实验编号： 12 **四川师大《IOS》实验报告 2018** 年 **11** 月 **28** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 数据持久化 \_

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

**实验 十二 \_\_\_\_\_\_**数据持久化**\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 掌握iOS中数据持久化的使用；
3. 掌握SQLite和CoreData的原理和使用；
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. 在命令提示符下熟悉并使用SQLite命令
   1. 新建sqlite3数据库；
   2. 新建Person表；
   3. 插入新的数据；
   4. 查询数据；
9. 采用多MVC和SQLite或Core Data构造一个Person信息维护程序，要求：
   1. 可插入新数据；
   2. 可删除修改原数据；
   3. 可查询特定数据；
10. 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）
11. 在命令提示符下熟悉并使用SQLite命令
    1. 新建sqlite3数据库；
    2. 新建Person表；
    3. 插入新的数据；
    4. 查询数据；

* 程序代码：

在命令提示符窗口代码如下：

Sqlite3 person.db;

.table

.dump

Create table person(name varchar(20),phone varchar(20));

Insert into person(name,phone) values(‘gzq’,’000000’)

Select \* from person;

* 运行结果：



1. 采用多MVC和SQLite或Core Data构造一个Person信息维护程序，要求：
   1. 可插入新数据；
   2. 可删除修改原数据；
   3. 可查询特定数据；
      * 程序代码：

CoreData:

personList界面：

//

//  PersonsTableViewController.swift

//  MultiCoreData

//

//  Created by Guozhouqian on 2018/12/15.

//  Copyright © 2018 Guozhouqian. All rights reserved.

//

import UIKit

import CoreData

class PersonsTableViewController: UITableViewController {

    let appDelegate = UIApplication.shared.delegate as! AppDelegate

    let context = (UIApplication.shared.delegate as! AppDelegate).persistentContainer.viewContext

    var persons:[Person]?

    fileprivate func reloadData() {

        let fetch:NSFetchRequest<Person> = Person.fetchRequest()

        //按照姓名升序排序

        fetch.sortDescriptors = [NSSortDescriptor(key: "name", ascending: true)]

        persons = try? context.fetch(fetch)

        tableView.reloadData()

    }

    override func viewDidLoad() {

        super.viewDidLoad()

        reloadData()

        // Uncomment the following line to preserve selection between presentations

        // self.clearsSelectionOnViewWillAppear = false

        // Uncomment the following line to display an Edit button in the navigation bar for this view controller.

        // self.navigationItem.rightBarButtonItem = self.editButtonItem

    }

    override func viewWillAppear(\_ animated: Bool) {

        reloadData()

    }

    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 {

        // #warning Incomplete implementation, return the number of sections

        return 1

    }

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

        // #warning Incomplete implementation, return the number of rows

        return persons?.count ?? 0

    }

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

        let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)

        if let p = persons?[indexPath.row]{

             cell.textLabel?.text = p.name

        }

        // Configure the cell...

        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

            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?) {

        if segue.identifier == "ShowDetail" {

            let secVc = segue.destination as! DetailViewController

            if let indexPath = tableView.indexPath(for: sender as! UITableViewCell),let person = persons?[indexPath.row]{

                secVc.person = person

            }

        }

    }

}

detail界面：

//

//  DetailViewController.swift

//  MultiCoreData

//

//  Created by Guozhouqian on 2018/12/15.

//  Copyright © 2018 Guozhouqian. All rights reserved.

//

import UIKit

class DetailViewController: UIViewController {

    var person:Person?

    @IBOutlet weak var lbName: UILabel!

    @IBOutlet weak var lbPhone: UILabel!

    override func viewDidLoad() {

        super.viewDidLoad()

        lbName.text = person?.name

        lbPhone.text = person?.phone

    }

    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

    /\*

    // 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.destinationViewController.

