<一心·时间管理app>

软件架构文档

版本 <1.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <31/10/2020> | <1.0> | <创建> | <沈佳威> |
| <08/11/2020> | <1.1> | <更新> | <沈佳威> |
|  |  |  |  |
|  |  |  |  |

目录

简介 5

目的 5

参考资料 5

用例视图 5

Use case 总图 5

Use case图 User功能 5

Use case图 Alarm功能 6

Use case图 Friends功能 6

Use case图 Study功能 7

Use case图 Management功能 7

Use case图 Statistics功能 7

注册账户 7

登录账户 8

管理个人信息 8

管理软件风格 8

查看通知 9

手机短信验证 9

管理闹钟 9

闹钟唤醒 9

闹钟分享 9

好友管理 10

开启学习模式 10

结束学习模式 10

学习分享 10

生成报表 11

报表分享 11

用户权限管理 11

发布通知 11

逻辑视图 12

概述 12

在构架方面具有重要意义的设计包 12

登录注册服务 12

安全服务 12

自习室服务 12

好友社交服务 12

时间管理服务 12

闹钟服务 12

进程视图 13

部署视图 13

APP端 13

注册服务 13

闹钟服务 13

学习服务 13

好友服务 13

自习室服务 13

实现视图 14

数据视图（可选） 15

核心算法设计（可选） 16

页面视图 16

设计战术 18

软件架构文档 （简化版）

# 简介

## 目的

本文档提供了系统的全面架构概述，使用许多不同的架构视图来描述系统的不同方面。它的目的是捕获和传达在系统上做出的重要架构决策。

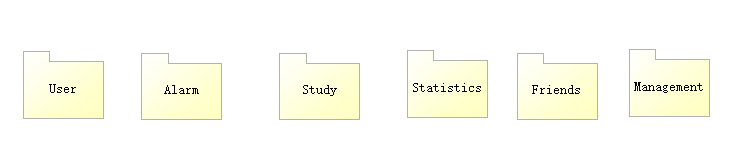
## 参考资料

[1] 沈备军&陈昊鹏&陈雨亭 《软件工程原理》 高等教育出版社

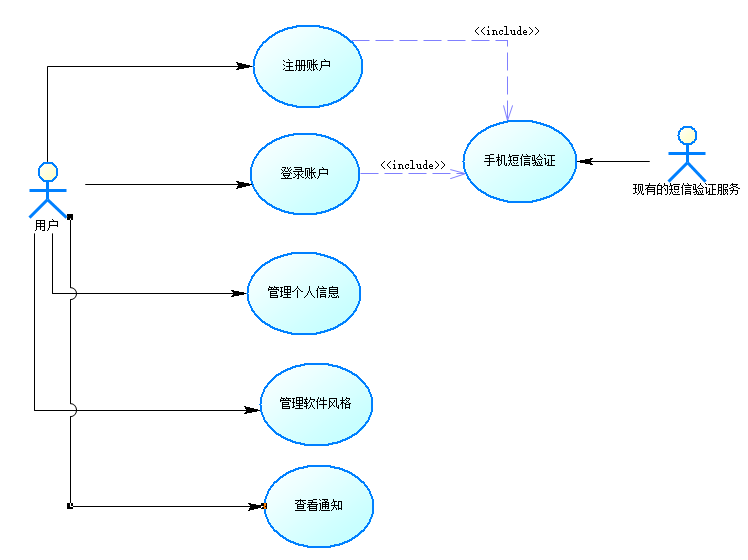
[2] S. Johnson Course Registration System Software Architecture Document

