|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **中国地质大学（北京）** | | | |
| **嵌入式开发技术课程实践报告** | | | |
|  | | | |
|  |  |
|  |  |
|  |  |
|  |  |
| 学号： | **1008151221** |
| 姓名： | **杨桦桉** |
| 完成时间： | **2017年11月5日** |

|  |
| --- |
|  |

**目录**

[1 需求分析 1](#_Toc497242450)

[1.1 功能需求 1](#_Toc497242451)

[1.2 软件功能树 1](#_Toc497242452)

[2 应用设计 2](#_Toc497242453)

[2.1 类设计 2](#_Toc497242454)

[2.2 界面设计 2](#_Toc497242455)

[2.2.1 账单界面（detailsView） 2](#_Toc497242456)

[2.2.2 图表界面 3](#_Toc497242457)

[2.2.3 记账界面 3](#_Toc497242458)

[2.2.4 工具界面 4](#_Toc497242459)

[2.2.5 我的界面 4](#_Toc497242460)

[2.2.6 登录界面 4](#_Toc497242461)

[2.2.7 编辑界面 5](#_Toc497242462)

[2.3 连接设计 5](#_Toc497242463)

[2.3.1 明细（detialsView） 5](#_Toc497242464)

[2.3.2 图表（ChartView） 6](#_Toc497242465)

[2.3.3 记账（addRecordView） 7](#_Toc497242466)

[2.3.4 工具（toolsView） 8](#_Toc497242467)

[2.3.5 我的（meView） 9](#_Toc497242468)

[2.3.6 登录（loginView） 9](#_Toc497242469)

[2.3.7 编辑（editView） 10](#_Toc497242470)

[2.4 界面跳转设计 11](#_Toc497242471)

[3 应用实现 12](#_Toc497242472)

[3.1 DataBaseController 12](#_Toc497242473)

[3.2 detailsViewController 13](#_Toc497242474)

[3.3 chartViewController 14](#_Toc497242475)

[3.4 addRecordViewController 14](#_Toc497242476)

[3.5 toolsViewController 14](#_Toc497242477)

[3.6 meViewController 14](#_Toc497242478)

[3.7 loginViewController 15](#_Toc497242479)

[3.8 editViewController 15](#_Toc497242480)

[4 应用截图 16](#_Toc497242481)

[4.1 明细界面 16](#_Toc497242482)

[4.2 图表界面 17](#_Toc497242483)

[4.3 记账界面 17](#_Toc497242484)

[4.4 工具界面 18](#_Toc497242485)

[4.5 我的界面 18](#_Toc497242486)

[4.6 登录和注册界面 19](#_Toc497242487)

[4.7 编辑界面 19](#_Toc497242488)

[5 对项目中使用的开源项目说明 20](#_Toc497242489)

[5.1 MJRefresh 20](#_Toc497242490)

[5.2 SQLiteDB 20](#_Toc497242491)

[5.3 Charts-iOS 20](#_Toc497242492)

[6 开发过程中遇到的问题和解决方案 20](#_Toc497242493)

[6.1 明细界面在调试多次后出现崩溃无法显示的情况 20](#_Toc497242494)

[6.2 CollectionView的scroll View全屏滚动遮挡其他试图 21](#_Toc497242495)

[6.3 数据库显示加密或不存在，无法链接数据库 21](#_Toc497242496)

[6.4 计算每天收入、支出总额时无法正确读取数据 21](#_Toc497242497)

[6.5 从“明细”页面到“编辑”页面无法通过提醒框跳转问题 21](#_Toc497242498)

[6.6 当在弹出菜单点击“删除”键后，再次长按弹出菜单偏移至左上角的问题 22](#_Toc497242499)

[6.7 场景跳转及传值问题 22](#_Toc497242500)

[7 总结 23](#_Toc497242501)

# 需求分析

随着移动支付的兴起，支付宝、微信等移动支付方式正逐步取代现金支付的方式。在移动支付给商家和客户带来便利的同时，这种看得见却摸不着的支付方式，让很多人渐渐淡化了对每一笔支出和收入的记忆能力，久而久之会产生“自己的钱都花到哪里去了”这样的疑惑。因此，需要一款应用，可以帮助人们记录每一笔支出和收入，并对这些支付记录进行分析和统计，帮助人们更好的理财，帮助人们管好自己的资金。

## 功能需求

开发一款记账App，可以按照收支类型，记录用户每天的每一笔支出和收入，并对每天的支出情况进行统计。以一个月为时间节点，向用户展示月账单，并统计该月的支出情况，从每天的收支情况和收支类型两个方面，统计用户的收支数据，并以图表的形式呈现。提供常用工具，如二手市场、快递查询、汇率查询等。实现用户管理。

