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：\_樊琳\_ 学号：2016110408\_ 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. 实验内容
5. 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
6. 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：

//

// AppDelegate.swift

// MultiMVC

//

// Created by student on 2018/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

//

// ViewController.swift

// MultiMVC

//

// Created by student on 2018/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

class FirstViewController: UIViewController {

@IBOutlet weak var tfNo: UITextField!

@IBOutlet weak var tfName: UITextField!

let userDefault = UserDefaults.standard

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

}

override func viewWillAppear(\_ animated: Bool) {

tfNo.text = userDefault.string(forKey: "no")

tfName.text = userDefault.string(forKey: "name")

}

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!

//store to file

userDefault.set(tfNo.text!, forKey: "no")

userDefault.set(tfName.text!, forKey: "name")

userDefault.synchronize()

print(NSTemporaryDirectory())

}

}

@IBAction func myback(segue:UIStoryboardSegue) {

if let secVC = segue.source as? SecondViewController {

tfNo.text = secVC.no

tfName.text = secVC.name

}

}

//

// @IBAction func showSecondVC(\_ sender: Any) {

// stu.no = tfNo.text!

// stu.name = tfName.text!

//

// let secVC = UIStoryboard(name: "Main", bundle: nil).instantiateViewController(withIdentifier: "SecondVC") as! SecondViewController

// secVC.stu = stu

//

// self.navigationController?.pushViewController(secVC, animated: true)

// }

//

// @IBAction func showThirdVC(\_ sender: Any) {

// let thirdVC = UIStoryboard(name: "Main", bundle: nil).instantiateViewController(withIdentifier: "ThirdVC")

// present(thirdVC, animated: true, completion: nil)

// }

}

//

// SecondViewController.swift

// MultiMVC

//

// Created by student on 2018/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

class SecondViewController: UIViewController {

var name = ""

var no = ""

@IBOutlet weak var tfNo: UITextField!

@IBOutlet weak var tfName: UITextField!

override func viewDidLoad() {

super.viewDidLoad()

tfNo.text = no

tfName.text = name

// Do any additional setup after loading the view.

}

//

// @IBAction func back(\_ sender: Any) {

// no = tfNo.text!

// name = tfName.text!

//

//

// navigationController?.popViewController(animated: true)

// }

//

// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation

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

no = tfNo.text!

name = tfName.text!

}

}

//

// ThirdViewController.swift

// MultiMVC

//

// Created by student on 2018/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

class ThirdViewController: UIViewController {

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

}

@IBAction func close(\_ sender: Any) {

dismiss(animated: true, completion: nil)