# 用例视图

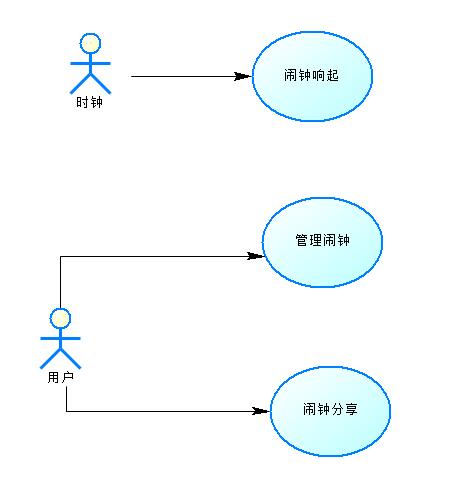
### Use case 总图



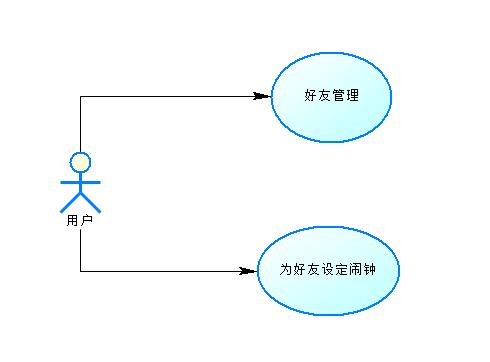
### Use case图 User功能



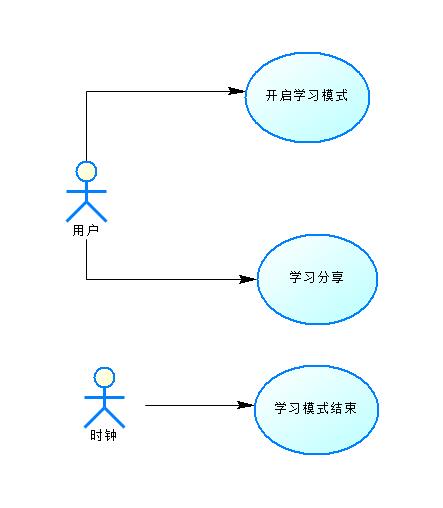
### Use case图 Alarm功能



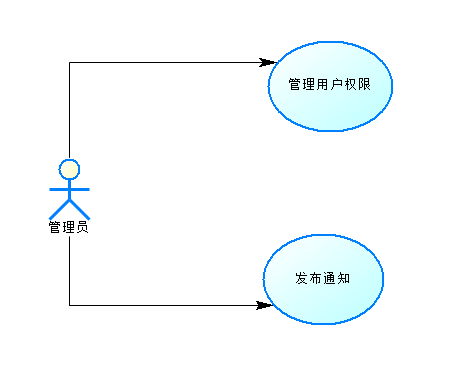
### Use case图 Friends功能



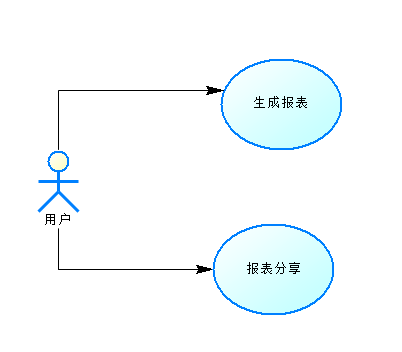
### Use case图 Study功能



### Use case图 Management功能



### Use case图 Statistics功能



### 注册账户

描述：用户注册账户，用于获取使用该产品的权限。

执行者：用户

后置条件：系统记录新创建用户的信息。

基本流：1.用户点击初始界面的“注册”按钮

2.用户填写手机号码，并点击获取验证码

3.现有的短信认证服务调用“手机短信验证”用例

4.用户填写用户名、密码、姓名、性别、电子邮箱等信息

5.系统检查用户密码的安全性以及电子邮件格式

6.系统向用户输入电子邮件地址发送确认邮件

7.用户打开确认邮件激活账户

备选流：2a.用户输入手机号格式不正确

提示用户手机号码格式错误，请重新输入。

5a.用户输入的密码安全性过低

提示用户更改密码使密码有更高的安全性

5b.用户输入电子邮件地址格式不正确

提示用户邮箱格式不正确

### 登录账户

描述:用户登录账户以使用该app的功能

执行者:用户

前置条件:用户已经注册了账号

基本流:1.用户进入主界面,输入手机号和密码,点击“登录”按钮

2.系统获取手机号,检验到账户存在

3.系统将密码与数据库中的信息匹配成功

4.用户登录成功

备选流:1a.用户点击“手机验证码登录”,并输入手机号

现有的短信认证服务调用“手机短信验证”用例

2a.系统检测到账户不存在

提示用户账户不存在,请先注册账户,用例结束

3a.系统检测到密码与数据库中存储信息不一致

提示用户密码错误,请重新输入密码

### 管理个人信息

描述:用户管理自己账户的个人信息

执行者:用户

前置条件:用户已经登录自己的账户

基本流:1.用户点击底边栏“我的”按钮

2.用户点击“个人信息设置”