## 软件功能树

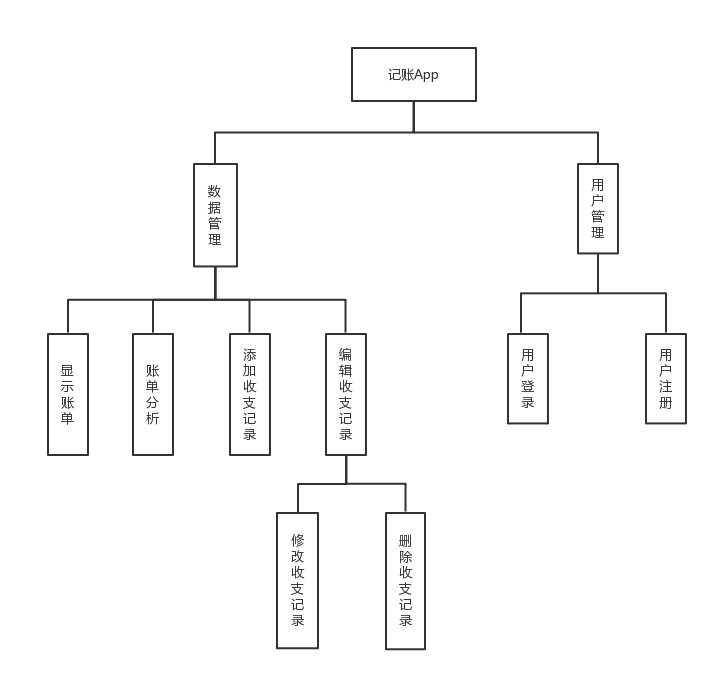


图 1 应用功能树

# 应用设计

## 类设计

表格 1 类设计表

|  |  |
| --- | --- |
| 类名 | 作用 |
| dataRecord | 存储一条支付记录 |
| user | 存储用户信息 |
| dataBaseController | 封装了对SQLite数据库的链接，以及增、删、改、查等方法 |
| detailsViewController | 实现detailsView逻辑功能的控制器类 |
| chartViewController | 实现chartView逻辑功能的控制器类 |
| addRecordViewController | 实现addRecordView逻辑功能的控制器类 |
| toolsViewController | 实现toolsView逻辑功能的控制器类 |
| meViewController | 实现meView逻辑功能的控制器类 |
| loginVeiwController | 实现loginView逻辑功能的控制器类 |
| editViewController | 实现editViewController逻辑功能的控制器类 |

## 界面设计

### 账单界面（detailsView）

表格 2 账单界面

|  |  |  |
| --- | --- | --- |
| 控件类型 | 数量 | 说明 |
| View | 4 | 1个为日期选择器窗口  1个为弹出菜单窗口  1个为提示框  1个为显示当前月总收支情况窗口 |
| CollectionView | 1 | 呈现账单记录以及每天开销情况 |
| Label | 17 | 8个位于collectionView中，用来显示每天的收支情况以及每条记录信息  2个位于提示窗口，用于显示系统提示信息  2个位于弹出菜单窗口，用于显示选项件的分割线  5个位于显示当月总收支情况窗口 |
| Button | 8 | 2个确定按钮  3个取消按钮  1个删除按钮  1个修改按钮  1个选择日期按钮 |
| DatePickerView | 1 | 用来选择提起 |

### 图表界面

表格 3图表界面

|  |  |  |
| --- | --- | --- |
| 控件类型 | 数量 | 说明 |
| View | 3 | 1个显示当月平均值和总收\支情况信息  1个选择记录类型为收入或支出窗口  1个日期选择窗口 |
| PieChartView | 1 | 呈现饼图 |
| LineCharView | 1 | 呈现折线图 |
| Label | 7 | 5个用于显示当月平均值和总收\支情况信息  2个位于选择收支类型窗口 |
| Button | 5 | 1个用于选择日期  1个确认按钮  1个取消按钮  2个用于选择收入或支出 |

### 记账界面

表格 4 记账界面

|  |  |  |
| --- | --- | --- |
| 控件类型 | 数量 | 说明 |
| View | 1 | 日期选择窗口 |
| Label | 5 | 5个用于显示当输入值的类型 |
| Button | 6 | 1个用于选择日期  1个确认按钮  1个取消按钮  2个用于选择收入或支出  1个保存安妮 |
| TextField | 5 | 用来输入信息 |

### 工具界面

表格 5 工具界面

|  |  |  |
| --- | --- | --- |
| 控件类型 | 数量 | 说明 |
| View | 1 | 日期选择窗口 |
| Button | 3 | 3个按钮用来显示不同的网页 |
| WebView | 1 | 显示网页 |

### 我的界面

表格 6 我的界面

|  |  |  |
| --- | --- | --- |
| 控件类型 | 数量 | 说明 |
| ImageView | 1 | 显示头像 |
| Button | 1 | 跳转登录界面 |
| Label | 1 | 显示用户名 |

### 登录界面

表格 7登录界面

|  |  |  |
| --- | --- | --- |
| 控件类型 | 数量 | 说明 |
| View | 2 | 登录窗口和注册窗口 |
| Button | 3 | 1个登录按钮  2个注册 |
| Label | 5 | 提示输入信息 |
| TextField | 5 | 输入信息 |
| ImageView | 1 | 头像 |

