



安卓端时光笔记APP

总体设计

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态： | 文件标识： | | SE-2018春-G01 |
| [ ] 草稿 | 当前版本： | | 0.4.0 |
| [ ] 正式发布 | 作 | 者： | 郦哲聪、陈铉文、刘晓倩 |
| [√] 正在修改 | 完成日期： | | 2018-06-16 |

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 参与者 | 起止日期 | 备注 |
| 0.1.0 | 郦哲聪、陈铉文、刘晓倩 | 2018-05-04至2018-05-07 | 总体设计初稿 |
| 0.2.0 | 郦哲聪、陈铉文、刘晓倩 | 2018-05-13至2018-05-16 | 按用户要求写改了logo、修改了HIPO图 |
| 0.3.0 | 郦哲聪、陈铉文、刘晓倩 | 2018-05-23至2018-06-03 | 进行内容的详细补充 |
| 0.4.0 | 郦哲聪、陈铉文、刘晓倩 | 2018-06-03至2018-06-16 | 文档随项目进度进一步增减 |

目录

[1．引言 5](#_Toc517212123)

[1.1编写目的 5](#_Toc517212124)

[1.2项目背景 5](#_Toc517212125)

[1.2.1 项目名称 5](#_Toc517212126)

[1.2.2 项目的提出者 5](#_Toc517212127)

[1.2.3 项目主动承担小组 5](#_Toc517212128)

[1.2.4 项目的用户 5](#_Toc517212129)

[1.2.5 项目与其他软件、项目的关系 5](#_Toc517212130)

[1.3定义 6](#_Toc517212131)

[1.4参考资料 6](#_Toc517212132)

[2．任务概述 6](#_Toc517212133)

[2.1目标 6](#_Toc517212134)

[2.2运行环境 6](#_Toc517212135)

[2.2.1设备 6](#_Toc517212136)

[2.2.2支持软件 6](#_Toc517212137)

[2.2.3条件与限制 7](#_Toc517212138)

[2.3需求概述 7](#_Toc517212139)

[2.3.1系统功能 7](#_Toc517212140)

[2.3.2系统性能 8](#_Toc517212141)

[2.3.3输入输出要求 8](#_Toc517212142)

[2.3.4数据管理能力要求 8](#_Toc517212143)

[2.3.5故障处理要求 9](#_Toc517212144)

[2.3.6其他专门要求 9](#_Toc517212145)

[2.4实现计划 9](#_Toc517212146)

[2.5选择方案 9](#_Toc517212147)

[3．总体设计 13](#_Toc517212148)

[3.1处理流程 13](#_Toc517212149)

[3.1.1 登录 13](#_Toc517212150)

[3.1.2事务管理 14](#_Toc517212151)

[3.1.3提醒业务 14](#_Toc517212152)

[3.2功能分配 15](#_Toc517212153)

[3.2.1用户管理模块 15](#_Toc517212154)

[3.2.2事务管理模块 16](#_Toc517212155)

[3.2.3提醒管理模块 16](#_Toc517212156)

[3.2.4清单管理模块 17](#_Toc517212157)

[3.2.5申诉管理模块 17](#_Toc517212158)

[3.2.6日历视图模块 18](#_Toc517212159)

[3.3总体结构和模块外部设计 18](#_Toc517212160)

[4．接口设计 18](#_Toc517212161)

[4.1外部接口 18](#_Toc517212162)

[4.1.1硬件接口 18](#_Toc517212163)

[4.1.2软件接口 19](#_Toc517212164)

[4.2内部接口 19](#_Toc517212165)

[4.3用户接口 19](#_Toc517212166)

[5．数据结构设计 19](#_Toc517212167)

[5.1逻辑结构设计 19](#_Toc517212168)

[5.1.1表UserInformation 19](#_Toc517212169)

[5.1.2表EventInformation 20](#_Toc517212170)

[5.1.3表ListIformation 20](#_Toc517212171)

[5.1.4表DRemindInformation 21](#_Toc517212172)

[5.1.5表PomoTimer 21](#_Toc517212173)

[5.2物理结构设计 21](#_Toc517212174)

[5.3数据结构与程序的关系 21](#_Toc517212175)

[6．运行设计 22](#_Toc517212176)

[6.1运行模块的组合 22](#_Toc517212177)

[6.2运行控制 22](#_Toc517212178)

[6.3运行时间 22](#_Toc517212179)

[7．出错处理设计 23](#_Toc517212180)

[7.1出错输出信息 23](#_Toc517212181)

[7.1.1登录模块错误 23](#_Toc517212182)

[7.1.2注册模块错误 23](#_Toc517212183)

[7.1.3信息修改模块错误 23](#_Toc517212184)

[7.1.4事务添加模块错误 23](#_Toc517212185)

[7.2出错处理对策 24](#_Toc517212186)

[8.安全保密设计 24](#_Toc517212187)

[9.维护设计 24](#_Toc517212188)

# 1．引言

## 1.1编写目的

为了保证项目小组能按时保质的完成项目目标，更好的学习工程化思想，合理有序的开展项目工作，我们采用ISO9001标准编写文档，该设计说明书用于指导软件实施时应遵从的规范。通过规范开发人员的开发过程达到保证项目团队完成软件项目的基本要求，提高软件质量的目的。开发者应根据本文档进行软件开发和编制软件开发文档。本文档是对软件项目开发的基本要求。

## 1.2项目背景

### 1.2.1 项目名称

安卓端时间管理APP

### 1.2.2 项目的提出者

杨枨老师

### 1.2.3 项目主动承担小组

浙江大学城市学院SE2018春-G01小组

### 1.2.4 项目的用户

浙江大学城市学院在校学生

### 1.2.5 项目与其他软件、项目的关系

本项目采用C/S架构，客户端程序建立在Android 5.0及以上的以JAVA语言开发的应用程序，C#语言开发服务器，采用Sql Sever的数据库服务程序，租借云端服务器（云翼计划）。