3.用户修改自己的个人信息并点击确认

4.系统更改数据库内的用户信息

### 管理软件风格

描述:用户更改软件的界面风格

执行者:用户

前置条件:用户已经登录自己的账户

基本流:1.用户点击底边栏“我的”按钮

2.用户点击“主题设置”

3.用户选择想要的主题

4.系统更改软件风格

### 查看通知

描述：用户查看管理员发布的系统最新通知

执行者：用户

前置条件：管理员已经发布了新通知

基本流：1.用户点击底边栏“我的”按钮

2.用户点击“关于一心”

3.用户查看通知

### 手机短信验证

描述:第三方认证服务利用用户手机号码提供认证服务

执行者: 现有的短信验证服务

前置条件:系统发送了认证请求

基本流:1.检验号码正确性

2.生成随机验证码发送给用户手机

3.接收用户输入的验证码并比对

4.比对成功返回一个正确的信息

备选流:1a.用户号码不存在

提示用户号码不存在,请使用正确的手机号

4a.比对失败

返回失败信息

### 管理闹钟

描述:用户设置、修改和删除个性化闹钟

执行者：用户

前置条件：用户已经登录自己的账户

基本流：1.用户点击底边栏的“闹钟”按钮

2.用户点击右上角的“+”按钮

3.用户设置闹钟的时间、标签、铃声、是否重复以及任务类型并点击保存

4.新建闹钟成功

备选流：2a.用户想修改闹钟

点击列表中的存在的闹钟，进入详情页

2b.用户想删除闹钟

点击列表中存在的闹钟，进入详情页，点击右上角的删除，用例结束

2c.用户想用自然语言设置闹钟

点击“语音设置”按钮，并说出要求。用例结束

### 闹钟唤醒

描述：规定时间到，时钟唤醒闹钟

执行者:时钟

前置条件：用户已经登录，且已经设置了想要的闹钟

后置条件：系统给予用户相应的任务奖励

基本流：1.指定时间到达，闹钟响起

2.用户完成规定的任务，闹钟关闭

备选流：2a.用户未完成规定的任务

闹钟持续响起直到用户完成任务

### 闹钟分享

描述：用户完成一次闹钟任务后，将其分享至社交平台或生成截图

执行者：用户

前置条件:用户完成了闹钟任务

基本流：1.系统询问是否分享本次闹钟任务

2.用户选择是并选择分享到第三方社交软件

3.系统将任务闹钟结果分享到第三方社交软件

备选流：1a.用户选择否

用例结束

2a.用户选择生成截图

系统为用户生成本次任务闹钟的截图并保存到相册，用例结束。

### 好友管理

描述：用户管理好友，包括添加、删除以及查看好友信息等

执行者：用户

前置条件：用户已经登录自己的账户

基本流：1.用户点击底边栏的“好友”按钮

2.用户点击右上角的“+”按钮

3.用户输入好友用户名或手机号搜索其他用户

4.系统查找到相应用户

5.用户点击“添加好友”按钮

6.系统向目标用户发起好友请求

7.目标用户接收到请求并选择“同意”

8.添加好友成功

备选流：2a.用户想查看好友信息

用户点击好友列表中想要查看信息的好友，查看信息。用例结束。

2b.用户想删除好友

用户点击好友列表中想要删除的好友，在详情页点击删除。用例结束。

### 开启学习模式

描述：用户开启学习模式，该模式下，用户必须在指定时间内专注学习，手机的使用收到限制。

执行者：用户

前置条件:用户已经登录自己的账户

基本流：1.用户点击底边栏的“学习”按钮

2.用户转动时间选择器选择学习时间

3.用户点击“开始学习”开启学习模式

4.持续时间内，用户远离手机学习

5.持续时间到，学习模式结束

6.系统给予用户相应的奖励

备选流：4a.持续时间内，用户使用了手机中不允许使用的应用

使用不允许使用的应用持续10s后，学习模式强制结束，用例结束。

### 结束学习模式

描述：指定时间到达，系统结束学习模式并结算积分

执行者:时钟

前置条件：用户开启了学习模式，并在规定时间内没有使用手机

基本流：1.时钟检测到时间结束

2.系统结束学习模式

3.系统结算积分

### 学习分享

描述:用户将自己远离手机学习的结果分享到社交平台或者生成截图。

执行者：用户

前置条件:用户完成了学习模式的学习

基本流：1.系统询问是否分享本次学习

2.用户选择是并选择分享到第三方社交软件

3.系统将学习结果分享到第三方社交软件

备选流：1a.用户选择否

用例结束

2a.用户选择生成截图

系统为用户生成本次学习模式的截图并保存到相册，用例结束。

### 生成报表

描述:系统为用户生成最近一段时间内时间管理的报表，以图标形式展现

执行者:用户

前置条件：用户已经登录自己的账户

基本流：1.用户点击底边栏“我的”按钮

2.用户点击“我的报表”按钮

3.用户点击“生成报表”