### 编辑界面

表格 8 编辑界面

|  |  |  |
| --- | --- | --- |
| 控件类型 | 数量 | 说明 |
| View | 1 | 日期选择窗口 |
| Label | 5 | 5个用于显示当输入值的类型 |
| Button | 7 | 1个用于选择日期  1个确认按钮  1个取消按钮  2个用于选择收入或支出  1个保存按钮  1个修改按钮 |
| TextField | 5 | 用来输入信息 |

## 连接设计

### 明细（detialsView）

#### 输出口

表格 9 明细界面输出口表

|  |  |  |
| --- | --- | --- |
| IBOutlet | 类型 | 说明 |
| collectionView | UICollectionView | 呈现账单记录 |
| yearLabel | UILabel | 显示年份 |
| incomeLabel | UILabel | 显示收入 |
| outcomeLabel | UILabel | 显示支出 |
| dataView | UIView | 选择日期界面 |
| dataPicker | UIDatePicker | 日期选择器 |
| dateBtn | UIButton | 呈现选择日期界面 |
| alertView | UIView | 提示界面 |
| alterConfirmView | UIView | 系统提示界面 |

#### 操作

表格 10明细界面操作表

|  |  |  |
| --- | --- | --- |
| IBAction | 连接控件类型 | 说明 |
| confirmDate(\_ sender：Any) | UIButton | 存储选择的日期并设置界面显示年月的标签，重新加载数据 |
| concel(\_ sender：Any) | UIButton | 取消选择日期操作，隐藏选择日期界面 |
| setDate(\_ sender：Any) | UIButton | 呈现选择日期界面 |
| confirmDelete(\_ sender：Any) | UIButton | 删除选中记录，关闭提示窗口 |
| confirmCancel(\_ sender：Any) | UIButton | 取消删除操作，关闭提示窗口 |
| deleteRecord(\_ sender：Any) | UIButton | 弹出提示用户确定删除的窗口 |
| editRecord(\_ sender：Any) | UIButton | 跳转至修改记录窗口 |
| hideAlterView (\_ sender：Any) | UIButton | 隐藏提醒窗口 |

### 图表（ChartView）

#### 输出口

表格 11 图表界面输出口表

|  |  |  |
| --- | --- | --- |
| IBOutlet | 类型 | 说明 |
| rctypeBtn | UIButton | 呈现选择支出或收入的窗口 |
| yearLabel | UILabel | 显示年份 |
| monthLabel | UIButton | 选择日期并显示月份 |
| totoalValueLabel | UILabel | 当月总支出 |
| avgValueLabel | UILabel | 月平均支出 |
| dateView | UIView | 选择日期窗口 |
| datePicker | UIDatePicker | 日期选择器 |
| setRcTypeView | UIView | 选择支出或收入窗口 |
| lineCharView | LineCharView | 折线图窗口 |
| pieChartView | PieChartView | 饼图窗口 |
| chooseCostBtn | UIButton | 选择支出 |
| chooseIncomeBtn | UIButton | 选择收入 |

#### 操作

表格 12 图表界面操作表

|  |  |  |
| --- | --- | --- |
| IBAction | 连接控件类型 | 说明 |
| confirm(\_ sender：Any) | UIButton | 选择日期并显示选择的年份，月份，更新图表师徒 |
| concel(\_ sender：Any) | UIButton | 隐藏选择日期窗口 |
| setDate(\_ sender：Any) | UIButton | 显示选择日期窗口 |
| setRcType (\_ sender：Any) | UIButton | 显示选择收入/支出窗口 |
| chooseCost(\_ sender：Any) | UIButton | 更改标题为“支出”更新图表数据为支出数据 |
| chooseIncome(\_ sender：Any) | UIButton | 更改标题为“收入”更新图表数据为收入数据 |

### 记账（addRecordView）

#### 输出口

|  |  |  |
| --- | --- | --- |
| IBOutlet | 类型 | 说明 |
| inputRecordType | UITextField | 输入记录类型收入或支出 |
| inputType | UITextField | 输入收入/支出类型 |
| inputValue | UITextField | 输入金额 |
| inputDesc | UITextField | 输入备注 |
| inputDate | UITextField | 输入记录日期 |
| datePicker | UIDatePicker | 日期选择器 |
| dateView | UIView | 日期选择窗口 |
| incomeBtn | UIButton | 选择收入 |
| costBtn | UIButton | 选择支出 |

表格 13 记账界面输出口表

#### 操作

表格 14 记账界面操作表

|  |  |  |
| --- | --- | --- |
| IBAction | 连接控件类型 | 说明 |
| saveRecord(\_ sender：Any) | UIButton | 保存记录 |
| confirmDate(\_ sender：Any) | UIButton | 选择日期 |
| setDate(\_ sender：Any) | UIButton | 显示日期选择窗口 |
| cancelConfirm(\_ sender：Any) | UIButton | 隐藏日期选择窗口 |
| setRecordType(\_ sender：Any) | UIButton | 设置记录类型为支出 |
| alsoSetRecordType(\_ sender：Any) | UIButton | 设置记录类型为收入 |
| viewCilck(\_ sender：Any) | UIControl | 隐藏键盘 |
| showTextView(\_ sender：Any) | UIButton | 防止键盘遮挡输入框 |

