**课 程 设 计 报 告**

**课程设计名称：**

移动应用开发

**课程设计题目**：

通讯录

院（系）：计算机与信息工程学院

专 业：软件工程

班 级：2014软件工程

学 号：20141105044

姓 名： 严伟

指导教师： 朝力萌

完成日期：2016.7.14

**目 录**

[第1章 概要设计 2](#_Toc406624046)

[1.1题目的内容与要求 2](#_Toc406624047)

[1.2总体结构 2](#_Toc406624048)

[第2章 详细设计 2](#_Toc406624049)

[2.1主模块 2](#_Toc406624050)

[第3章 调试分析 3](#_Toc406624051)

[第4章 使用说明与执行结果 4](#_Toc406624052)

# 第1章 概要设计

## 1.1题目的内容与要求

此次程序设计的主要内容就是更好的熟悉ios环境下xcode的编程环境，学习并理解使用swift语言在ios环境下编写程序。要求调试并连接数据库，并使用swift语言编写一个简易的通讯录，能够实现储存联系人，查找联系人，删除联系人等功能。

## 1.2总体结构

删除联系人：按名字删除，按号码删除和返回button

输入姓名和电话的text view框

添加联系人姓名，电话，邮箱，地址的text view框。

保存和返回button

主界面

添加联系人

删除联系人

查询联系人

显示联系人

显示联系人：显示所有联系人，清除联系人信息和清空所有联系人的button

显示联系人信息的text view框

查询联系人：按名字查询和返回button

输入名字和显示所查联系人信息的text view框

# 第2章 详细设计

## 2.1源代码

import UIKit  
  
class ViewController: UIViewController {  
      
    var db:SQLiteDB!  
      
    @IBOutlet weak var txtUname: UITextField!  
    @IBOutlet weak var txtMobile: UITextField!   
    @IBOutlet weak var email: UITextField!  
    @IBOutlet weak var address: UITextField!  
     
    override func viewDidLoad() {  
        super.viewDidLoad()  
          
        //获取数据库实例  
        db = SQLiteDB.sharedInstance()  
        //如果表还不存在则创建表（其中uid为自增主键）  
        db.execute("create table if not exists tuser(uid integer primary key,uname varchar(20),mobile varchar(20),email varchar(20),address varchar(20))")  
        //如果有数据则加载  
       // initUser()  
    }  
      
    //点击保存  
    @IBAction func saveClicked(sender: AnyObject) {  
        saveUser()  
    }  
      
    //从SQLite加载数据  
    func initUser() {  
        let data = db.query("select \* from tuser")  
        if data.count > 0 {  
            //获取最后一行数据显示  
            let user = data[data.count - 1]  
            txtUname.text = user["uname"] as? String  
            txtMobile.text = user["mobile"] as? String  
            address.text = user["email"] as? String  
            email.text = user["address"] as? String  
        }  
    }  
      
    //保存数据到SQLite  
    func saveUser() {  
        let uname = self.txtUname.text!  
        let mobile = self.txtMobile.text!  
        let email = self.email.text!  
        let address = self.address.text!  
  
        //插入数据库，这里用到了esc字符编码函数，其实是调用bridge.m实现的  
        let sql = "insert into tuser(uname,mobile,email,address) values('\(uname)','\(mobile)','\(email)','\(address)')"  
        print("sql: \(sql)")  
        //通过封装的方法执行sql  
        let result = db.execute(sql)  
        print(result)  
    }  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
    }  
}  
  
import UIKit  
  
class delete: UIViewController {  
      
    var db:SQLiteDB!  
      
    @IBOutlet weak var text1: UITextField!  
    @IBOutlet weak var text2: UITextField!  
      