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

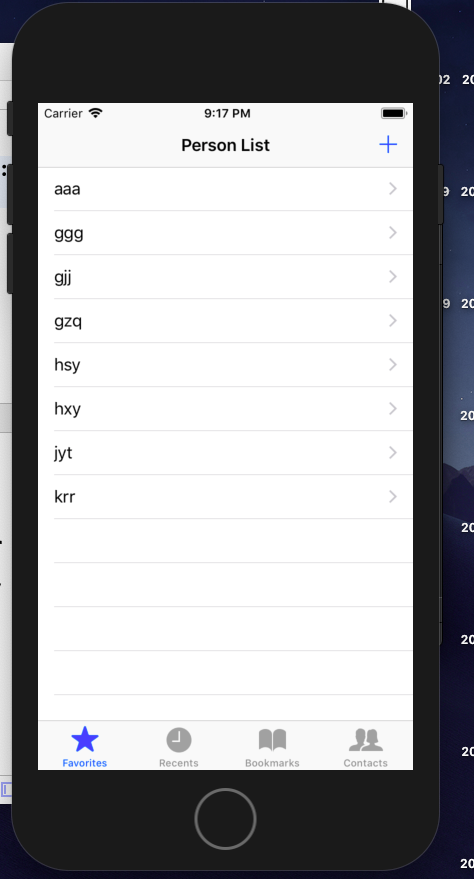
    }

    \*/

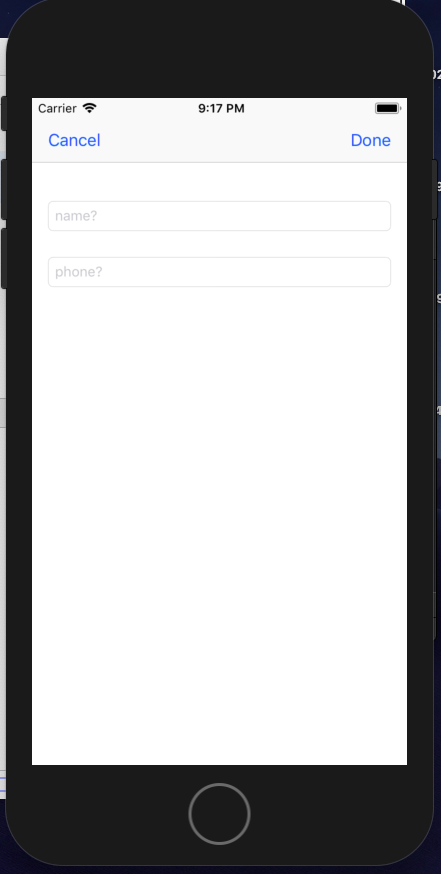
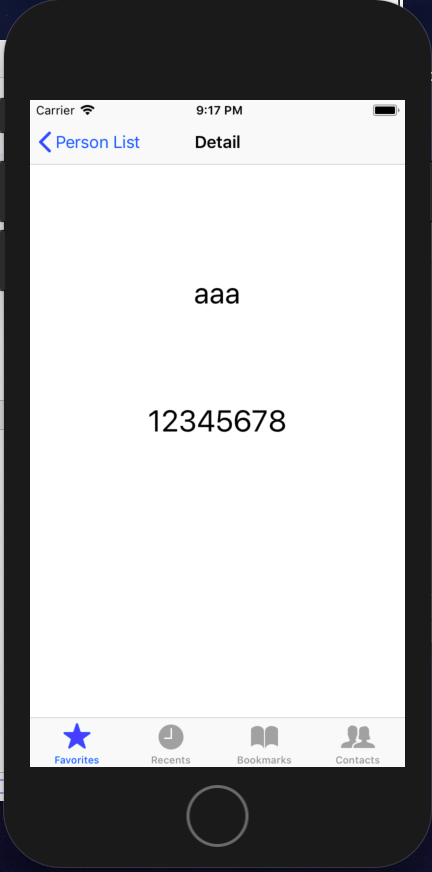
}

* + - 运行结果：

从数据库中读数据：



添加数据，以及显示详细数据：

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：

本次实验主要学习的是数据库方面的内容一个是个sqlite3，另一个CoreData，sqlite还是是数据库操作居多，代码中有很多的sql语句，而CoreData主要是面向对象的编程，代码中并没有很多的sql语句，这样看来，其实我们更应该用coreData来操作我们的数据，这样更方便，不会因为sql语句的问题而出错。

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。