}

/\*

// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation

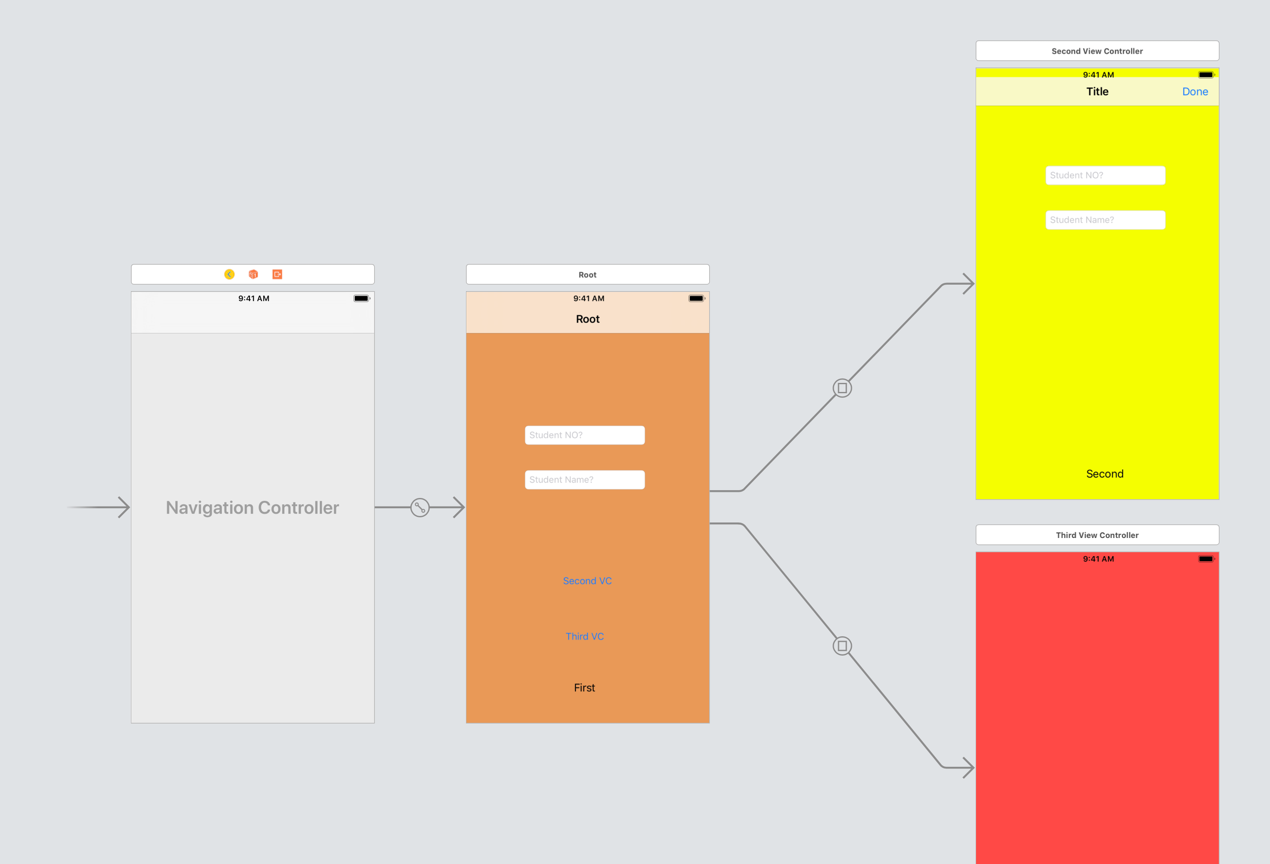
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

// Get the new view controller using segue.destination.