### 工具（toolsView）

#### 输出口

表格 15 工具界面输出口表

|  |  |  |
| --- | --- | --- |
| IBOutlet | 类型 | 说明 |
| wbView | UIWebView | 显示网页 |

#### 操作

表格 16工具界面操作表

|  |  |  |
| --- | --- | --- |
| IBAction | 连接控件类型 | 说明 |
| sechand(\_ sender：Any) | UIButton | 显示58同城二手市场 |
| express(\_ sender：Any) | UIButton | 查询快递100 |
| exchange\_rate(\_ sender：Any) | UIButton | 查询汇率 |

### 我的（meView）

#### 输出口

表格 17 我的界面输出口表

|  |  |  |
| --- | --- | --- |
| IBOutlet | 类型 | 说明 |
| loginBtn | UIBtton | 跳转到登录窗口 |
| userLabel | UILabel | 显示用户名 |

#### 操作

表格 18 我的界面操作表

|  |  |  |
| --- | --- | --- |
| IBAction | 连接控件类型 | 说明 |
| set(\_ sender：Any) | UIButton | 跳转到登录窗口 |

### 登录（loginView）

#### 输出口

表格 19 登录界面输出口表

|  |  |  |
| --- | --- | --- |
| IBOutlet | 类型 | 说明 |
| loginView | UIView | 登录窗口 |
| signView | UIView | 注册窗口 |
| inputUserName | UITextField | 登录窗口输入用户名 |
| inPutPwd | UITextField | 登录窗口输入密码 |
| inputUsername | UITextField | 注册窗口输入用户名 |
| inputPwd | UITextField | 注册窗口输入密码 |
| inputConfirmPwd | UITextField | 注册窗口输入确认密码 |

#### 操作

表格 20 登录界面操作表

|  |  |  |
| --- | --- | --- |
| IBAction | 连接控件类型 | 说明 |
| login(\_ sender：Any) | UIButton | 验证用户，登录，跳转窗口 |
| return(\_ sender：Any) | UIButton | 返回“我的”窗口 |
| signUp(\_ sender：Any) | UIButton | 显示注册窗口 |
| isSignUp(\_ sender：Any) | UIButton | 注册用户，显示登录窗口 |
| hide1(\_ sender: Any) | UIControl | 隐藏键盘 |
| hide2(\_ sender: Any) | UIControl | 隐藏键盘 |

### 编辑（editView）

#### 输出口

表格 21 编辑界面输出口表

|  |  |  |
| --- | --- | --- |
| IBOutlet | 类型 | 说明 |
| inputRecordType | UITextField | 输入记录类型收入或支出 |
| inputType | UITextField | 输入收入/支出类型 |
| inputValue | UITextField | 输入金额 |
| inputDesc | UITextField | 输入备注 |
| inputDate | UITextField | 输入记录日期 |
| datePicker | UIDatePicker | 日期选择器 |
| dateView | UIView | 日期选择窗口 |
| incomeBtn | UIButton | 选择收入 |
| costBtn | UIButton | 选择支出 |

#### 操作

表格 22 编辑界面操作表

|  |  |  |
| --- | --- | --- |
| IBAction | 连接控件类型 | 说明 |
| saveRecord(\_ sender：Any) | UIButton | 保存记录，跳转回“明细”窗口 |
| confirmDate(\_ sender：Any) | UIButton | 选择日期 |
| setDate(\_ sender：Any) | UIButton | 显示日期选择窗口 |
| cancelConfirm(\_ sender：Any) | UIButton | 隐藏日期选择窗口 |
| setRecordType(\_ sender：Any) | UIButton | 设置记录类型为支出 |
| alsoSetRecordType(\_ sender：Any) | UIButton | 设置记录类型为收入 |
| viewCilck(\_ sender：Any) | UIControl | 隐藏键盘 |
| showTextView(\_ sender：Any) | UIButton | 防止键盘遮挡输入框 |
| back(\_ sender：Any) | UIButton | 返回“明细”窗口 |
| deleteRecord(\_ sender：Any) | UIButton | 弹出对话框，是否删除记录，并跳转 |