4.系统为用户生成报表

备选流：4a.用户近期未使用过时间管理功能

系统提示“您近期未有时间管理记录”

### 报表分享

描述：用户将自己的时间管理报表分享到社交平台

执行者：用户

前置条件：用户已经生成了自己的时间管理报表

基本流：1.用户点击底边栏“我的”按钮

2.用户点击“我的报表”按钮

3.用户点击“分享报表”按钮

4.用户选择分享到第三方社交软件

5.系统将报表分享到第三方社交软件

备选流: 4a.用户选择生成截图

系统为用户生成报表的截图并保存到相册

### 用户权限管理

描述：管理员管理用户使用权限

执行者：管理员

前置条件：无

基本流：1.管理员登录网页端管理网站。

2.管理员点击侧边栏“用户管理”按钮

3.管理员查询想要更改权限的用户

4.管理员更改用户权限。

### 发布通知

描述：管理员发布系统相关的通知

前置条件：无

后置条件：用户收到通知

基本流：1.管理员登录网页端管理网站。

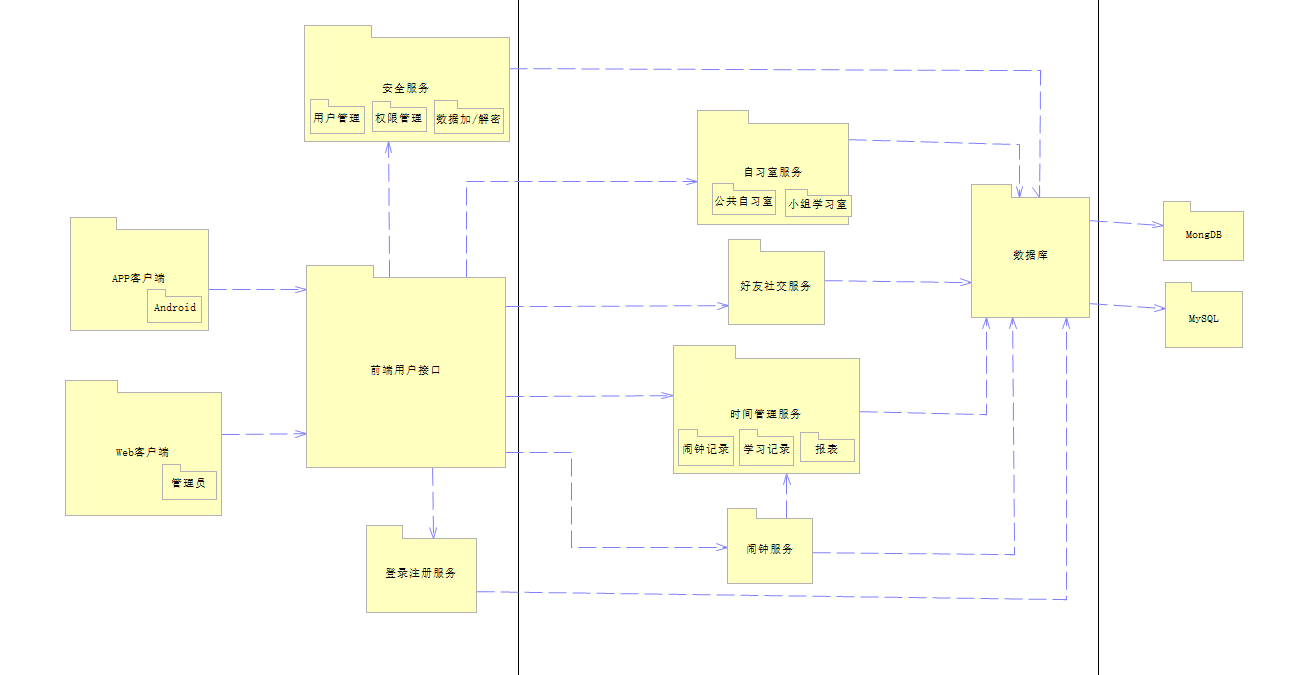
2.管理员点击侧边栏“通知发布”按钮

3.管理员编辑要发送的内容

4.管理员点击发送

# 逻辑视图

## 概述



我们将采用微服务架构，把一个大型的单个应用程序和服务拆分为数个甚至数十个的支持微服务，它可扩展单个组件而不是整个的应用程序堆栈，从而满足服务等级协议，用一些功能比较明确、业务比较精练的服务去解决更大、更实际的问题，在分散的组件中使用云架构和平台式部署、管理和服务功能。我们主要分为以下几个服务：登录注册服务、安全服务、自习室服务、好友社交服务、时间管理服务和闹钟服务，每个服务有着自己的数据库、可独立地进行开发、管理和迭代。