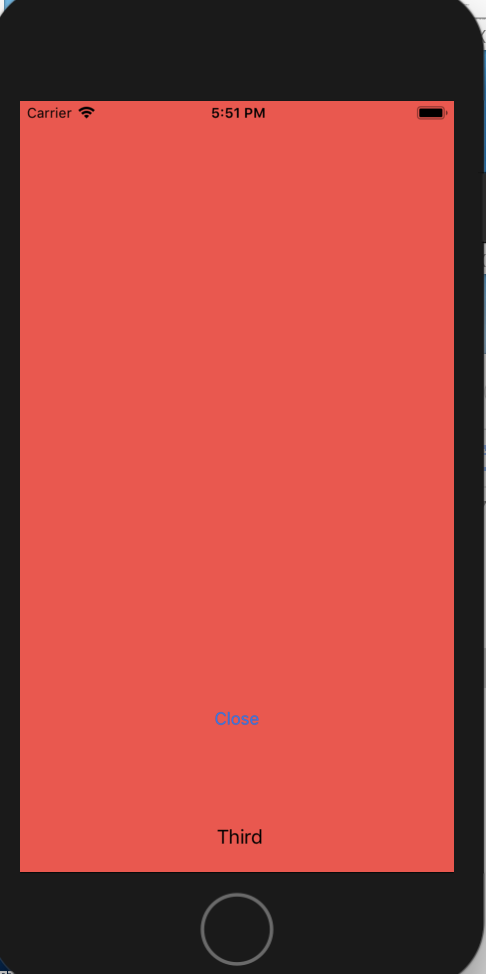
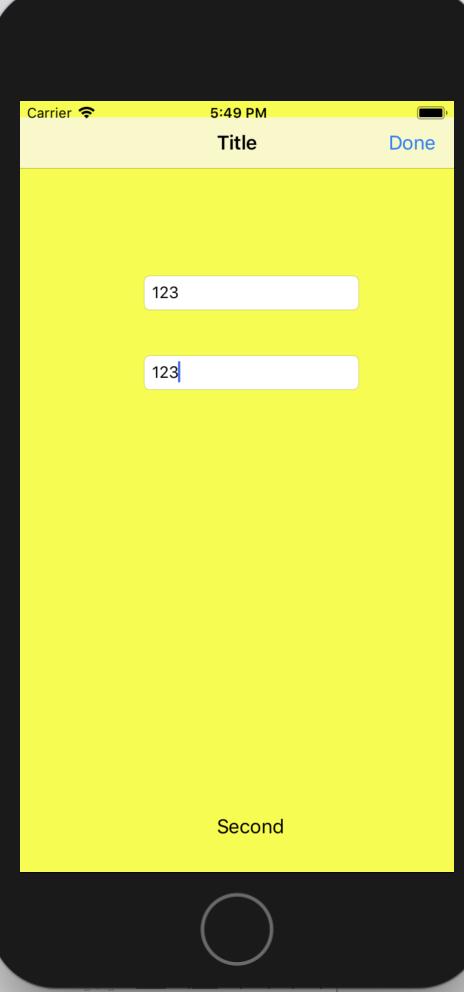
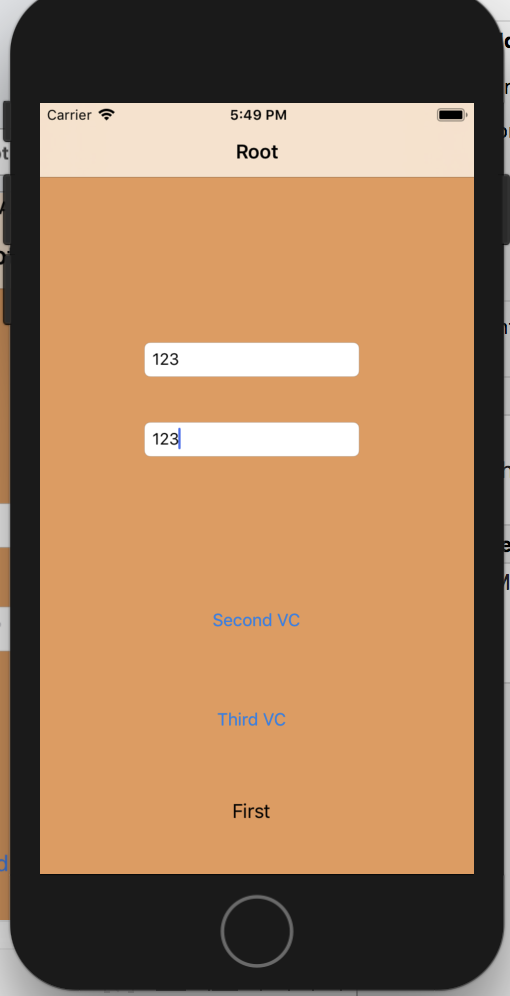
// Pass the selected object to the new view controller.

}

\*/

}

* 运行结果：



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：

//

// AppDelegate.swift

// FoodTracker

//

// Created by student Appleseed on 18/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

//

// MealViewController.swift

// FoodTracker

//

// Created by student Appleseed on 18/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import os.log

class MealViewController: UIViewController, UITextFieldDelegate, UIImagePickerControllerDelegate, UINavigationControllerDelegate {

//MARK: Properties

@IBOutlet weak var nameTextField: UITextField!

@IBOutlet weak var photoImageView: UIImageView!

@IBOutlet weak var ratingControl: RatingControl!

@IBOutlet weak var saveButton: UIBarButtonItem!

/\*

This value is either passed by `MealTableViewController` in `prepare(for:sender:)`

or constructed as part of adding a new meal.

\*/

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

}

// Enable the Save button only if the text field has a valid Meal name.

updateSaveButtonState()

}

//MARK: UITextFieldDelegate

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

}

//MARK: UIImagePickerControllerDelegate

func imagePickerControllerDidCancel(\_ picker: UIImagePickerController) {

// Dismiss the picker if the user canceled.

dismiss(animated: true, completion: nil)

}

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

// The info dictionary may contain multiple representations of the image. You want to use the original.

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 the picker.

dismiss(animated: true, completion: nil)

}

//MARK: Navigation

@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

}

}

//

// RatingControl.swift

// FoodTracker

//

// Created by student Appleseed on 18/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

@IBDesignable class RatingControl: UIStackView {

//MARK: Properties

private var ratingButtons = [UIButton]()

var rating = 0 {

didSet {

updateButtonSelectionStates()

}

}

@IBInspectable var starSize: CGSize = CGSize(width: 44.0, height: 44.0) {

didSet {

setupButtons()

}

}

@IBInspectable var starCount: Int = 5 {

didSet {

setupButtons()