## 1.3定义

## 1.4参考资料

书籍资料：

《软件工程原书第八版》机械工业出版社等 RogerS.Pressman Bruce R.Maxim著 2017年1月第一版 第294545号

文档资料：

《SE2018春-G01-项目计划》

《SE2018春-G01-测试计划（初稿）》

《SE2018春-G01-用户手册（初稿）》

《SE2018春-G01-需求分析报告（初稿）》

网络资料：

<http://www.docin.com/p-1322536948.html> 2018.5.5 12:13

<https://blog.csdn.net/dropeye/article/details/6510325> 2018.5.5 9:03

<https://www.liuchuo.net/archives/1848> 2018.5.5 9:13

# 2．任务概述

## 2.1目标

基本目标：市场上已有的一些有关于时间管理的APP，我们发现虽然手机市场上有关于时间管理的APP种类繁多，但是很多APP有关于界面设计、闹钟设置等问题存在很多缺陷，导致用户在使用软件时出现种种问题。设计一款功能较全面、界面简洁、操作便捷的时间管理APP并基本实现选定的功能。

终极目标：操作简洁方便，处理问题及时快速，确实能够帮助用户管理时间。

## 2.2运行环境

### 2.2.1设备

租借的云端服务器（阿里云云翼计划）

系统为Android 5.0及以上的智能手机

### 2.2.2支持软件

App运行环境：Android5.0

软件开发环境：

操作系统：Microsoft Windows 7、Microsoft Windows 10

开发软件：MySQL 、Android Studio

办公软件：Microsoft Office 2016，Microsoft project 2016

界面设计工具：Axure RP 8

负载测试工具：Loadrunner 12等

单元测试工具：Junit

配置管理工具：GitHub Desktop

硬件环境：三台电脑，以及一台服务器

### 2.2.3条件与限制

应具备的条件：三人协作的开发团队，具备基础的开发知识和经验，配置好的开发环境

已具备的条件：三人协作的开发团队，配置好的开发环境，符合要求的云端服务器，一年的域名使用权

尚需创造的条件：基础的开发知识和经验

## 2.3需求概述

### 2.3.1系统功能

1）用户登陆功能

用户能拥有自己的登陆账号，并能从云端获取属于自己的数据

允许提交遗忘密码的申请

2）更直观的日历视图

在日历中直观查看日程安排，修改任务时可以通过日历视图，更加便捷

3）强大的提醒功能

可以设置提醒的时间

可以根据设置的时间准时提醒或者提前提醒

可以选择提醒一次，也可以选择每次到特定时段提醒，让你再也不会错过到期时间

振铃、提示音等选择

4）分类功能

系统清单（包括所有、今天、垃圾桶）

个人清单（创建）主清单和子清单

清单管理（显示哪些清单）

同类型的清单可以整理到一个任务组中，一目了然。

5) 事务管理

事务创建时除了名字都可以按照默认来设置，默认的内容也可以自主设定

事务的修改有多种方式，包括时间修改（推迟一定时间、提到当前、从日历选择）、描述修改（子任务、备注说明）、完成状态（过期、完成、未完成）

事务的删除，删除的事务会放入垃圾桶，在垃圾桶内删除事务则彻底删除数据

事务的查询，可以通过日历视图查询，也可以按清单分类查询，也可以按完成情况查询

优先级设置，可以对任务标记优先级，分清事件缓急，帮助用户更好的规划日程

通知显示：在手机通知里显示

桌面挂件显示：按日期在手机桌面上列出事务和简单的添加

屏保中显示事务

### 2.3.2系统性能

服务对象为浙江大学城市学院全体学生（预计使用量和推广情况），约为3000-5000人

最大并发操作数量：1500人

存储空间：20G以上存储空间

#### 2.3.2.1精度

提醒时间误差在半分钟之内

#### 2.3.2.2时间特性要求

需要在3秒内响应。

#### 2.3.2.3可靠性

当断网时，用户端可以读取在手机中保存的信息来提供监听提醒。

#### 2.3.2.4灵活性

暂无，进一步的系统将提供多平台公用账号的机制。用户可以根据情况在不同的设备来管理自己的行程和获得提醒。

### 2.3.3输入输出要求

输入数据都有严格的规格要求，具体要求见《SE2018春-G01-测试计划》

错误信息输出应尽量以小弹框形式。

数据的显示应尽量采取分页的形式

### 2.3.4数据管理能力要求

当发生事务错误时要支持回滚操作。

当发生不可逆的数据库故障时，应备份有至少1天前的数据。

当发生数据库更新时用户网络故障时要支持事务回滚。

### 2.3.5故障处理要求

若遇到数据丢失等问题，可以使用定时备份和更新数据库的方法在数据库破坏时恢复数据库，同时有必要定时检查服务器，交换机等硬件设备以防临时出错，做到万无一失。

若遇到同时在线人数过多，导致系统卡顿、奔溃，或者消息更新滞后等故障，我们将在APP中设置一个问题反馈板块，实时收纳所有上报问题，小组三人及时进行维护。

### 2.3.6其他专门要求

暂无

## 2.4实现计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块号 | 包含内容（HIPO图中的编号） | 伪代码负责人 | 实现负责人 | 验收负责人 |
| 1 | 1.1 、1.2、2.1、3.1、4.1、5.1、6.1 | 刘晓倩 | 郦哲聪 | 陈铉文 |
| 2 | 1.3、2.4、3.2、4.4、5.2、6.2 | 郦哲聪 | 陈铉文 | 刘晓倩 |
| 3 | 2.2、2.3、4.2、4.3、5.3、6.3 | 陈铉文 | 刘晓倩 | 郦哲聪 |