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
        //获取数据库实例  
        db = SQLiteDB.sharedInstance()  
        //如果表还不存在则创建表（其中uid为自增自减）  
        db.execute("create table if not exists tuser(uid integer primary key,uname varchar(20),mobile varchar(20),email varchar(20),address varchar(20))")  
        //如果有数据则加载  
        //initUser()  
    }  
    //点击保存  
      
    @IBAction func deletename(sender: AnyObject) {  
        deleteuser()  
    }  
    func deleteuser(){  
        let value=text1.text!  
        let sql = "delete from tuser where mobile='\(value)' "  
        let result = db.execute(sql)  
        print(result)  
    }  
  
    @IBAction func deletenum(sender: AnyObject) {  
        deletenum()  
    }  
    func deletenum(){  
        let value1=text2.text!  
        let sql = "delete from tuser where mobile='\(value1)' "  
        let result = db.execute(sql)  
        print(result)  
    }  
      
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }      
}  
  
import UIKit  
  
class search: UIViewController {  
      
    var db:SQLiteDB!  
      
    @IBOutlet weak var text3: UITextField!  
    @IBOutlet weak var text4: UITextView!  
      
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
        //获取数据库实例  
        db = SQLiteDB.sharedInstance()  
        //如果表还不存在则创建表（其中uid为自增自减）  
        db.execute("create table if not exists tuser(uid integer primary key,uname varchar(20),mobile varchar(20),email varchar(20),address varchar(20))")  
        //如果有数据则加载  
        //initUser()  
    }  
      
    @IBAction func srarchname(sender: AnyObject) {  
        sn()  
    }  
    func sn(){  
      
        text4.text=""  
        let a=text3.text!  
        let data = db.query("select \* from tuser where uname='\(a)'")  
        for (var i=0;i<data.count;i++)  
        {  
        let user = data[i]  
        text4.text!+="姓名: "+String(user["uname"]!)+"电话: "+String(user["mobile"]!)+"邮箱: "+String(user["email"]!)+"地址: "+String(user["address"]!)+"\n"  
        }  
    }  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }      
}

import UIKit  
  
class show: UIViewController {  
    var db:SQLiteDB!  
    @IBAction func qingchu(sender: AnyObject) {  
        text5.text=""  
    }  
    @IBAction func qingkong(sender: AnyObject) {  
        let sql = "delete from tuser"  
        let result = db.execute(sql)  
        print(result)  
    }

    @IBOutlet weak var text5: UITextView!  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        //获取数据库实例  
        db = SQLiteDB.sharedInstance()  
        //如果表还不存在则创建表（其中uid为自增自减）  
  db.execute("create table if not exists tuser(uid integer  primary key,uname varchar(20),mobile varchar(20),email var char(20),address varchar(20))")  
        //如果有数据则加载  
        //initUser()  
    }  
    @IBAction func showuser(sender: AnyObject) {  
        su()  
    }  
    func su(){  
        let data = db.query("select \* from tuser")  
        for var j=0;j<data.count;j++  
        {  
            let user = data[j]  
            text5.text!+="姓名: "+String(user["uname"]!)+"电话: "+String(user["mobile"]!)+"邮箱: "+String(user["email"]!)+"地址: "+String(user["address"]!)+"\n"  
        }  
    }  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
}

# 第3章 调试分析 import UIKit class ViewController: UIViewController {          var db:SQLiteDB!          @IBOutlet weak var txtUname: UITextField!     @IBOutlet weak var txtMobile: UITextField!      @IBOutlet weak var email: UITextField!     @IBOutlet weak var address: UITextField!         override func viewDidLoad() {         super.viewDidLoad()                  //获取数据库实例         db = SQLiteDB.sharedInstance()         //如果表还不存在则创建表（其中uid为自增主键）         db.execute("create table if not exists tuser(uid integer primary key,uname varchar(20),mobile varchar(20),email varchar(20),address varchar(20))")         //如果有数据则加载        // initUser()     }          //点击保存     @IBAction func saveClicked(sender: AnyObject) {         saveUser()     }          //从SQLite加载数据     func initUser() {         let data = db.query("select \* from tuser")         if data.count > 0 {             //获取最后一行数据显示             let user = data[data.count - 1]             txtUname.text = user["uname"] as? String             txtMobile.text = user["mobile"] as? String             address.text = user["email"] as? String             email.text = user["address"] as? String         }     }          //保存数据到SQLite     func saveUser() {         let uname = self.txtUname.text!         let mobile = self.txtMobile.text!         let email = self.email.text!         let address = self.address.text!         //插入数据库，这里用到了esc字符编码函数，其实是调用bridge.m实现的         let sql = "insert into tuser(uname,mobile,email,address) values('\(uname)','\(mobile)','\(email)','\(address)')"         print("sql: \(sql)")         //通过封装的方法执行sql         let result = db.execute(sql)         print(result)     }     override func didReceiveMemoryWarning() {         super.didReceiveMemoryWarning()     } }

