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：\_\_蒋星\_ 学号：2016110418 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：

Sqlite3 my.db

Create table Person(

Name varchar(30),

Age int not null

)

Insert into Person values(‘jxkicker’,15);

Insert into Person values(‘wangzun,25);

.dump

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：

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

let person = Person(context:context)

person.name = tfName.text

person.phone = tfNumber.text

appDelegate.saveContext()

}

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

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

fetch.predicate = NSPredicate(format:"name=%@",tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first{

context.delete(p)

appDelegate.saveContext()

}

}

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

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

fetch.predicate = NSPredicate(format:"name=%@",tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first{

p.phone = tfNumber.text

appDelegate.saveContext()

}

}

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

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

fetch.predicate = NSPredicate(format:"name=%@",tfName.text!)

let persons = try? context.fetch(fetch)

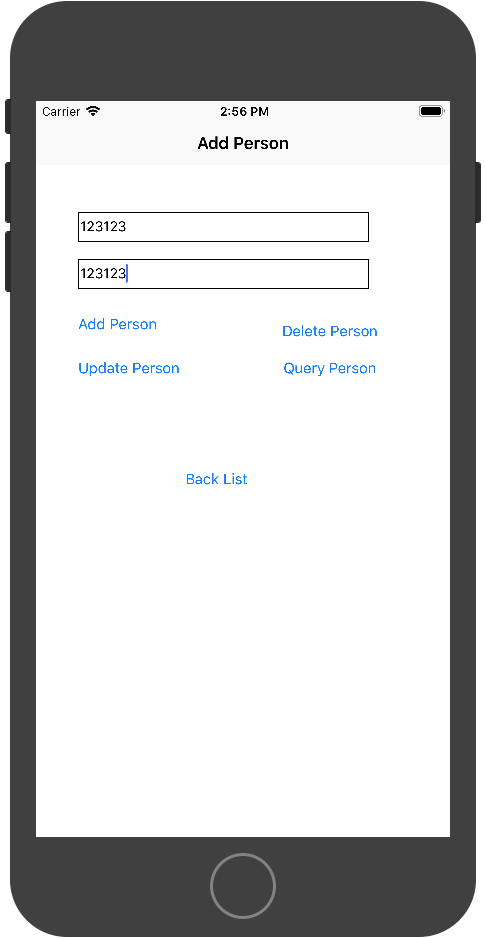
if let p = persons?.first{

tfNumber.text = p.phone

}

* + - 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：

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。