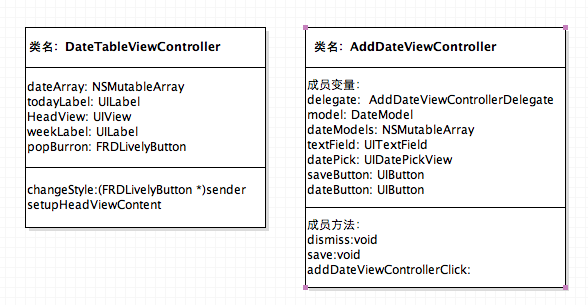
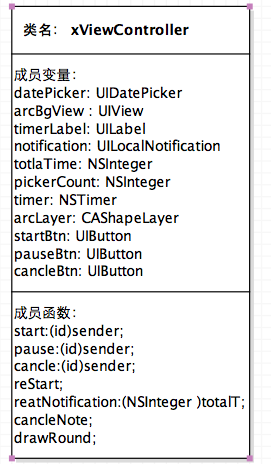
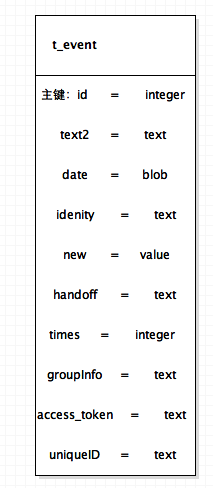
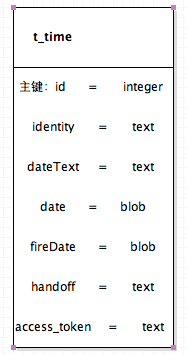
* **提醒精灵APP开发**
* 开发团队：何林狄，闻人超杰
* 组长：何林狄
* 组员：闻人超杰
* 目录
* 提醒精灵APP开发…………………………………………1
* 1 概述………………………………………………………3
* 1.1开发背景……………………………………………3
* 1.2 开发目标……………………………………………3
* 1.3 参考资料……………………………………………3
* 2 需求分析…………………………………………………3
* 2.1 需求分析……………………………………………3
* 2.2 操作用例……………………………………………3
* 2.3 功能分析划分………………………………………3
* 2.3.1 用户登录………………………………………3
* 2.3.2 用户管理………………………………………3
* 2.4 运行环境……………………………………………3
* 3 总体设计…………………………………………………3
* 3.1 系统建模……………………………………………3
* 3.1.1 层次方框图……………………………………3
* 3.1.2 类图设计………………………………………3
* 3.2 接口设计……………………………………………3
* 3.2.1 内部接口设计…………………………………3
* 3.2.2 登录界面设计…………………………………3
* 3.2.3 用户管理界面设计……………………………3
* 3.3 数据库结构设计……………………………………3
* 3.3.1 数据库E-R图…………………………………3
* 3.3.2 数据库逻辑设计………………………………3
* 4 详细设计…………………………………………………3
* 4.1 程序流程图…………………………………………3
* 5 实现测试…………………………………………………3
* 5.1 编码…………………………………………………3
* 5.1.1 代码约定………………………………………3
* 5.1.2 代码编写原则…………………………………3
* 5.2 测试要点……………………………………………3
* 5.2.1 登录测试要点…………………………………3
* 5.2.2 主界面测试要点………………………………3
* 5.3 测试结果和总结……………………………………3
* **一、概述**
* **1.1开发背景**
* 在这个移动互联网迅速发展，移动终端设备如iPhone， Android，,,Windows Phone 智能手机的不断普及的背景下，使得日常生活更获取信息更加便利，迅速，但同时也充斥的更多一些冗杂，垃圾的信息，使精力更易分散，导致一些更加重要的事情被拖延，甚至被遗漏，一定程度的干扰到了现代人的正常生活与工作。我们团队因此决定开发一款关于集事务，纪念日，计时于一体用于改善用户生活质量，提高工作效率的APP《提醒精灵》。
* **1.2开发目标**
* 对于《提醒精灵》APP，我们定位为一款具有简洁清爽UI界面，便利的操作的手机端应用，其功能设定分为 自定义事务的到时提醒，特殊纪念日的提醒，以及倒计时提醒这三个模块，并且拥有简单用户登录管理功能，以及事务数据的好友间分享，和应用数据的持久，利用SQLite3数据库对应用数据提供增，删，查，改的操作。
* **1.3参考资料**
* 提醒精灵涉及的开源项目：
* FMDB ：IOS平台的SQLite数据库的第三方框架。
* 下载地址：<https://github.com/ccgus/fmdb>
* AF Networking ：用于IOS以及Mac OS X的一个使用起来方便快捷的网络通信框架。