}

}

//MARK: Initialization

override init(frame: CGRect) {

super.init(frame: frame)

setupButtons()

}

required init(coder: NSCoder) {

super.init(coder: coder)

setupButtons()

}

//MARK: Button Action

func ratingButtonTapped(button: UIButton) {

guard let index = ratingButtons.index(of: button) else {

fatalError("The button, \(button), is not in the ratingButtons array: \(ratingButtons)")

}

// Calculate the rating of the selected button

let selectedRating = index + 1

if selectedRating == rating {

// If the selected star represents the current rating, reset the rating to 0.

rating = 0

} else {

// Otherwise set the rating to the selected star

rating = selectedRating

}

}

//MARK: Private Methods

private func setupButtons() {

// Clear any existing buttons

for button in ratingButtons {

removeArrangedSubview(button)

button.removeFromSuperview()

}

ratingButtons.removeAll()

// Load Button Images

let bundle = Bundle(for: type(of: self))

let filledStar = UIImage(named: "filledStar", in: bundle, compatibleWith: self.traitCollection)

let emptyStar = UIImage(named:"emptyStar", in: bundle, compatibleWith: self.traitCollection)

let highlightedStar = UIImage(named:"highlightedStar", in: bundle, compatibleWith: self.traitCollection)

for index in 0..<starCount {

// Create the button

let button = UIButton()

// Set the button images

button.setImage(emptyStar, for: .normal)

button.setImage(filledStar, for: .selected)

button.setImage(highlightedStar, for: .highlighted)

button.setImage(highlightedStar, for: [.highlighted, .selected])

// Add constraints

button.translatesAutoresizingMaskIntoConstraints = false

button.heightAnchor.constraint(equalToConstant: starSize.height).isActive = true

button.widthAnchor.constraint(equalToConstant: starSize.width).isActive = true

// Set the accessibility label

button.accessibilityLabel = "Set \(index + 1) star rating"

// Setup the button action

button.addTarget(self, action: #selector(RatingControl.ratingButtonTapped(button:)), for: .touchUpInside)

// Add the button to the stack

addArrangedSubview(button)

// Add the new button to the rating button array

ratingButtons.append(button)

}

updateButtonSelectionStates()

}

private func updateButtonSelectionStates() {

for (index, button) in ratingButtons.enumerated() {

// If the index of a button is less than the rating, that button should be selected.

button.isSelected = index < rating

// Set accessibility hint and value

let hintString: String?

if rating == index + 1 {

hintString = "Tap to reset the rating to zero."

} else {

hintString = nil

}

let valueString: String

switch (rating) {

case 0:

valueString = "No rating set."

case 1:

valueString = "1 star set."

default:

valueString = "\(rating) stars set."

}

button.accessibilityHint = hintString

button.accessibilityValue = valueString

}

}

}

//

// Meal.swift

// FoodTracker

//

// Created by student Appleseed on 18/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import os.log

class Meal: NSObject, NSCoding {

//MARK: Properties

var name: String

var photo: UIImage?

var rating: Int

//MARK: Archiving Paths

static let DocumentsDirectory = FileManager().urls(for: .documentDirectory, in: .userDomainMask).first!

static let ArchiveURL = DocumentsDirectory.appendingPathComponent("meals")

//MARK: Types

struct PropertyKey {

static let name = "name"

static let photo = "photo"

static let rating = "rating"

}

//MARK: Initialization

init?(name: String, photo: UIImage?, rating: Int) {

// The name must not be empty

guard !name.isEmpty else {

return nil

}

// The rating must be between 0 and 5 inclusively

guard (rating >= 0) && (rating <= 5) else {

return nil

}

// Initialization should fail if there is no name or if the rating is negative.

if name.isEmpty || rating < 0 {

return nil

}

// Initialize stored properties.

self.name = name

self.photo = photo

self.rating = rating

}

//MARK: NSCoding

func encode(with aCoder: NSCoder) {

aCoder.encode(name, forKey: PropertyKey.name)

aCoder.encode(photo, forKey: PropertyKey.photo)

aCoder.encode(rating, forKey: PropertyKey.rating)

}

required convenience init?(coder aDecoder: NSCoder) {

// The name is required. If we cannot decode a name string, the initializer should fail.

guard let name = aDecoder.decodeObject(forKey: PropertyKey.name) as? String else {

os\_log("Unable to decode the name for a Meal object.", log: OSLog.default, type: .debug)

return nil

}

// Because photo is an optional property of Meal, just use conditional cast.

let photo = aDecoder.decodeObject(forKey: PropertyKey.photo) as? UIImage

let rating = aDecoder.decodeInteger(forKey: PropertyKey.rating)

// Must call designated initializer.

self.init(name: name, photo: photo, rating: rating)

}

}

//

// MealTableViewCell.swift

// FoodTracker

//

// Created by student Appleseed on 18/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

class MealTableViewCell: UITableViewCell {

//MARK: Properties

@IBOutlet weak var nameLabel: UILabel!