软件设计划分为三大模块，分别由刘晓倩、郦哲聪、陈铉文进行设计、实现与验收。为保证小组成员都能深入理解软件设计，更好地合作开发时光笔记APP，我们选择轮流进行代码编写、实现和验收，帮助我们及时发现和修改错误。

## 2.5选择方案

Android 5.0端APP

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 5000个新的API，设备整体性、互联性更强 * 运行效率提高，优化了流畅度 | * 个人数据不能跨终端转移 * 碎片化问题严重 * 功能按键不完美 |
| 机会(Opportunities) | SO | WO |
| * UI用户界面创新设计 * 全新的省电模式 * 技术较为成熟 | -根据良好的界面进行APP设计  -加快更新产品功能 | -完善产品  -增加用户与开发人员的交流，及时满足消费需求 |
| 风险(Threats) | ST | WT |
| * 后台应用频繁重启 * 长时间未重启，系统应用性能下降，反应迟钝 | 及时对系统进行维护和管理，出现问题立即处理 | -提高技术，使用云端解决终端转移问题  -正确面对问题，注意规避风险 |

ios端app

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 能够得到ios庞大的用户资源 * 运行流畅 * 产权保护措施完善 | * 开发工具需要收费，作为学生难以负担 * 隐私保护措施严格 * 难以做出需要系统权限的特殊功能 |
| 机会(Opportunities) | SO | WO |
| * 能让更多的人群接触到我们的产品 * 可能获取更多的盈利 | -针对ios客户端进行优化，提升用户体验  -完善产品功能使用户满意 | -争取以盈利来填补支出  -针对ios无法获取用户信息的环境进行开发 |
| 风险(Threats) | ST | WT |
| * 陌生的领域 * 可能的开发失败 * 可能store审核不通过 | -对于未知的领域积极的进行学习  -了解ios Appstore的审核机制 | -进行风险评估分析  -多上网了解ios开发的过程以及注意点，规避风险 |

安卓系统内嵌

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 提供最佳的用户体验 * 与系统的兼容性好，界面与交互更加适应特定的系统 * 能访问硬件设备的底层功能，可访问本地资源 | * 移植到不同的平台上比较麻烦 * 维持多个版本成本较高 * 需要手机厂商的确认通过 |
| 机会(Opportunities) | SO | WO |
| * 可以让更多的用户体验到我们的产品 * 节约了广告宣传的成本 | -与多个平台合作开发，增加受众群体  -针对不同的平台进行优化 | -省略广告宣传步骤  -更多的投入到对于不同平台的优化上面 |
| 风险(Threats) | ST | WT |
| * 对此方面不够了解 * 开发难度大，工程量高 * 在多种平台上的移植困难 * 合作方难以找到 | -对于陌生的领域积极的进行学习  -投入更多的时间 | -适应合作方的要求  -积极联系合作方  -积极的进行学习 |

微信小程序

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 即用即走 * 跨平台开发 * 容易达成线上线下场景的连接与互动 * 开发成本低 | * 入口很深 * 展示很简陋 * 受控于微信 * 无法开发较大型的程序 |
| 机会(Opportunities) | SO | WO |
| * 巨大流量入口 * 技术较为成熟 * 满足客户新鲜感体验 | 在小程序框架内，专注做低频、刚需的场景，扩大渠道用户 | 小程序使用成熟的H5做轻量级的广告游戏等营销获客 |
| 风险(Threats) | ST | WT |
| * 数据安全较低 * 自有度受制于腾讯 * 对自家产品造成冲击 | -重视用户的信息安全、隐私保护等问题  -利用小程序的便捷性向APP引流 | APP与小程序并行，APP做高频、验密的场景 |

通过比较，我们可以发现iOS花费开销较大，而且我们小组缺乏iOS相关设备，所以暂时并不考虑iOS。

微信小程序开发可以尝试，最近小程序开发火热，不过小程序不支持超过2M的程序，而且我们的开发软件严格来说不符合微信小程序的要求，可能出现问题，暂不考虑。

安卓系统内嵌开发难度大，我们目前时间不足，不足以进行深入的学习，不能完全掌握技术，暂不考虑。

综上所述，我们最后选择了Android 5.0

**操作系统：**

Windows7

优点：BUG出现的几率小，软件的兼容性较好

缺点：Win7已经停止服务支持，安全性降低

Windows10

优点：游戏性能更出色，软件兼容全平台，安全性提升

缺点：容易崩溃

**开发软件：**

SQL Server 2008

优点:安全性高、高效、智能、稳定性高

缺点：需要付费，未学习

费用：344.40（华东区包月20G）(阿里云报价)