# 因为这次做的通讯录是多个页面，要实现页面的跳转，将两个页面之间用button选择model连接，就可以实现页面的跳转。其次，一个页面对应相应的代码，所以建立4个file.swift文件，四个页面分别为：ViewController，delete，select，show，四个页面对应相应的实现代码。以上是储存联系人页面的代码。首先要添加四个输入姓名，电话，邮箱，地址的text view框，将之与页面绑定。设定一个保存button，在其中写一个saveUser函数，将之保存到数据库中。

**删除联系人：**

import UIKit  
  
class delete: UIViewController {  
      
    var db:SQLiteDB!  
      
    @IBOutlet weak var text1: UITextField!  
    @IBOutlet weak var text2: UITextField!  
      
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
        //获取数据库实例  
        db = SQLiteDB.sharedInstance()  
        //如果表还不存在则创建表（其中uid为自增自减）  
        db.execute("create table if not exists tuser(uid integer primary key,uname varchar(20),mobile varchar(20),email varchar(20),address varchar(20))")  
        //如果有数据则加载  
        //initUser()  
    }  
    //点击保存  
      
    @IBAction func deletename(sender: AnyObject) {  
        deleteuser()  
    }  
    func deleteuser(){  
        let value=text1.text!  
        let sql = "delete from tuser where mobile='\(value)' "  
        let result = db.execute(sql)  
        print(result)  
    }  
  
    @IBAction func deletenum(sender: AnyObject) {  
        deletenum()  
    }  
    func deletenum(){  
        let value1=text2.text!  
        let sql = "delete from tuser where mobile='\(value1)' "  
        let result = db.execute(sql)  
        print(result)  
    }  
      
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }      
}  
创建两个输出名字和电话的text view框，将之与页面绑定。创建两个按名字删除，按电话删除button按钮，编写两个deleteuser，deletenum函数，实现删除功能。

**查找联系人：**

import UIKit  
  
class search: UIViewController {  
    var db:SQLiteDB!  
   
    @IBOutlet weak var text3: UITextField!  
    @IBOutlet weak var text4: UITextView!  
      
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
        //获取数据库实例  
        db = SQLiteDB.sharedInstance()  
        //如果表还不存在则创建表（其中uid为自增自减）  
        db.execute("create table if not exists tuser(uid integer primary key,uname varchar(20),mobile varchar(20),email varchar(20),address varchar(20))")  
        //如果有数据则加载  
        //initUser()  
    }  
    @IBAction func srarchname(sender: AnyObject) {  
        sn()  
    }  
    func sn(){  
        text4.text=""  
        let a=text3.text!  
        let data = db.query("select \* from tuser where uname='\(a)'")  
        for (var i=0;i<data.count;i++)  
        {  
        let user = data[i]  
        text4.text!+="姓名: "+String(user["uname"]!)+"电话: "+String(user["mobile"]!)+"邮箱: "+String(user["email"]!)+"地址: "+String(user["address"]!)+"\n"  
        }  
    }  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }      
}

创建两个text view，一个用于输入名字，一个用于显示联系人信息。创建一个按名字查找button，编写sn函数实现从数据库中查找联系人信息的功能。

**显示联系人：**

import UIKit  
class show: UIViewController {  
    var db:SQLiteDB!  
    @IBAction func qingchu(sender: AnyObject) {  
        text5.text=""  
    }  
    @IBAction func qingkong(sender: AnyObject) {  
        let sql = "delete from tuser"  
        let result = db.execute(sql)  
        print(result)  
    }

