**<**有我在**>**

软件架构文档

版本 **<1.1>**

[注：用方括号括起来并以蓝色斜体（样式*=InfoBlue*）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式*=Body Text*）。*]*

*[*要定制 *Microsoft Word* 中的自动字段（选中时显示灰色背景），请选择 *File>Properties*，然后将 *Title*、*Subject* 和 *Company* 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 *Edit>Select All*（或 *Ctrl-A*）并按 *F9*，或只是在字段上单击并按 *F9*，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 *Alt-F9*，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 *Word* 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 |
| <11/7/2019> | <1.0> | <第一次迭代> | <周欣仪> |
| <2/9/2019> | <1.1> | <第四次迭代> | <刘佳> |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4
   1. 目的 4
   2. 参考资料 4
2. 用例视图 4
3. 逻辑视图 7
   1. 三层架构 7

3.2表现层 7

3.3数据层 8

1. 部署视图 9
2. 技术视图 10
3. 质量属性的设计 10
   1. 易用性设计 10
   2. 可靠性设计 10
   3. 性能设计 10
   4. 可支持性设计 10

软件架构文档（简化版）

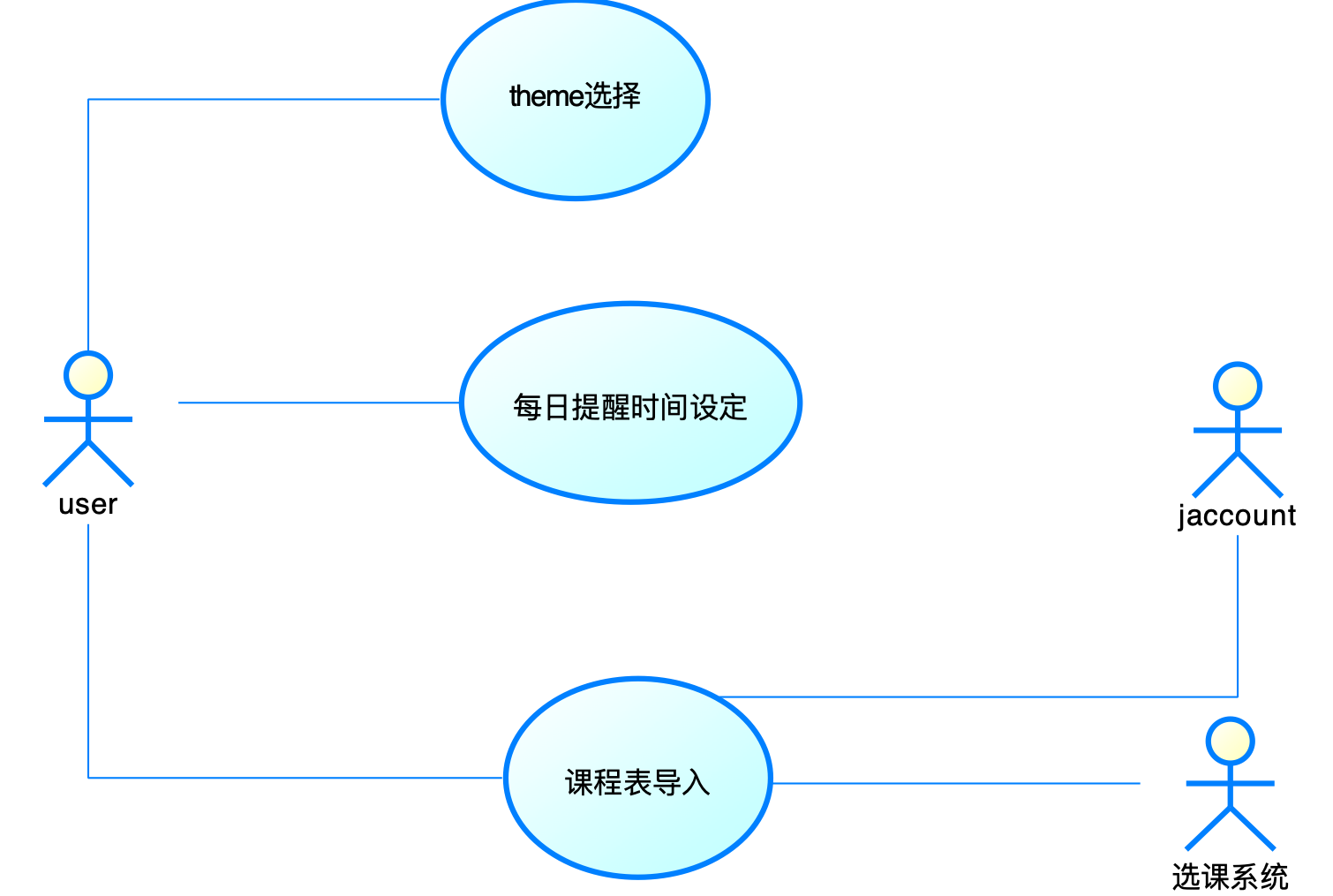
# 简介

## 目的

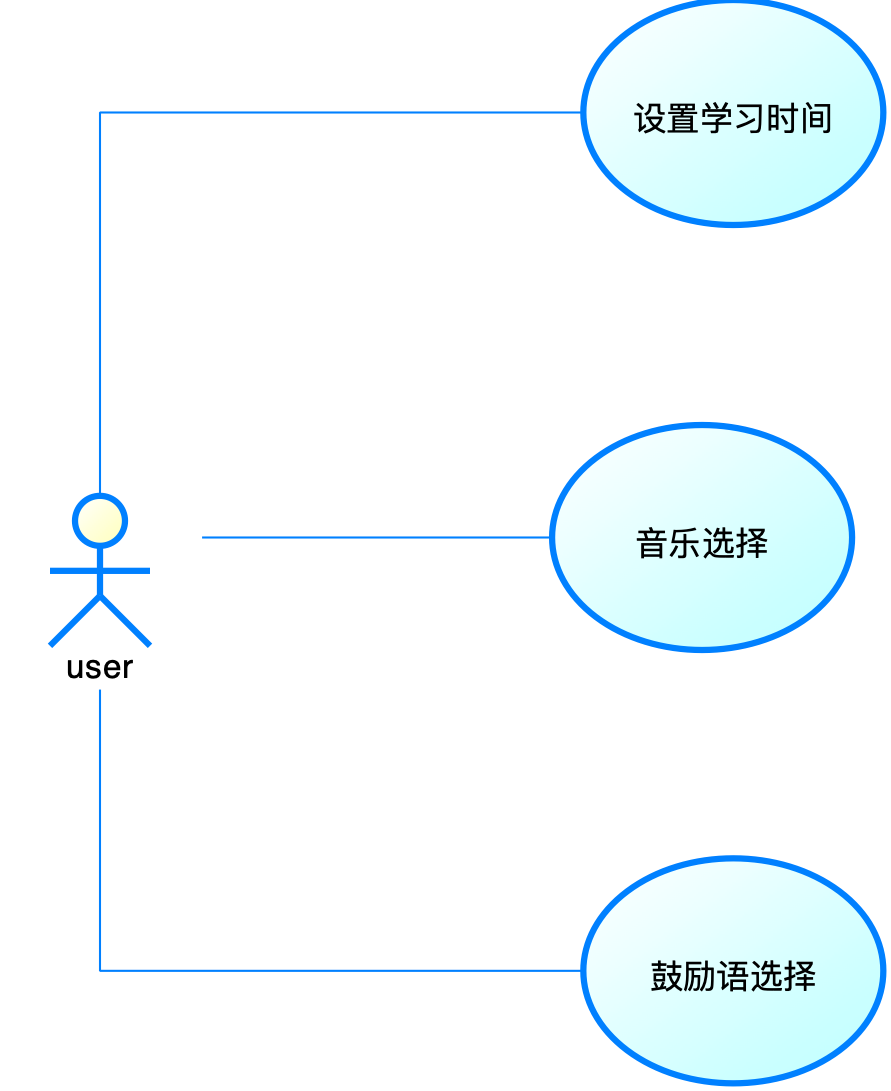
本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