MySQL

优点：体积小、速度快、总体拥有成本低、开源、现阶段正在学习相对比较熟悉

缺点：稳定性较差、存储结构相对混乱

费用：免费、开源

Android Studio

优点：多屏预览、即时预览、操作简单、更多的智能补全机制、速度更快、UI编辑器强大

缺点：文件组织方式不如Eclipse、版本兼容性差、无法使用自定义的framework

费用：免费、开源

Eclipse

优点：

占用内存少、文件组织方式合理、相对比较熟悉、有一定的智能补全机制

缺点：UI设计比较简陋、占CPU

费用：免费

**办公软件：**

Microsoft Office 2016

优点：操作方便、稳定、跨平台（win7以上、  
Android 4.4以上）

缺点：必须同一版本不然存在兼容性问题、付费使用

费用:499（包年）（包涵了Microsoft project）

**界面设计工具：**

Axure RP

优点：使用广泛，沟通、传输、修改方便，比较成熟的专业原型设计工具。功能一般比其他的软件较为完善，使用起来操作复杂度不会很高，上手容易

缺点：需要收费，还从未接触过，不熟悉

费用：专业版 4000团队版 7000企业版 9000

**负载测试工具：**

Jtest

优点：使预防代码错误成为可能，节约成本，提高软件质量和开发效率，使单元测试——包括白盒、黑盒以及回归测试成为可能，使代码规范检查和自动纠正成为可能

缺点：价格昂贵，从未接触过，需要学习

Junit

优点：Free，使用方便，撰写测试非常简单，测试检验其结果并提供立即的回馈

缺点：从未接触过，需要学习

费用：开源，免费

# 3．总体设计

## 3.1处理流程

处理流程：

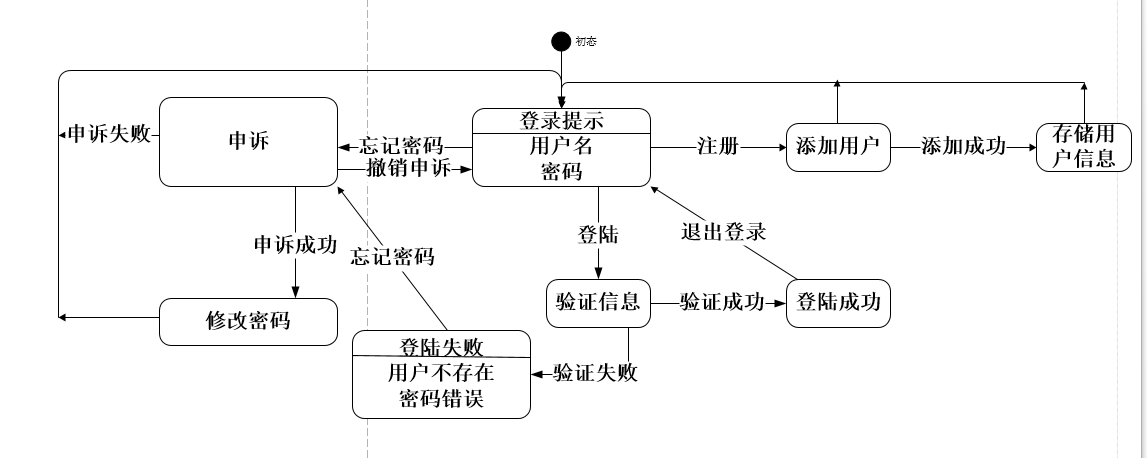
新建事务：提供事务信息->将申请事务录入云端数据库

