDidBeginEditing(\_ textField: UITextField) {

        // Disable the Save button while editing.

        saveButton.isEnabled = false

    }

    func textFieldShouldReturn(\_ textField: UITextField) -> Bool {

        // Hide the keyboard.

        textField.resignFirstResponder()

        return true

    }

    func textFieldDidEndEditing(\_ textField: UITextField) {

        updateSaveButtonState()

        navigationItem.title = textField.text

    }

    func imagePickerControllerDidCancel(\_ picker: UIImagePickerController) {

        dismiss(animated: true, completion: nil)

    }

    func imagePickerController(\_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : Any]) {

        guard let selectedImage = info[UIImagePickerControllerOriginalImage] as? UIImage else {

            fatalError("Expected a dictionary containing an image, but was provided the following: \(info)")

        }

        // Set photoImageView to display the selected image.

        photoImageView.image = selectedImage

        dismiss(animated: true, completion: nil)

    }

    @IBAction func cancel(\_ sender: UIBarButtonItem) {

        // Depending on style of presentation (modal or push presentation), this view controller needs to be dismissed in two different ways.

        let isPresentingInAddMealMode = presentingViewController is UINavigationController

        if isPresentingInAddMealMode {

            dismiss(animated: true, completion: nil)

        }

        else if let owningNavigationController = navigationController{

            owningNavigationController.popViewController(animated: true)

        }

        else {

            fatalError("The MealViewController is not inside a navigation controller.")

        }

    }

    // This method lets you configure a view controller before it's presented.

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

        super.prepare(for: segue, sender: sender)

        // Configure the destination view controller only when the save button is pressed.

        guard let button = sender as? UIBarButtonItem, button === saveButton else {

            os\_log("The save button was not pressed, cancelling", log: OSLog.default, type: .debug)

            return

        }

        let name = nameTextField.text ?? ""

        let photo = photoImageView.image

        let rating = ratingControl.rating

        // Set the meal to be passed to MealTableViewController after the unwind segue.

        meal = Meal(name: name, photo: photo, rating: rating)

    }

    //MARK: Actions

    @IBAction func selectImageFromPhotoLibrary(\_ sender: UITapGestureRecognizer) {

        // Hide the keyboard.

        nameTextField.resignFirstResponder()

        // UIImagePickerController is a view controller that lets a user pick media from their photo library.

        let imagePickerController = UIImagePickerController()

        // Only allow photos to be picked, not taken.

        imagePickerController.sourceType = .photoLibrary

        // Make sure ViewController is notified when the user picks an image.

        imagePickerController.delegate = self

        present(imagePickerController, animated: true, completion: nil)

    }

    //MARK: Private Methods

    private func updateSaveButtonState() {

        // Disable the Save button if the text field is empty.

        let text = nameTextField.text ?? ""

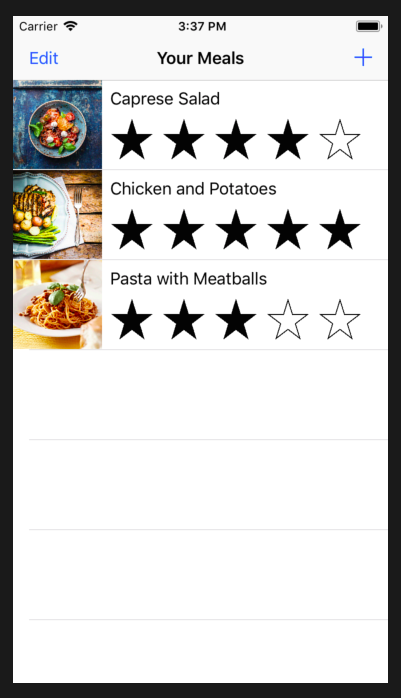
        saveButton.isEnabled = !text.isEmpty

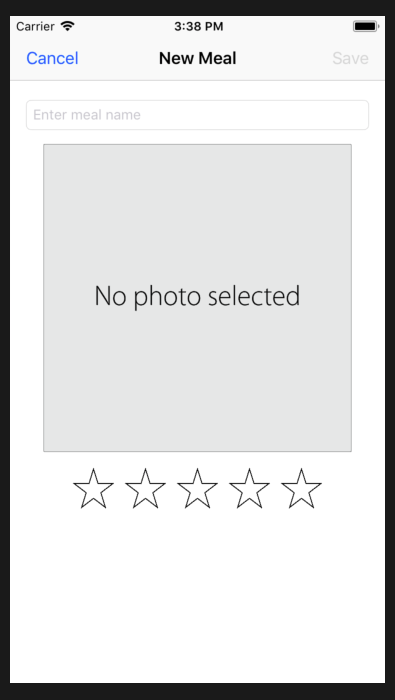
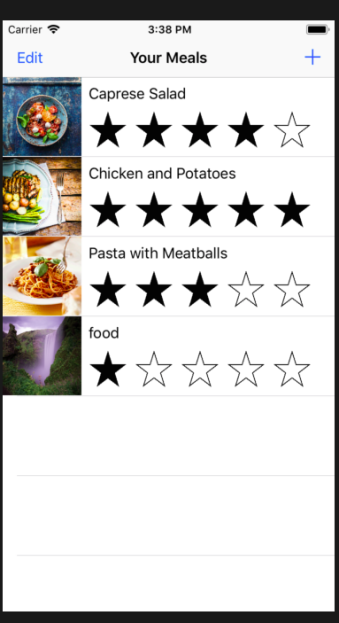
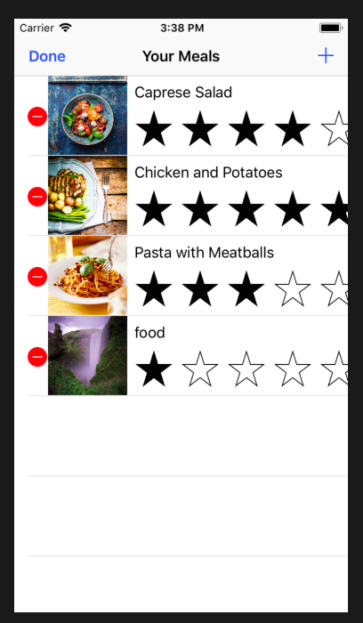
    }

}

这里只粘贴了主要的代码，完整代码在GitHub上下载

* 运行结果：

1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

Github地址：

本次实验主要学要我们学会的多MVC之间的正向反向传值。作业中有两个小作业，第一个是简单的两个页面的传值问题，第二个就是参照官网制作一个多MVC的实例app。

传值有三种方法，但是我还是比较喜欢公共对象传值，segue也还可以，用协议的话略显麻烦。

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。