* 下载地址：<https://github.com/AFNetworking/AFNetworking>
* MB Progress HUD：用于网络登陆加载或场景切换，信息提示的提示框第三方库
* 下载地址：<https://github.com/jdg/MBProgressHUD>
* 参考书籍：
* 《IOS网络编程与云端应用》 关东升著
* 《IOS图形图像、动画和多媒体编程技术最佳时间》关东升著
* 《iPhone与IPAD开发实战》 董玉萍,关东升著
* **二、需求分析**
* **2.1需求陈述**
* 在事务提醒的模块上，通过在输入框内输入需要提醒的事件，选择提醒的时间点，然后选择提醒的频率，是单次，每天，还是每星期，还有 播放时的铃声，最后再进行工作，学习，生活上的分组，一切设定好后会创建一个提醒事项在该模块窗口上，并且有个开关供用户选择该事件是否进入提醒队列，除此之外，对单个事项视图点击后进入编辑视图，可修改文本内容，时间，播放的铃声，以及提醒的频率，并且通过分享按钮，邮件或者短信方式发送给好友起到通知他人的作用。如果有多个提醒事件，都将按时间先后顺序排列，到时间后，系统会发送通知给用户，用户点击响应该事件。
* 纪念日模块上，中央视图展现当天的年月日以及星期，提醒用户当前日子，以及一个加号按钮，用来添加纪念日数据，点击后进入添加视图，输入关键文本信息，以及日子，保存后回到纪念日主视图，在下方出现一个纪念日事件视图，显示醒目的文本信息，距离设定时间的准确天数，并在日子来到的前一天晚上21：30提醒用户。创建多个后逐次向下添加，手指向左滑动屏幕可删除所选的纪念日事件。
* 倒计时模块上，设定需要计时的时间片，点击开始按钮，就开始显示剩余时间，中央视图的圆形并且随着剩余时间的减少更变化，在程序没有完全退出的状态下，程序一直计时，直至视图显示的剩余时间为零，弹出提示框，告知用户所设定的时间片一到，在中间过程可以点击暂停按钮，或者结束按钮操作终止计时。
* **2.2 操作用例**
* 首次打开《提醒精灵》，进入用户登录，可以选择新浪微博账号，人人网账号，或者本地匿名登录进入应用的功能主界面，在事项提醒界面上，点击添加提醒横条，用键盘输入提醒事件的文本内容确认，弹出时间选择器，选择时间点，按时钟小图标确定，再弹出关于提醒频率和铃声的选择器，选择后按音乐小图标确定，进入分组选择器，结束后，在弹出的确认用户确定添加的表单，按确定后展示所设定的提醒事项的全部详细在展示视图上，有按钮开关，计划时间点，文本内容，以提醒的频率；按底部工具栏，转到纪念日提醒视图，点击加号按钮，弹出添加视图，填入Text内容，在时间选择器上选择需要纪念的年，月，日的，按保存回到原来视图，纪念日数据展现在表视图上，并且简洁明显地提示用户距离指定时间的准确天数，还有填入的文本信息，对于表视图上指定的一条纪念日信息，手指向左滑动，出现删除按钮，可进行删除操作，删除后提醒队列删除该提醒事件。
* 再按底部工具栏，进入倒计时提醒视图，选择好要计时的时间，按开始计时，计时时间一到弹出提醒窗口提示用户，在时间到前可按暂停按钮，延时计时，或者按结束按钮，重置倒计时回到原来界面，原来提醒的通知取消，用户可以重新设定。
* **2.3 功能分析划分**
* **2.3.1 用户登录**
* 1)本地登录：应用数据信息都为同一个本地匿名用户所有，事项提醒，纪念日，倒计时的数据都以匿名用户为标示符存入数据库中，读取。
* 2)网络用户登录：新浪用户可以通过登录微博账户，进入应用主功能界面，并且全部的数据信息都已当前所登录的用户保留，退出登录后用户数据被保护而不再显示，直至用户再次以相同账号登录后读取，人人网用户也是对应用数据采用相同的形式保留。
* 2.3.2 用户管理
* 登录的用户，能对该用户的事项提醒数据进行增，删，查，改操作，对纪念日数据进行添加，删除操作，对倒计时提醒进行能暂停，重置操作，并且在设置视图中按退出按钮后注销登录，返回用户登录界面。
* **2.4 运行环境**
* Mac OSX 10.8.8以上，Xcode5.1.1的iPhone5s模拟器的IOS7.1系统上.