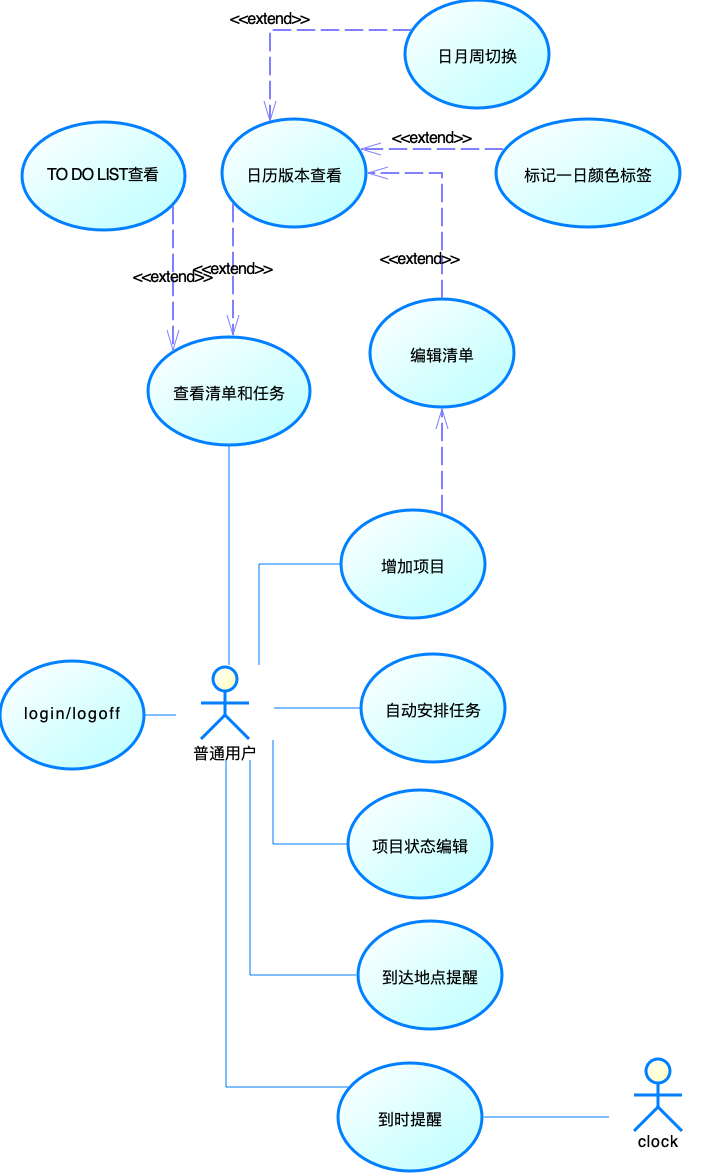
## 参考资料

# 用例视图



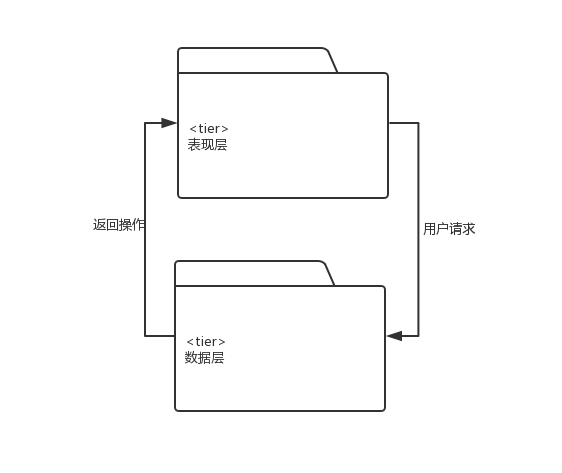
* 用户可在设置界面选择设置主题，在给定主题中选择主题，对应用进行换肤。
* 用户可以在登录界面选择jaccount登录，然后后台根据用户信息向选课系统获取用户课表（暂无权限）；也可由用户手动导入。
* 用户可在每次添加或修改任务的时候设置提醒时间、提醒周期。