## 界面跳转设计

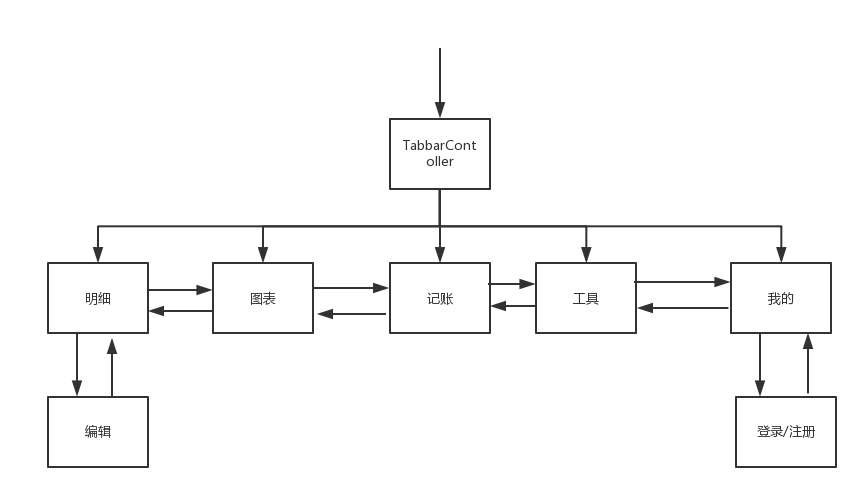


图 2 界面之间跳转关系

**说明**

* 明细、图表、记账、工具、我的，五个界面之间通过TabbarController进行跳转，且不存在参数的传递。
* 明细和编辑界面之间通过segue实现跳转，从明细界面传递选中的记录信息到编辑中，由编辑中的dataRecord对象record接收，并填充编辑界面中的各输出口，供用户修改记录时进行参考
* 我的和登录/注册界面之间同样通过segue实现跳转，当登录后，会返回用户名，由我的界面中的username字符串变量接收，并赋值给显示用户名的Label。

# 应用实现

## DataBaseController

dataBaseController类运用GitHub上的SQLite开源项目，封装了对SQLite数据库的访问，创建表格，以及对数据的增、删、改、查的操作。

表格 23 dataBaseController 方法功能说明表

|  |  |
| --- | --- |
| 关键方法 | 实现功能 |
| initDateBase() | 链接数据库，若表不存在，创建表 |
| insertData(record\_type:String="",type:String="",value:Double=0.0,year:String="",month:String="",day:String="",desc:String="") | 插入数据（插入记录） |
| insertData(username:String,password:String) | 插入数据（插入用户名、密码） |
| queryData(month:String,year:String)->[[String:Any]] | 查询数据，重载函数，符合各种查询需要 |
| queryData(record\_type:String,year:String,month:String,groupby:String)->[[String:Any]] |
| queryData(username:String)->[[String:Any]] |
| updateData(id:Int,record\_type:String,type:String,value:Double,year:String,month:String,day:String,desc:String) | 更新记录号为id的记录 |
| deleteData(rid:Int) | 删除记录号为rid的记录 |

## detailsViewController

detailsViewController主要实现根据年月份查询账单记录，并计算每一天的支出和记录总和，并实现长按弹出菜单，选择修改或删除记录。detailsViewController继承UIController的同时，实现了UICollectionViewDelegate、UICollectionDataSource协议，实现了根据数据内容动态创建表单试图的功能

表格 24 detailsViewController 方法功能说明表

|  |  |
| --- | --- |
| 关键方法 | 实现功能 |
| reload() | 加载数据，并计算每天的支出和收入总金额 |
| prepare(for segue: UIStoryboardSegue, sender: Any?) | 界面跳转，并将当前选中记录已dataRecord对象的形式传至子视图 |
| longPressd(gesture: UILongPressGestureRecognizer) | 长按事件响应方法，在选中记录上长按弹出编辑菜单 |
| headerRefresh() | 实现下拉刷新数据功能 |
| numberOfSections(in collectionView: UICollectionView) -> Int | UICollectionViewDelegate协议方法，返回collectionView中的分区数 |
| ollectionView(\_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int | UICollectionViewDelegate协议方法，返回collectionView中的每个分区中的单元格数量 |
| collectionView(\_ collectionView: UICollectionView,  viewForSupplementaryElementOfKind kind: String, at indexPath: IndexPath) -> UICollectionReusableView | UICollectionViewDelegate协议方法，编辑每个分区的头部和尾部，在头部显示记录日期，以及当天的总支出，总收入 |
| collectionView(\_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell | UICollectionViewDelegate协议方法，动态创建单元格 |

## chartViewController

chartViewController使用了GitHub的开源项目Charts-ios，实现了对消费收入记录的数据统计和处理，并以图表的形式展现出来。

表格 25 chartViewController 方法功能说明表

|  |  |
| --- | --- |
| 关键方法 | 实现功能 |
| loadChartData(condition:String) | 读取数据库，加载图表数据，x轴数据和y轴数据 |
| setLineChart(dataPoints: [String], values: [Double]) | 设置折线图数据 |
| setPieChart(dataPoints: [String], values: [Double]) | 设置饼图数据 |
| setLineChartView() | 加载折线图数据，设置折线图样式 |
| setPieCharView() | 加载饼图数据，设置饼图样式 |

## addRecordViewController

addRecordViewController主要实现账单记录的添加功能，并解决了弹出键盘遮挡输入框的问题。

表格 26 addRecordViewController 方法功能说明表

|  |  |
| --- | --- |
| 关键方法 | 实现功能 |
| showTextView(\_ sender: Any) | 当键盘遮挡输入框时，将整个界面上移200个像素 |
| saveRecord(\_ sender: Any) | 调用通过dbHelper,调用dataBaseController的插入数据库方法 |

## toolsViewController

toolsViewController主要通过按钮响应动作设置WebView的网络请求URLRequest，从而使web View可以浏览不同的网页.