## 在构架方面具有重要意义的设计包

### 登录注册服务

用户可以用手机号码注册账号，在注册完成后可以完善自己的个人信息；已注册的用户可以用手机验证码或者密码登录账号，体验软件的完整功能。

### 安全服务

管理员可以进行用户的管理，设置用户的权限，传输的数据进行加密和解密。

### 自习室服务

用户可以进入公共自习室和他人一起学习，也可以创建小组学习室和好友一起学习。

### 好友社交服务

用户可以添加和删除好友、为好友添加闹钟、分享自己的学习时长。

### 时间管理服务

该服务分为三个模式，分别是普通模式（用户可使用部分手机功能）、深度模式（用户禁止使用手机）和番茄学习法（学习一定时长后可自由使用手机）。

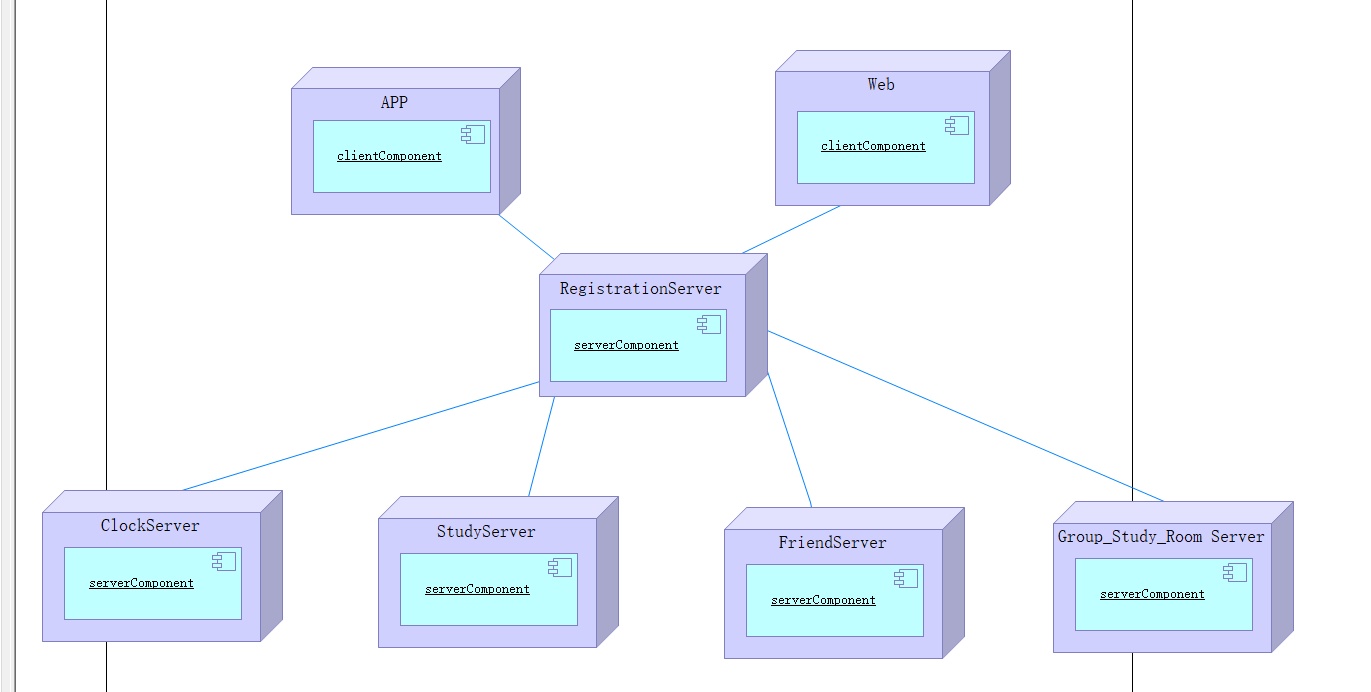
### 闹钟服务

用户可以在应用内指定闹钟的时间、任务、周期、奖励积分、铃声等，到用户设定的时间时，闹钟响起，用户需要完成指定任务才能够关闭闹钟继续使用手机。

# 进程视图

我们采用的是spring cloud的微服务架构，故不存在进程视图。

# 部署视图



### APP端

用户使用app，通过互联网拨号连接到服务器。

### 注册服务

注册服务器是服务器端限于成本将依赖于Amazon Web Service提供的一系列云计算服务。同时，它将通过JDBC来获取由MySQL及MongoDB支持的数据库服务。