修改事务：提供要修改事务信息->获得云端数据库相关事务信息->提供相关修改信息->将云端数据库信息更新

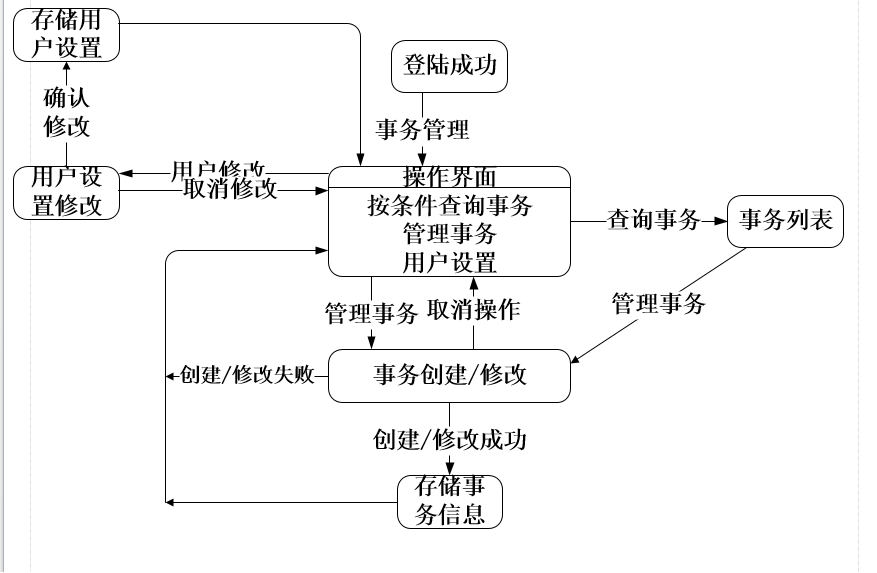
查询事务：提供查询事务信息->通过相关信息查询云端数据库->获得相关事务信息

删除事务：提供删除事务信息->将云端数据库信息更新->进一步删除垃圾桶->将云端数据库的信息删除

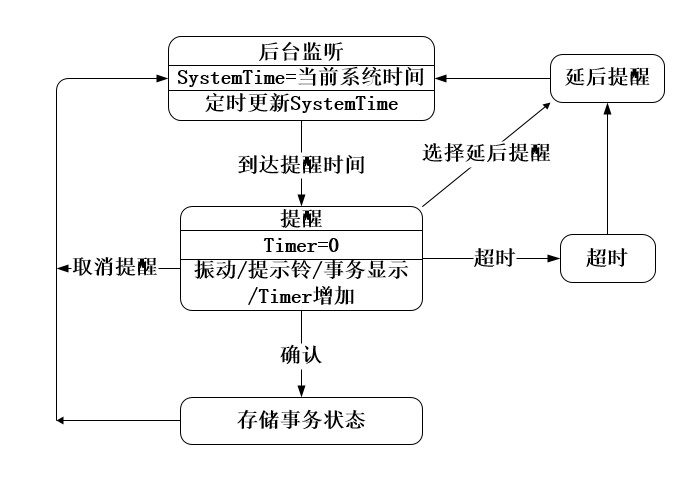
### 3.1.1 登录



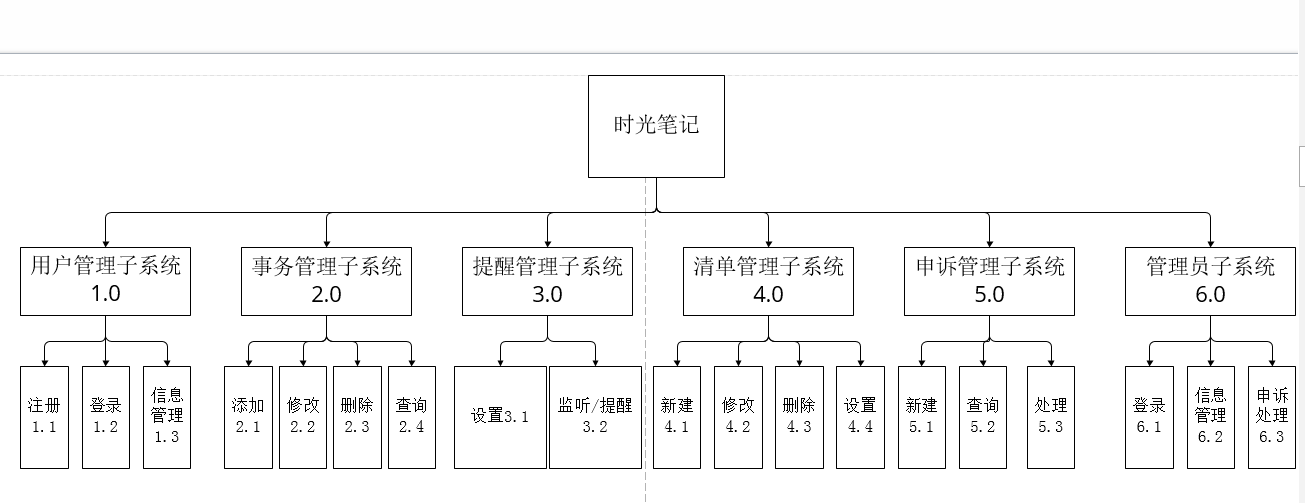
### 3.1.2事务管理



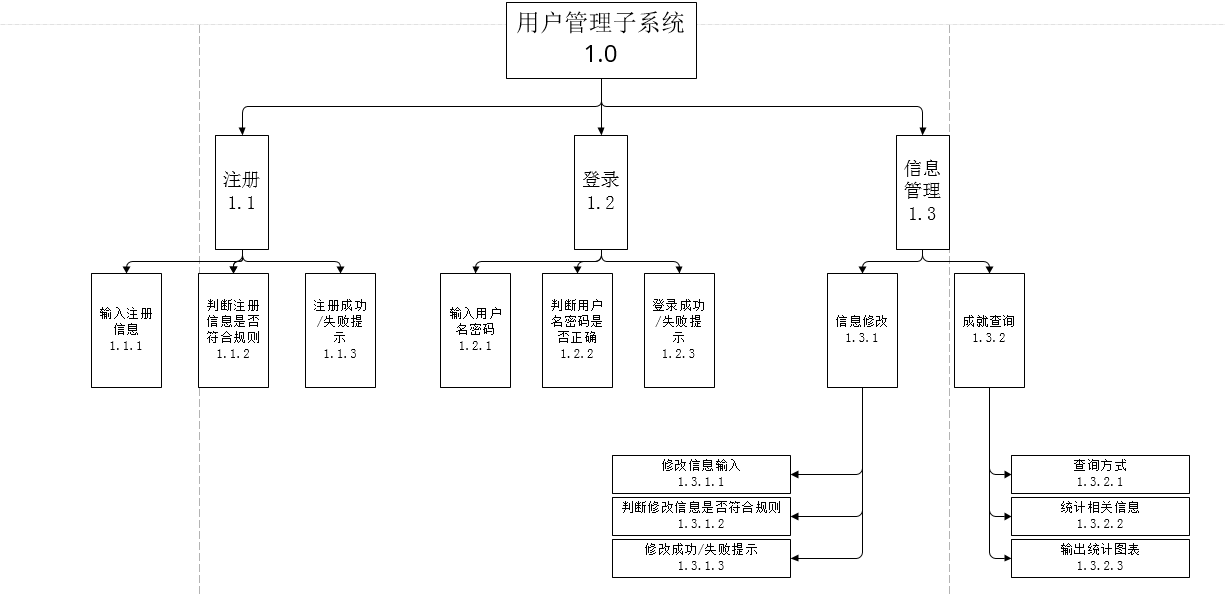
### 3.1.3提醒业务



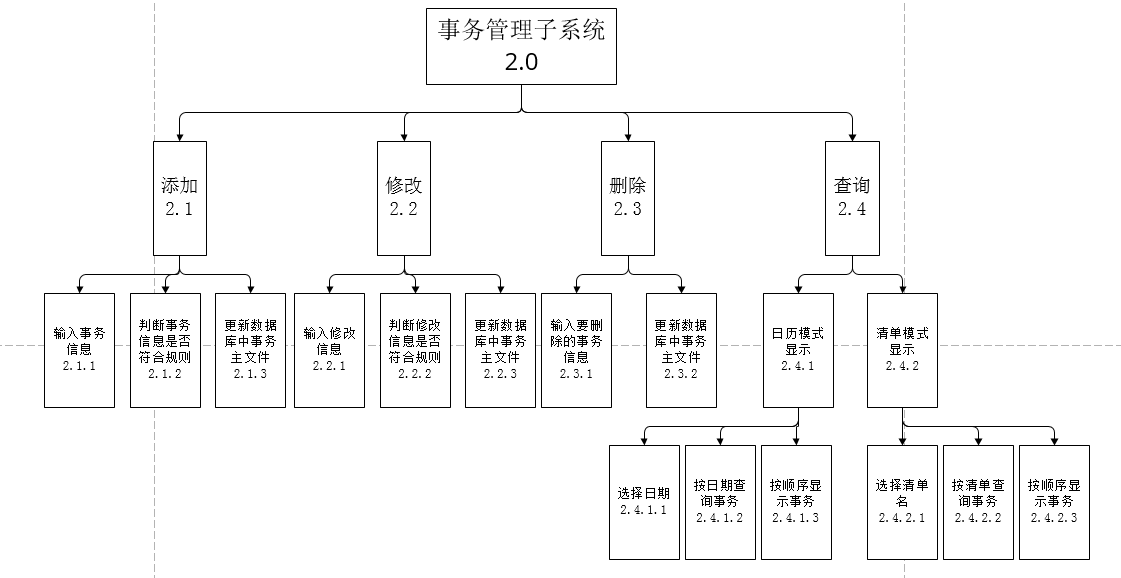
## 3.2功能分配



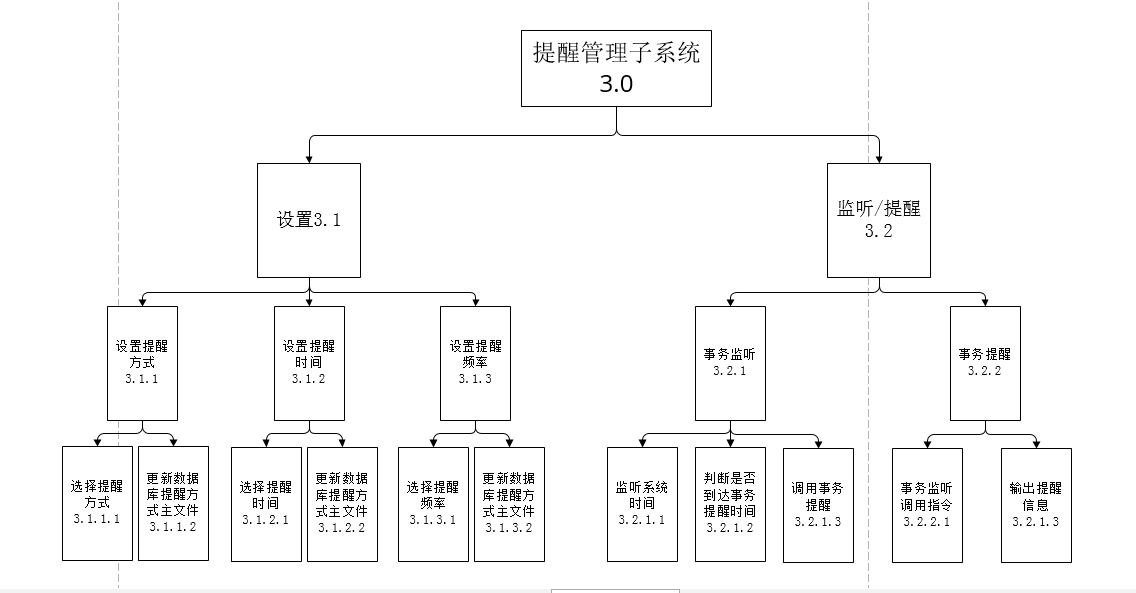
### 3.2.1用户管理模块



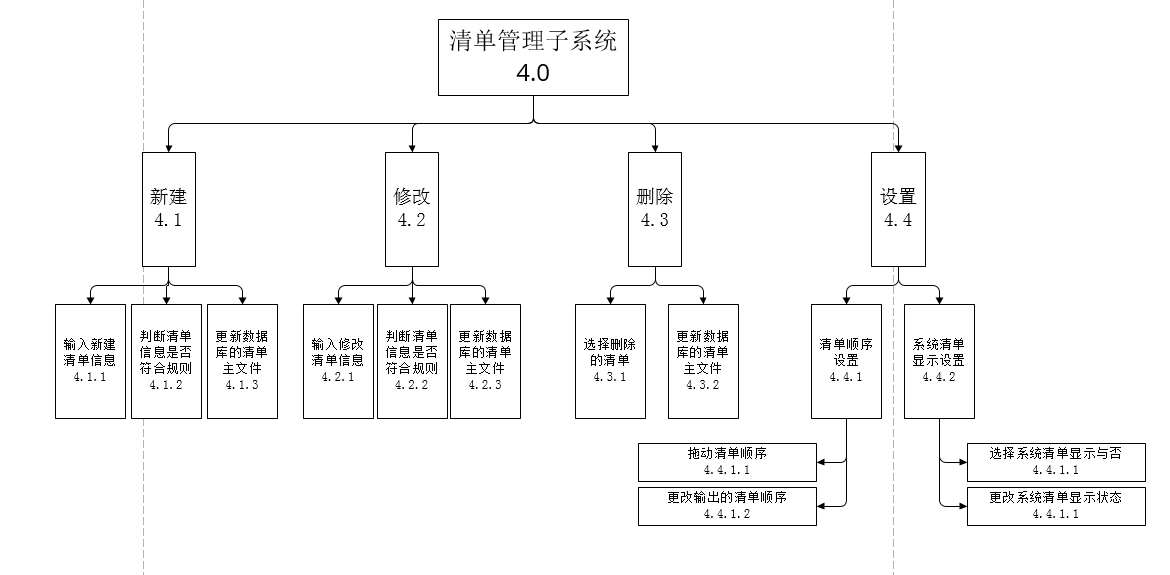
### 3.2.2事务管理模块



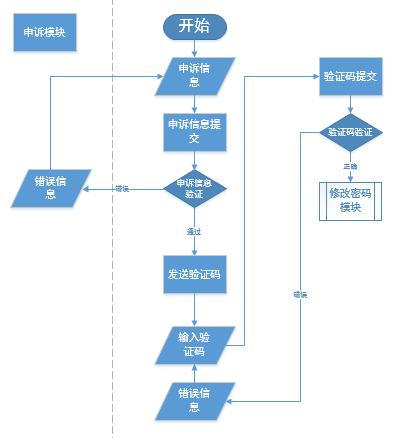
### 3.2.3提醒管理模块



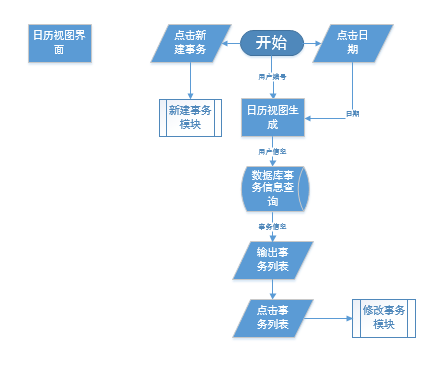
### 3.2.4清单管理模块



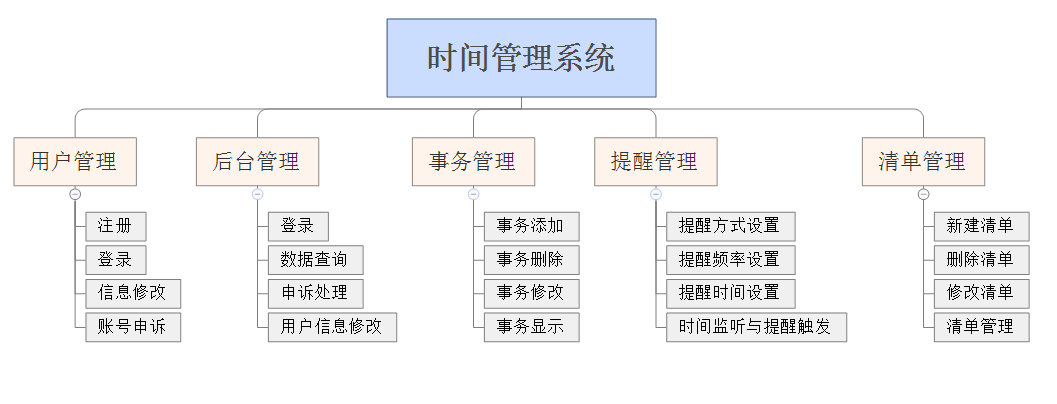
### 3.2.5申诉管理模块



### 3.2.6日历视图模块



## 3.3总体结构和模块外部设计



# 4．接口设计

## 4.1外部接口

### 4.1.1硬件接口

服务器配置：1M带宽、1核（Inter（R） Xeon（R）CPU E5-2682 v4　@ ２.５０ＧＨｚ　２.４９ＧＨｚ）、２Ｇ内存、４０Ｇ系统盘