## meViewController

meViewController主要实现用户的管理功能，反应应用的登录状态

表格 27 meViewController 方法功能说明表

|  |  |
| --- | --- |
| 关键方法 | 实现功能 |
| close(segue:UIStoryboardSegue) | 接收登陆界面传回来的用户名，显示登录状态 |

## loginViewController

loginViewController实现了用户的登录和注册功能

表格 28 loginVIewController 方法功能说明表

|  |  |
| --- | --- |
| 关键方法 | 实现功能 |
| loginUser() | 链接数据库，查询用户数据，匹配用户名和密码 |
| signUpUser() | 链接数据库，写入用户数据 |

## editViewController

editViewController实现对选中记录的修改和删除功能

表格 29 editViewController 方法功能说明表

|  |  |
| --- | --- |
| 关键方法 | 实现功能 |
| saveRecord(\_ sender: Any) | 通过dataBaseController的对象dbHelper调用更新数据库方法，并跳转回“明细”界面 |
| deleteRecord(\_ sender: Any) | 弹出提示框，通过dataBaseController的对象dbHelper调用删除记录方法，删除相应记录，并跳转回“明细”界面 |

# 应用截图

## 明细界面

|  |  |
| --- | --- |
| Simulator Screen Shot 2017年10月28日 下午4  图 3明细窗口界面 | Simulator Screen Shot 2017年10月28日 下午4  图 4 按月份查询账单 |
| Simulator Screen Shot 2017年10月28日 下午4  图 5仿对话框弹出菜单 | Simulator Screen Shot 2017年10月28日 下午4  图 6 仿提示框 |