    @IBOutlet weak var text5: UITextView!  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        //获取数据库实例  
        db = SQLiteDB.sharedInstance()  
        //如果表还不存在则创建表（其中uid为自增自减）  
  db.execute("create table if not exists tuser(uid integer  primary key,uname varchar(20),mobile varchar(20),email var char(20),address varchar(20))")  
        //如果有数据则加载  
        //initUser()  
    }  
    @IBAction func showuser(sender: AnyObject) {  
        su()  
    }  
    func su(){  
        let data = db.query("select \* from tuser")  
        for var j=0;j<data.count;j++  
        {  
            let user = data[j]  
            text5.text!+="姓名: "+String(user["uname"]!)+"电话: "+String(user["mobile"]!)+"邮箱: "+String(user["email"]!)+"地址: "+String(user["address"]!)+"\n"  
        }  
    }  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
}

创建一个text view，将之与页面绑定，来显示所有人信息。创建显示所有人，清空所有联系人button，编写一个su函数实现显示所有联系人的功能。

# 第4章 使用说明与执行结果

运行操作及结果:



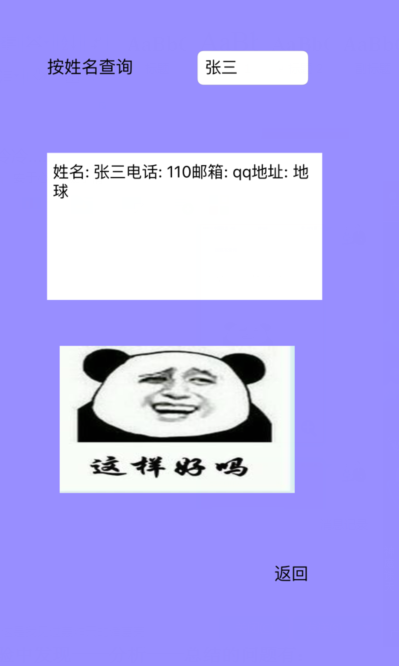
添加一个联系人



显示所有联系人和清空所有联系人



查找联系人



删除联系人



**二．课程设计小结**

**在实验中发现——分析——总结的问题有：**

移动应用开发课程是培养我们综合运用所学知识，发现，提出，分析和解决实际问题，锻炼实际能力的重要环节，是对我们实际工作能力的具体训练和考察过程。经过每天反复的看视频学习，和自己动手实际上机练习，基本掌握了在ios环境下xcode中使用swift编写程序的方法。此次编写了一个简易的通讯录，不仅有基本的添加联系人，储存联系人功能，还学会最主要的连接调试数据库，按名字删除，按号码删除，显示所有联系人，清空所有联系人等功能。但是遇到的问题更多，最直观的就是界面过于简陋，虽然会修改页面的背景颜色和添加图片，但是离一个基本的APP软件界面还差的很远。其次就是各个button和text view绑定的时候，会出现各种各样让人头疼的问题，例如第一次绑定错了，第二次绑定，需要把它之前的所有东西都删掉，不然程序就会报错，而且还会莫名其妙的出现段点，编译就会无限报错。还有，在编译的时候使用iphone6，程序编写没有错误，编译成功，但是没把联系人信息存到数据库里，找遍了所有地方都没错，最后换成iphone5编译就成功了。许多问题让我很烦躁，但慢慢一个一个解决之后，都会有成就感，我也体会到了对于swift语言的学习还有很多不足，要坚持学习。

**参考文献：**

**[1] 伍星,罗飞.** **Swift实战入门.人民邮电出版社，2015**

**[2]关东升,赵志荣.** **Swift开发入门.人民邮电出版社，2014**

Github网址：

**https://github.com/yan786362806/SQLtxl/tree/master/SQLtxl**