其他子系统：普通智能机、无线网卡

### 4.1.2软件接口

服务器配置：

Ｗｉｎｄｏｗｓ　Ｓｅｒｖｅｒ２００８　Ｒ２企业版、ＭｙＳＱＬ５．６、ＪＤＫ　１.８以上

其他子配置：

Ａｎｄｒｏｉｄ５.０以上

## 4.2内部接口

所有子系统都要通过wifi模块将信息传递到云端数据库，其中申诉信息由管理员子系统查询、处理后再传送到云端数据库。

## 4.3用户接口

用户通过手机屏幕触摸激活各个功能模块，向系统发送相应信息。

# 5．数据结构设计

## 5.1逻辑结构设计

### 5.1.1表UserInformation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 实际字段名 | 类型 | 是否能为空 | 键型 | 说明 |
| 用户编号 | UserId | varchar(20） | 否 | 主码 | 唯一地标识消用户的关键域 |
| 用户名称 | UserName | varchar(50） | 否 |  | 用户的名字或昵称 |
| 用户密码 | UserPassword | varchar(20) | 否 |  | 用户自定义的密码信息 |
| 用户邮箱 | UserEmail | varchar(50） | 否 |  | 用户自身的邮箱信息 |
| 默认提醒设置编号 | DRemindId | varchar(20） | 否 | 外键 | 唯一的标识提醒设置的关键域 |
| 创建日期 | CreatDate | DATETIME | 否 |  | 用户创建的日期信息 |
| 停用日期 | StopDate | DATETIME |  |  | 用户停用的日期信息 |
| 停用人编号 | StopUserId | varchar(20) |  |  | 唯一的标识管理员的关键域 |