## 图表界面

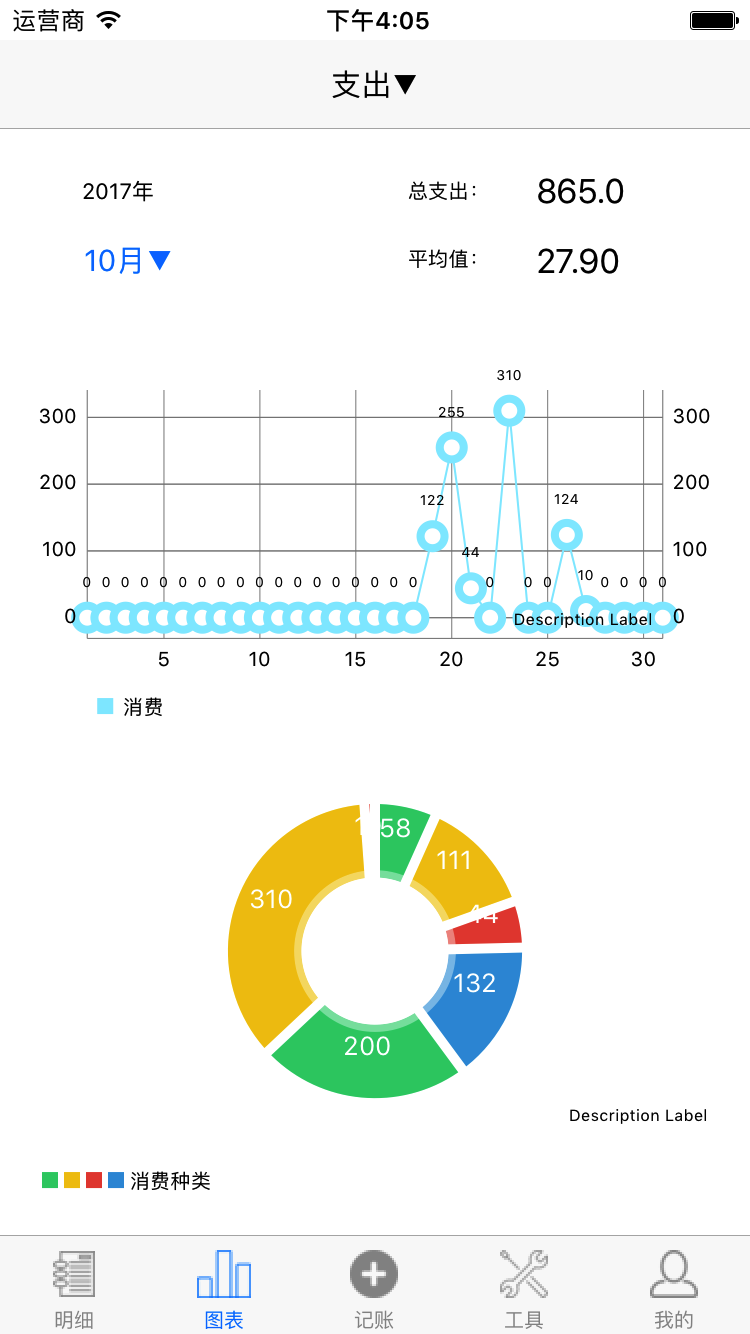
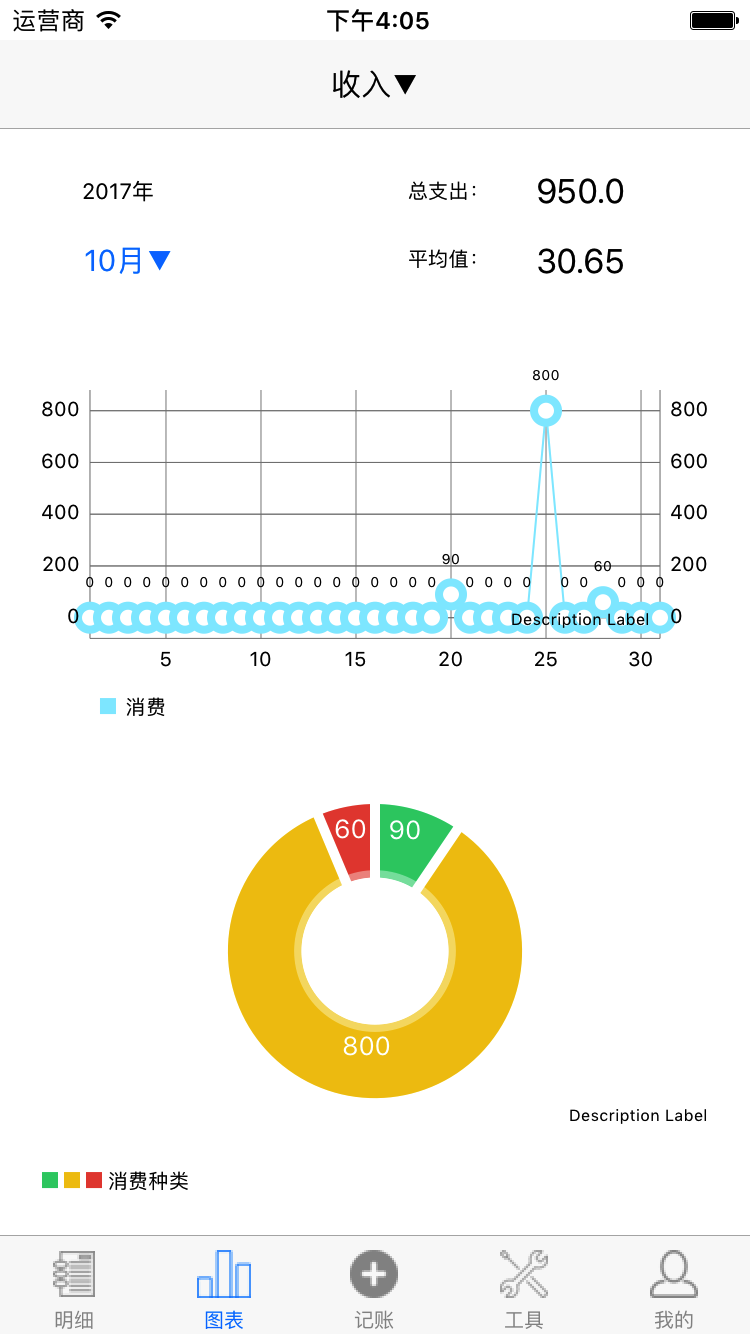


图 7 支出情况统计

图 8 支出情况统计

## 记账界面



图 9 记账界面

图 10 防止键盘遮挡

## C:\Users\yangh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Simulator Screen Shot 2017年10月28日 下午4.09.00.pngC:\Users\yangh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Simulator Screen Shot 2017年10月28日 下午4.08.52.png工具界面

图 11 查询汇率

图 12 查询快递

## 我的界面



图 13 我的界面

## 登录和注册界面



图 14 注册界面

图 15 登录界面

## 编辑界面



图 16 编辑界面

图 17 提示窗口

# 对项目中使用的开源项目说明

## MJRefresh

MJRefresh使用Objective-C编写，使用时需要配置头文件，将MJRefresh包含到项目中，MJRefresh可以使collectionView和TableView实现下拉刷新或上滑刷新的操作，并由文字提示和动画显示。

## SQLiteDB

SQLite是专为移动设备打造的一款轻量级数据库，在Android和iOS中都有广泛的应用，SQLiteDB开源项目使用swift3.0编写，封装了关于SQLite操作的相关方法，极大方便了开发人员对SQLite的操作。只需声明一个SQLiteDB对象，通过SQLiteDB.share方法，即可获取数据库实例db，通过db.open()方法打开数据库。