### 闹钟服务

用户可以在应用内指定闹钟的时间、任务、周期、奖励积分、铃声等，到用户设定的时间时，闹钟响起，用户需要完成指定任务才能够关闭闹钟继续使用手机。

### 学习服务

该服务分为三个模式，分别是普通模式（用户可使用部分手机功能）、深度模式（用户禁止使用手机）和番茄学习法（学习一定时长后可自由使用手机）。

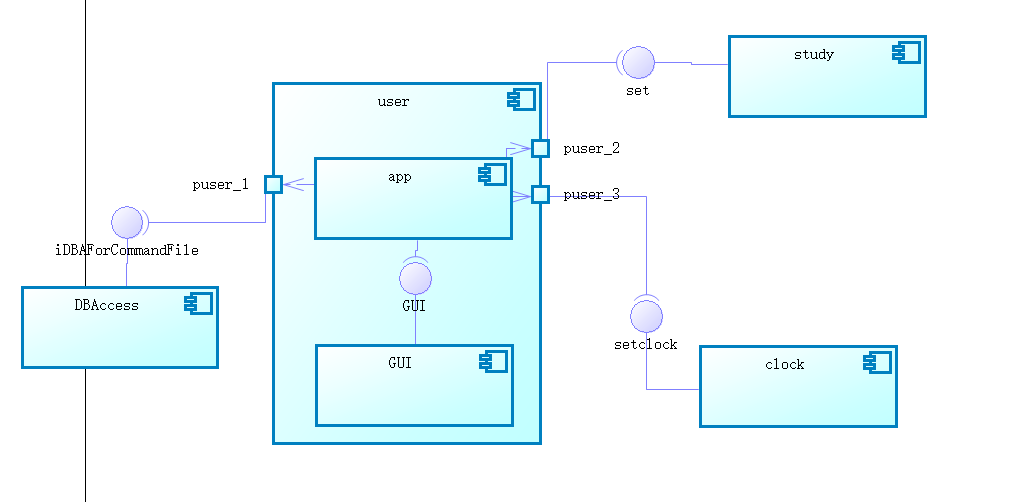
### 好友服务

用户可以添加和删除好友、为好友添加闹钟、分享自己的学习时长。

### 自习室服务

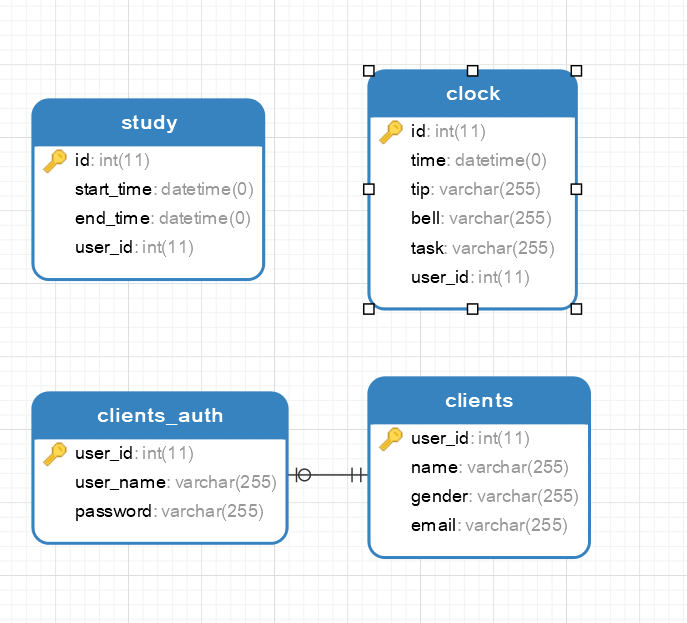
用户可以进入公共自习室和他人一起学习，也可以创建小组学习室和好友一起学习。

# 实现视图



User使用app,调用GUI，使用DBAccess，study和clock

# 数据视图（可选）



在mysql中有四个表：

Client：存用户的基本信息

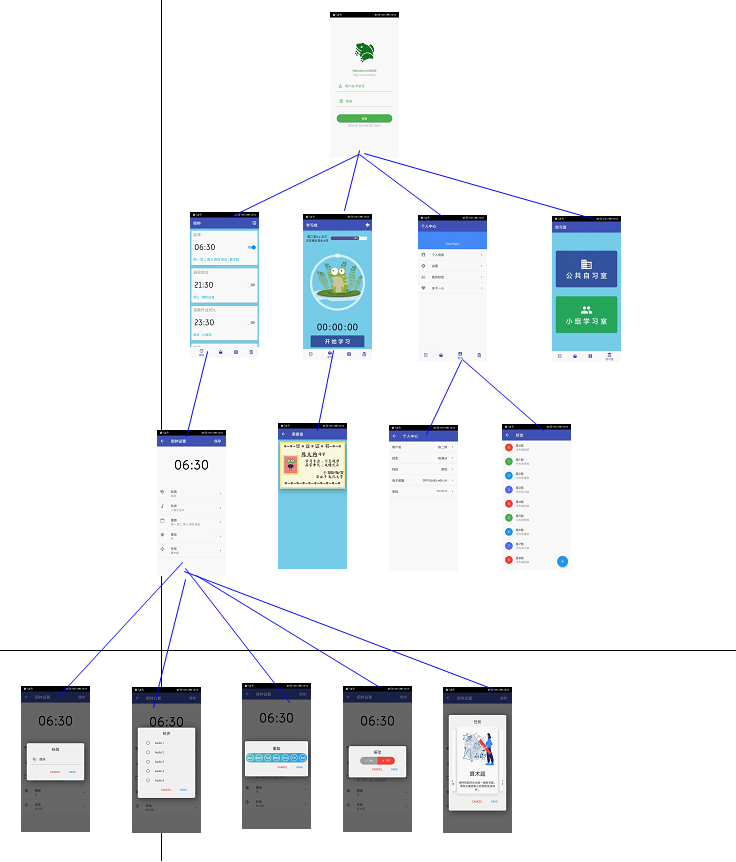
Clients\_auth：存用户账户密码

Clock：存闹钟信息

Study：存学习信息

其他如好友信息将放在mogodb中。

# 页面视图



页面如下：  
登录注册页面

闹钟页面-闹钟设置页面-标签设置，闹铃设置，重复时间设置，震动设置，任务设置