| 日历模式 | UserCalendar | Boolean | 否 |  | 是、否代表阳历和农历 |
| 字体大小 | UserTypeface | varchar(10） | 否 |  | 仅能为“大”、“中”、“小”三个字段 |
| 番茄钟设置编号 | PomoId | int | 否 | 外键 | 唯一识别番茄钟设置的关键域 |
| 成就值 | Achievement | int | 否 |  | 成就值=1\*计划数+2\*完成数+1\*番茄数 |

### 5.1.2表EventInformation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 实际字段名 | 类型 | 是否能为空 | 键型 | 说明 |
| 事务编号 | EventId | varchar(20) | 否 | 主码 | 唯一地标识用户事务信息的关键域 |
| 事物名称 | EventName | varchar(50) | 否 |  | 事务的主题名称信息 |
| 事务备注 | EventNote | varchar(200) |  |  | 描述事务的文字信息 |
| 事务优先级 | EventPriority | {1,2,3,4,5} |  |  | 事务的重要等级信息 |
| 用户编号 | UserId | varchar(20) | 否 | 外码 | 唯一地标识消用户的关键域 |
| 清单编号 | ListId | varchar(20) | 否 | 外码 | 唯一地标识事务所属清单的关键域 |
| 提醒编号 | DRemindId | varchar(20） |  |  | 唯一的标识提醒设置的关键域 |
| 事务时间 | EventDate | DATETIME |  |  | 事务需要完成的时间 |
| 完成时间 | EventFinishDate | DATETIME |  |  | 事务实际完成的时间 |