通过db.excute()方法，输入相应的SQL语句，即可对数据库进行查询、更新、删除等操作。

通过db.query（）方法，输入相应的SQL语句，即可对数据库进行查询，该方法将返回一个dictionary数组，将数据库中的列名作为键，将数据作为值存储在该字典数组中，开发人员只需设置变量接受查询结果，即可读取到内部的查询结果。

另外，该开源项目也支持将数据库中的表、元组等映射成对象，通过swift代码，操作这些对象，即可完成数据库的创建和增删改查等操作。

## Charts-iOS

Charts-iOS是从Android移植过来的图表库，提供了丰富的图表试图，比如折现图、柱状图、饼图等，现在已经更新到支持Xcode9 和Swift4。使用时，需要将该图表库加入项目中，并import Chart，在StoryBoard中这只某个View空间为XXXCharView的子类，即可在运行时实例化Char View对象，配合相应方法，将数据以图表的形式显示。功能强大。

# 开发过程中遇到的问题和解决方案

## 明细界面在调试多次后出现崩溃无法显示的情况

解决办法：最初只设置collection View的约束，后来发现是XCode本身的问题，因此，出现崩溃后，重新设置约束，清理并重建项目即可解决。

## CollectionView的scroll View全屏滚动遮挡其他试图

解决办法：storyboard中有类似PS的图层机制，调整控件的上下顺序即在调整层级关系，自上而下，显示优先级越来越高，因此，只需将collection View放置在最上层，则可解决遮挡问题。

## 数据库显示加密或不存在，无法链接数据库

解决办法：再对比自己的项目设置和样例的项目设置后，仍找不到原因，因此，把我的项目文件全部导入样例的项目中后，解决问题。

## 计算每天收入、支出总额时无法正确读取数据

解决方案：优化代码，修改逻辑，并修改dataBaseController中的查询方法。

## 从“明细”页面到“编辑”页面无法通过提醒框跳转问题

解决方案：根据storyboard各界面的层级关系，新建一个Window对象，设置一个作为背景的View，颜色设置成黑色，透明图调制0.6，并将该View添加到window中。在创建一个View对象，根据需要放置按钮，与后台以Action连接，将该界面隐藏。在长按选中的记录后，弹出window以及其上的对象并显示隐藏的view，即可实现swift的模态效果，模拟对话框，并实现跳转功能。



图 18 模拟模态

## 当在弹出菜单点击“删除”键后，再次长按弹出菜单偏移至左上角的问题

解决方案：经过反复试验，发现swift的模态退出后貌似会重置window层中的某些对象，具体原因不得而知。解决方法和之前模拟对话框一样，新建一个view，当点击弹出菜单的“删除”后出现，模拟模态效果，成功解决问题



图 19 真正模态

图 20 模拟模态

上面两张图片左边为模拟模态，以view和window组合。右边为真正的提示对话框。

## 场景跳转及传值问题

解决方案：虽然用navigationController也可以实现页面跳转和传值，但个人感觉操作比较麻烦，所以选择了比较简单的view到view之间通过segue跳转和传值的方法。

# 总结

本款app最核心的功能依赖于对数据库的操作，因为数据库的支持，才可以做到永久存储数据读取数据，实现应用的核心功能。

项目开发周期大致在10天左右，前期主要是构思整个应用的层次结构以及学习相应的swift语法知识以及开源项目的使用方法。由于苹果的开发语言更新太过频繁，导致无论是网上或是书上可以参考的资料十分不足，所以在开发过程中问题频出，为了解决一个问题，可能需要在网上查找好久才能找到可以参考的解决方案。

就swift语言本身而言，确实有着其他语言没有的优势，比如可以添加表情作为变量，可以随意将变量连接到字符串中等等，但仍有很多很不方便的地方，比如字符串取子串，操作就显得十分麻烦。另外，swift本身的命名规则值得学习，让人能够单凭字面意思就可以理解一个变量或常亮或函数的功能作用，以及函数中参数的意义。

XCode作为开发苹果系列操作系统的唯一开发工具，与其他编程IDE相比，还有很多不成熟和不稳定的地方。在开发项目时经常遇到修改正确但是错误提示仍然存在的情况并且不让编译。只有重新启动XCode后才可以生效。

XCode Interface Builder 虽然提供了很多实用的类库，控件库，且通过storyboard搭建应用的UI界面确实很方便，但控件的使用、创建以及事件响应机制上却略感不变，控件种类也相对较少不能满足某些特定的需要，比如很常用的下拉列表，取而代之的事IB中的PickerView，但是，要实现PickerView却必须在UIViewController中实现PickerViewDataSource和PickerVIewDelegate协议，相似的还有TableView和CollectionView，后两个控件虽然有相应得Controller可以直接用，但是如果想在普通的ViewController中使用就必须采用和实现PickerView相同的方法，略显繁琐，且XCode并不提示协议中封装的方法，对于初学者来说，不是很友好。

不过遇到问题，解决问题其实是对自己能力最好的锻炼。通过本次实践，掌握了ios开发的基础知识，可以更好的学习嵌入式开发，也认识到紧跟时代潮流，不断学习新知识的重要性。