* **三、总体设计**
* **3.1 系统建模**
* 3.1.1 层次方框图
* 3.1.2 类图设计
* 三个主要功能模块的类图：
* 提醒事项视图控制器类
* 
* 纪念日视图控制器类
* 
* 倒计时视图控制器类
* 
* **3.2 接口设计**
* 3.2.1内部接口设计
* 提醒事项视图控制器类
* 表视图控制器协议接口:
* 1.UITableViewDelegate, 2.UITableViewDataSource
* 选择器控件协议接口：
* 3.UIPickViewDelegate, 4.UIPickViewDataSource
* 自定义Cell协议接口：
* 5.remainCellDelegate
* 手势识别协议接口：
* 6. UIGestureRecognizerDelegate1
* 警告框协议接口：
* 7.UIActionSheetDelegate 8.UIAlertViewDelegate
* 邮件视图控制器接口：
* 9.MFMailComposeViewControllerDelegate
* 短信视图控制器接口：
* 10.MFMessageComposeViewControllerDelegate
* 文本输入框接口：
* 11.UITextViewDelegate
* 纪念日视图控制器类
* 自定义添加日期视图控制器协议接口：
* 1.AddDateViewControllerDelegate
* 文本输入框协议接口:
* 2.UITextFieldDelegate
* 表视图控制器协议接口:
* 3.UITableViewDelegate, 4.UITableViewDataSource
* 倒计时视图控制器类
* 警告框协议接口：
* 1.UIAlertViewDelegate
* 设置视图控制器类
* 表视图控制器协议接口:
* 1.UITableViewDelegate, 2.UITableViewDataSource
* 动作表单协议接口：
* 3.UIActionSheetDelegate
* 邮件组件控制器协议接口：
* 4.MFMailComposeViewControllerDelegate
* 3.2.2登录界面设计
* **3.3 数据库结构设计**
* 3.3.1 数据库E-R图
* 事项提醒的数据表ER图 纪念日的数据表ER图
* ** **
* 3.3.2 数据库逻辑设计
* 1.事项提醒的数据库表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * 序号 | * 字段名 | * 类型 | * 长度 | * 默认值 | * 允许空 | * 主键 |
| * 1 | * Id | * Integer | * 10 | * 1 | * Null | * Yes |
| * 2 | * Text2 | * Text | * 100 | * Null | * Null | * No |
| * 3 | * Music | * Text | * 20 | * 7.caf | * Null | * No |
| * 4 | * Date | * Blob | * 50 | * Null | * Null | * No |
| * 5 | * Identity | * Text | * 20 | * Null | * Null | * No |
| * 6 | * New | * Text | * 10 | * Null | * Null | * No |
| * 7 | * Hand Off | * Text | * 10 | * Null | * Null | * No |
| * 8 | * Times | * Text | * 5 | * Null | * Null | * No |
| * 9 | * Group Info | * Text | * 10 | * Null | * Null | * No |
| * 10 | * Access token | * Text | * 50 | * Anyone | * Null | * No |
| * 11 | * Unique ID | * Text | * 20 | * Null | * Null | * No |

* 2.纪念日提醒的数据库表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * 序号 | * 字段名 | * 类型 | * 长度 | * 默认值 | * 允许空 | * 主键 |
| * 1 | * Id | * Integer | * 10 | * 1 | * Null | * Yes |
| * 2 | * Identity | * Text | * 20 | * Null | * Null | * No |
| * 3 | * Date Text | * Text | * 40 | * Null | * Null | * No |
| * 4 | * Date | * Blob | * 50 | * Null | * Null | * No |
| * 5 | * Fire Date | * Blob | * 50 | * Null | * Null | * No |
| * 6 | * Access token | * Text | * 50 | * Anyone | * Null | * No |

* **四、详细设计**
* **4.1登陆功能具体设计**
* **4.2事务提醒具体设计**
* **4.3 纪念日提醒具体设计**
* **4.4倒计时提醒具体设计**
* （开始）单击开始按钮，根据选择控件的数据设定倒计时时间，并且调用创建通知 方法设定本地通知。然后开启定时器执行回调方法Time Fire Method刷新标签显示时间和进度条进度。
* （暂停）通过是否暂停标识和定时器是否初始化进行相应相应判断当前的状态，并做出相应的动作。
* （停止）调用暂停方法，取消通知，停止定时器刷新，去除动画效果。
* **4.5数据展示设计**
* （数据加载）：在应用暂停或重新进入时，数据必须重新加载。我们在每个视图控制器的view Did Load方法中调用view Did Load State方法和数据懒加载方法进行数据初始化。
* （添加） 在Head View中填写事务相关信息并设定相应的提醒形式后将数据存入数组，再将数组转化为一个模型存入模型数组中，方便数据传递，与此同时通过Model注册并创建通知。数据添加完成后让表视图调用重载方法根据数组模型刷新列表。
* （修改）点击cell后弹出一个警告窗，我们在这里自定义了一个nib文件和对应的类，提供了修改提醒时间、提醒音乐和文本的接口。
* （保存）在事项被创建或修改完成后将数据保存到数组对象中，并将数组转换为事项数据模型数组替换原模型数组中对对应的数组。
* **4.6数据库实现**
* 新建数据库类管理文件工具类Event Date Tool和Model Date Tool，传入模型数据，利用第三方数据库框架FMDB提供数据库管理接口，把用户信息，事项提醒数据，纪念日数据写入SQLite表，实现数据本地化持久化。
* **五、编码**
* **5.1 详细代码请访问GITHUB**
* 手机提醒精灵源代码：
* https://github.com/MobileSprite/project
* **5.2 测试要点**
* **主要讲求**系统功能完整性、系统界面合理性、体系构架合理性、交互合理性
* **5.2.1 登录测试要点**
* 分别登录界面的各个按钮，并输入对应的平台的账号及密码，看能否正常登录。
* **5.2.2 主界面测试要点**
* 输入提醒内容，并设置相应的提醒类型。在指定时间是否触发提醒机制

5.3 测试结果和总结