### 5.1.3表ListIformation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 实际字段名 | 类型 | 是否能为空 | 键型 | 说明 |
| 清单编号 | ListId | varchar (20) | 否 | 主码 | 唯一地标识事务所属清单的关键域 |
| 用户编号 | UserId | varchar (20) | 否 | 外码 | 唯一地标识消用户的关键域 |
| 清单名称 | ListName | varchar (50) | 否 |  | 清单的名称信息 |
| 创建日期 | CreatDate | DATETIME | 否 |  | 创建日期 |
| 背景颜色 | BackgroundColor | Varchar (10) |  |  | 清单的所属的事务显示的背景颜色 |

### 5.1.4表DRemindInformation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 实际字段名 | 类型 | 是否能为空 | 键型 | 说明 |
| 提醒编号 | DRemindId | varchar（20） | 否 | 主码 | 唯一的标识提醒设置的关键域 |
| 提醒方式 | DRemindModel | varchar（200） | 否 |  | 事务到达提醒时间时的提醒方式 |
| 提醒频率 | DRemindRepeat | varchar（20） | 否 |  | 事务提醒用户的频率 |
| 提醒时间 | DRemindDate | DATETIME | 否 |  | 事务设定的提醒时间 |
| 用户编号 | UserId | varchar (20） | 否 | 外码 | 唯一地标识消用户的关键域 |

### 5.1.5表PomoTimer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 实际字段名 | 类型 | 是否能为空 | 键型 | 说明 |
| 叶子钟设置编号 | PomoId | int | 否 | 主键 | 唯一识别番茄钟设置的关键域 |
| 叶子钟时长 | PomoDuration | int | 否 |  | 每个番茄的时长 |
| 短休息时长 | ShortBreakDuration | int | 否 |  | 每个短休息的时长 |
| 长休息时长 | LongBreakDuration | int | 否 |  | 每个长休息的时长 |
| 长休息间隔 | LongBreakEvery | int | 否 |  | 多少番茄进行一次长休息 |

## 5.2物理结构设计

见5.1

## 5.3数据结构与程序的关系

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 用户模块 | 事务模块 | 提醒模块 | 清单模块 | 申诉模块 | 管理员模块 |
| UserInformation | 调用 |  |  |  |  |  |
| EventInformation |  | 调用 |  |  |  |  |
| ListIformation |  |  |  | 调用 |  |  |
| DRemindInformation | 调用 |  | 调用 |  |  |  |
| UserManagerInformation |  |  |  |  |  | 调用 |
| AppealInformation |  |  |  |  | 调用 | 调用 |
| PomoTimer | 调用 | 调用 |  |  |  |  |

# 6．运行设计

## 6.1运行模块的组合

见3.2

## 6.2运行控制

软件运行主要通过触屏点击和滑动来调用不同的子系统，以实现与云端数据库的交互。

## 6.3运行时间

每个子系统的响应时间应在1-2s之内，给云端数据库处理信息留有一定的时间余地。

# 7．出错处理设计

## 7.1出错输出信息

### 7.1.1登录模块错误

|  |  |
| --- | --- |
| 错误类型 | 错误信息 |
| 用户名小于6个字符或大于20个字符 | “输入用户名不符合规则” |
| 用户名不存在 | “用户名不存在” |
| 密码不匹配 | “输入账号或密码错误” |
| 匹配时间超过30秒 | “登录超时，请检查网络” |

### 7.1.2注册模块错误

|  |  |
| --- | --- |
| 错误类型 | 错误信息 |
| 用户名少于6个字符多余20个字符 | “输入用户名不符合规则” |
| 用户名已经存在 | ”用户名已存在” |
| 昵称少于1个字符多余50字符 | ”输入昵称不符合要求” |
| 密码少于8位字符多余20个字符 | ”输入密码不符合要求” |
| 密码与密码确认不匹配 | ”输入密码不一致” |
| 邮箱格式错误 | “输入邮箱不符合规则” |

### 7.1.3信息修改模块错误

|  |  |
| --- | --- |
| 错误类型 | 错误信息 |
| 新密码少于8位字符多余20个字符（为空除外） | ”输入密码不符合要求” |
| 新密码与新密码确认不匹配 | ”输入密码不一致” |
| 昵称多余50字符 | ”输入昵称不符合要求” |
| 邮箱格式错误 | “输入邮箱不符合规则” |

### 7.1.4事务添加模块错误

|  |  |
| --- | --- |
| 错误类型 | 错误信息 |
| 新密码少于8位字符多余20个字符（为空除外） | ”输入密码不符合要求” |
| 新密码与新密码确认不匹配 | ”输入密码不一致” |
| 昵称多余50字符 | ”输入昵称不符合要求” |
| 邮箱格式错误 | “输入邮箱不符合规则” |

## 7.2出错处理对策

如果出现网络异常错误，将采用读取手机内的存储信息来作为提醒管理的依据。

如果云端服务器故障（出现数据丢失），将再修复后载入备份数据（由管理员每日备份）。

# 8.安全保密设计

用户登录时需要密码，三次错误后需要半分钟后才能再次尝试。

申诉操作需要再用户账号的注册邮箱认证后才能允许修改密码。

# 9.维护设计

系统维护工作贯穿于系统的整个运行过程中，包括：系统应用程序的维护、数据的维护、代码的维护、硬件设备维护，系统维护的重点是系统应用软件的维护工作。而系统维护工作不应总是被动的等待用户提出要求后才进行，应进行主动的预防性维护。系统在运行时，也要随着环境的变化根据不同的需求及变化对系统进行必要的修改，使得系统功能更加完善。

当产生会到时数据不统一的造作错误时，需要进行事务回滚。

管理员需要定期进行数据备份，以防云服务器故障导致不可挽回的信息损失。