@IBOutlet weak var photoImageView: UIImageView!

@IBOutlet weak var ratingControl: RatingControl!

override func awakeFromNib() {

super.awakeFromNib()

// Initialization code

}

override func setSelected(\_ selected: Bool, animated: Bool) {

super.setSelected(selected, animated: animated)

// Configure the view for the selected state

}

}

//

// MealTableViewController.swift

// FoodTracker

//

// Created by student Appleseed on 18/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import os.log

class MealTableViewController: UITableViewController {

//MARK: Properties

var meals = [Meal]()

override func viewDidLoad() {

super.viewDidLoad()

// Use the edit button item provided by the table view controller.

navigationItem.leftBarButtonItem = editButtonItem

// Load any saved meals, otherwise load sample data.

if let savedMeals = loadMeals() {

meals += savedMeals

}

else {

// Load the sample data.

loadSampleMeals()

}

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

//MARK: - Table view data source

override func numberOfSections(in tableView: UITableView) -> Int {

return 1

}

override func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return meals.count

}

override func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

// Table view cells are reused and should be dequeued using a cell identifier.

let cellIdentifier = "MealTableViewCell"

guard let cell = tableView.dequeueReusableCell(withIdentifier: cellIdentifier, for: indexPath) as? MealTableViewCell else {

fatalError("The dequeued cell is not an instance of MealTableViewCell.")

}

// Fetches the appropriate meal for the data source layout.

let meal = meals[indexPath.row]

cell.nameLabel.text = meal.name

cell.photoImageView.image = meal.photo

cell.ratingControl.rating = meal.rating

return cell

}

// Override to support conditional editing of the table view.

override func tableView(\_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {

// Return false if you do not want the specified item to be editable.

return true

}

// Override to support editing the table view.

override func tableView(\_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {

if editingStyle == .delete {

// Delete the row from the data source

meals.remove(at: indexPath.row)

saveMeals()

tableView.deleteRows(at: [indexPath], with: .fade)

} else if editingStyle == .insert {

// Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view

}

}

/\*

// Override to support rearranging the table view.

override func tableView(\_ tableView: UITableView, moveRowAt fromIndexPath: IndexPath, to: IndexPath) {

}

\*/

/\*

// Override to support conditional rearranging of the table view.

override func tableView(\_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool {

// Return false if you do not want the item to be re-orderable.

return true

}

\*/

//MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation

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

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

switch(segue.identifier ?? "") {

case "AddItem":

os\_log("Adding a new meal.", log: OSLog.default, type: .debug)

case "ShowDetail":

guard let mealDetailViewController = segue.destination as? MealViewController else {

fatalError("Unexpected destination: \(segue.destination)")

}

guard let selectedMealCell = sender as? MealTableViewCell else {

fatalError("Unexpected sender: \(sender)")

}

guard let indexPath = tableView.indexPath(for: selectedMealCell) else {

fatalError("The selected cell is not being displayed by the table")

}

let selectedMeal = meals[indexPath.row]

mealDetailViewController.meal = selectedMeal

default:

fatalError("Unexpected Segue Identifier; \(segue.identifier)")

}

}

//MARK: Actions

@IBAction func unwindToMealList(sender: UIStoryboardSegue) {

if let sourceViewController = sender.source as? MealViewController, let meal = sourceViewController.meal {

if let selectedIndexPath = tableView.indexPathForSelectedRow {

// Update an existing meal.

meals[selectedIndexPath.row] = meal

tableView.reloadRows(at: [selectedIndexPath], with: .none)

}

else {

// Add a new meal.

let newIndexPath = IndexPath(row: meals.count, section: 0)

meals.append(meal)

tableView.insertRows(at: [newIndexPath], with: .automatic)

}

// Save the meals.

saveMeals()