* 用户在免打扰模式下可以设置自己的学习时长、选择自己想听的音乐和不同的鼓励语。
* 用户可以进入登录界面进行登录操作，可以注册或通过QQ、jaccount登录。
* 用户在主界面可以查看清单和任务列表，也可通过日历形式查看任务清单、调整任务时间、调整子任务的所属。
* 用户可通过“+”按钮添加新的任务并进入新任务的编辑界面。
* 用户可以在设置界面选择智能安排的选项，通过将本地数据传至后台进行人工智能算法处理来获取自动的任务安排。
* 用户可以直接在任务清单列表勾选任务前的复选框表示已完成任务。
* 若用户的某些任务设置了地点，那么当用户所处位置靠近目的地时，用户将会收到提醒通知。
* 若用户的某些任务设置了提醒时间，系统则会在指定时间弹出提醒用户的界面。

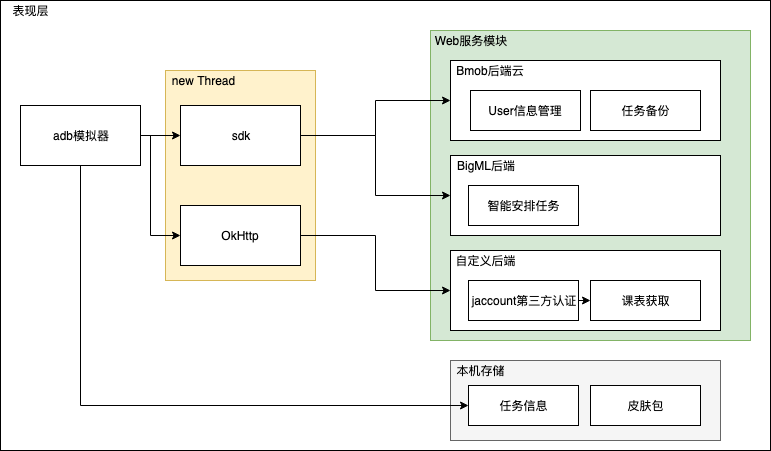
# 逻辑视图

## 三层架构



系统分为表现层和数据层。用户可在表现层与系统发起交互，当用户进行操作时，表现层调用读取数据层或者修改数据。数据层负责知识库的保存，以及准备调用所需数据。

## 3.2表现层

其中adb模拟器负责和用户交互，获取用户信息和用户定义的任务信息。如果选择不登录可直接使用adb的所有任务管理和智能安排的功能；如果选择登录可以完成除不登录外的数据备份和迁移、获取课表的功能。

## 3.3数据层

* 本机存储：使用.json文件直接保存本地插入的数据，每次本地打开应用，从中读取数据加载显示；后期云同步数据也准备直接传输.json文件。
* Bmob后端云：按照每个User分开存储，ArrayList<TaskCollector>属于每个User的属性。
* BigML计算：存储部分合理数据以此预测单个缺失属性的数据。

# 部署视图

# Picture 1

# 技术视图

本系统的技术选型如下：

1. 编程语言：Java

Java用于开发IDE插件前端，以及数据储存模块

Python用于进行算法相关的试验。

1. 使用android studio开发，espresso进行UI测试

# 质量属性的设计

## 易用性设计

需求：用户在第一次使用即可掌握本产品各种功能的使用。

## 可靠性设计

需求：

* 最小故障间隔时间：1周。
* 平均修复时间：小于等于2小时。
* 全年故障停运时间：小于等于30小时。

设计策略：使用Three Tier架构，不同Tier之间实现充分解耦。一旦出现故障，可以快速定位并修复。

## 性能设计

需求：

* 最大并发数：200，响应时间不应该超过5秒。
* 单机操作响应较快，不超过5秒。

设计策略：分析性能瓶颈可能在BigML，可升级使用学生套餐。使用异步通讯进行网络间通讯。

## 可支持性设计

需求：

* Java代码根据Java的标准风格规范进行编写。

设计策略：组员编程参照Google Java之间互相进行Code Review