学习页面-荣誉墙页面

设置页面-个人中心页面，好友页面

小组学习页面

# 设计战术

|  |  |
| --- | --- |
| **质量因素** | **设计战术** |
| 易用性 | 1.为用户提供适当的反馈和协助，在获取用户权限前提醒用户  2.将用户接口与应用的其余部分分离  3.提供“取消”、“撤消”等命令 |
| 可用性 | 1.错误检测，使用微服务架构，有很好的监测系统运行状态的工具  2.错误恢复：在服务器和用户本地都维护有数据的备份，利用冗余提供错误恢复的能力  3.错误预防：事务机制、进程监视、从服务中删除 |
| 性能 | 1.资源需求：提高计算效率、减少计算开销、控制采样频率、限制队列大小  2.资源管理：微服务架构提供了良好的负载均衡策略，设计按需伸缩的系统架构  3.资源仲裁：先进先出、优先级调度 |
| 安全性 | 1.抵抗和检测攻击：Spring Security提供了用户身份验证与授权，通过加密关键信息限制暴露的信息  2.从攻击中恢复：维持审计追踪、采用可用性中恢复战术 |
| 可测试性 | 1.测试的输入/输出，记录/回放、将接口与实现分离、特化访问路线/接口  2.内部监视，内置监视器 |
| 可维护性 | 1.模块化，使得不同模块之间的耦合度降低，降低维护成本。  2.防止连锁反应，信息隐藏、维持现有接口、限制通信路径、仲裁者的使用  3.推迟绑定时间 |