}

}

//MARK: Private Methods

private func loadSampleMeals() {

let photo1 = UIImage(named: "meal1")

let photo2 = UIImage(named: "meal2")

let photo3 = UIImage(named: "meal3")

guard let meal1 = Meal(name: "Caprese Salad", photo: photo1, rating: 4) else {

fatalError("Unable to instantiate meal1")

}

guard let meal2 = Meal(name: "Chicken and Potatoes", photo: photo2, rating: 5) else {

fatalError("Unable to instantiate meal2")

}

guard let meal3 = Meal(name: "Pasta with Meatballs", photo: photo3, rating: 3) else {

fatalError("Unable to instantiate meal2")

}

meals += [meal1, meal2, meal3]

}

private func saveMeals() {

let isSuccessfulSave = NSKeyedArchiver.archiveRootObject(meals, toFile: Meal.ArchiveURL.path)

if isSuccessfulSave {

os\_log("Meals successfully saved.", log: OSLog.default, type: .debug)

} else {

os\_log("Failed to save meals...", log: OSLog.default, type: .error)

}

}

private func loadMeals() -> [Meal]? {

return NSKeyedUnarchiver.unarchiveObject(withFile: Meal.ArchiveURL.path) as? [Meal]

}

}

//

// FoodTrackerTests.swift

// FoodTrackerTests

//

// Created by student Appleseed on 18/12/5.

// Copyright © 2018年 fl. All rights reserved.

//

import XCTest

@testable import FoodTracker

class FoodTrackerTests: XCTestCase {

//MARK: Meal Class Tests

// Confirm that the Meal initializer returns a Meal object when passed valid parameters.

func testMealInitializationSucceeds() {

// Zero rating

let zeroRatingMeal = Meal.init(name: "Zero", photo: nil, rating: 0)

XCTAssertNotNil(zeroRatingMeal)

// Positive rating

let positiveRatingMeal = Meal.init(name: "Positive", photo: nil, rating: 5)

XCTAssertNotNil(positiveRatingMeal)

}

// Confirm that the Meal initialier returns nil when passed a negative rating or an empty name.

func testMealInitializationFails() {

// Negative rating

let negativeRatingMeal = Meal.init(name: "Negative", photo: nil, rating: -1)

XCTAssertNil(negativeRatingMeal)

// Rating exceeds maximum

let largeRatingMeal = Meal.init(name: "Large", photo: nil, rating: 6)

XCTAssertNil(largeRatingMeal)

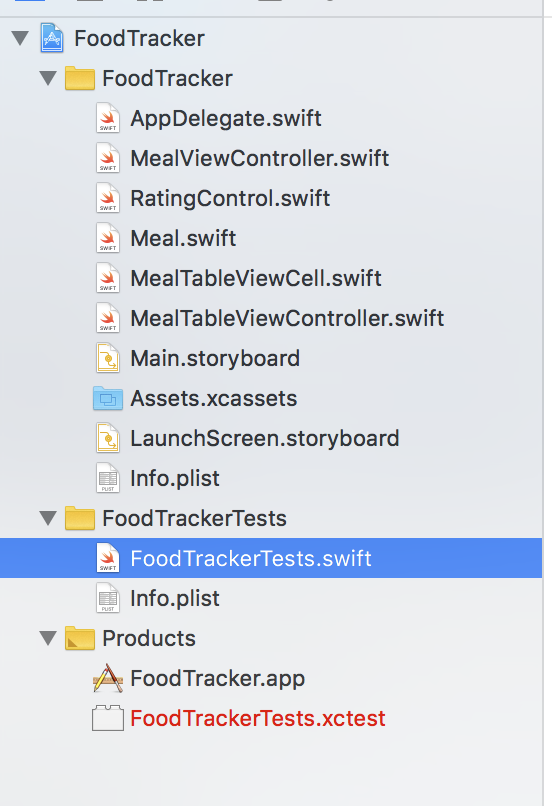
// Empty String

let emptyStringMeal = Meal.init(name: "", photo: nil, rating: 0)

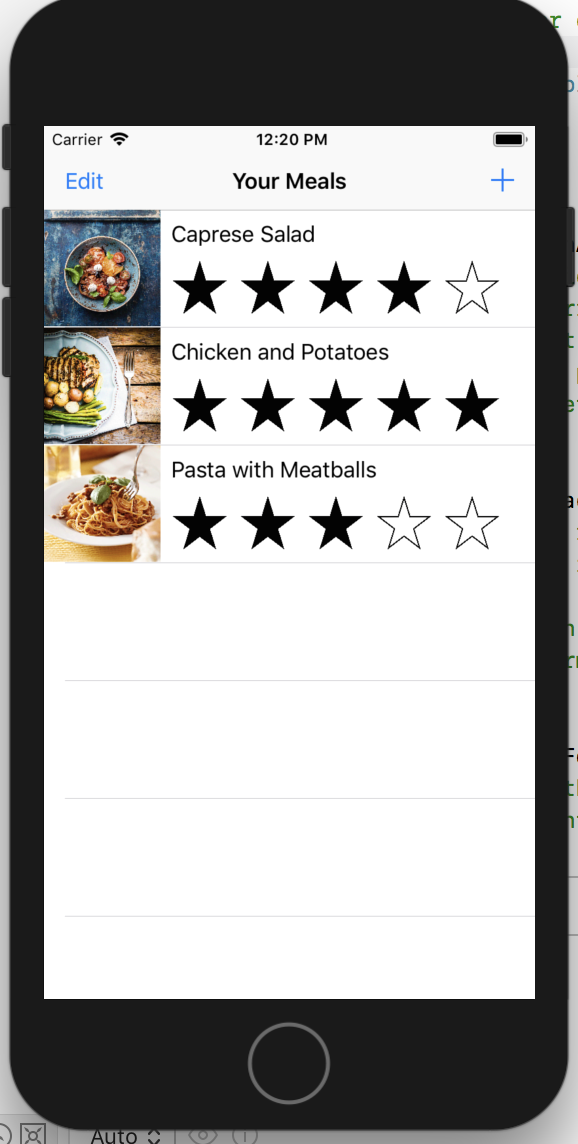
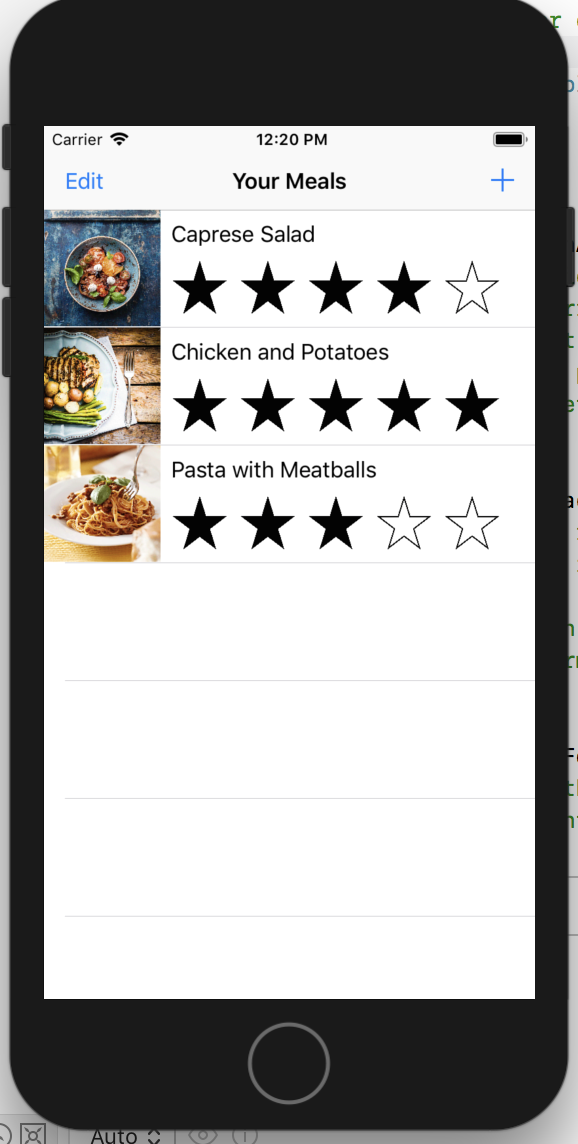
XCTAssertNil(emptyStringMeal)

}

}



* 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）

这一章开始对我与来说比较难，一下接触多mvc，多控制器，有点乱。最后把老师关于这一章的录屏看了很多遍，我觉得超级有用，现在对传参，多mvc页面调用理解也比较透彻了。

Github地址：

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。