原型技术方案设计与实现

目录

1 项目简介

1.1 项目背景

如今，是一个信息爆炸时代，在大量碎片化信息的冲击中，我们逐渐被信息洪流裹挟，面对碎片化的目标，突如其来的任务不知所措，而匆匆忙忙中，生活的碎片也被时间的洪流冲散，被失落在过去的某一时刻。因此，一款时间线应用对我们来说十分重要，规划时间，勾连记忆的碎片，记录生活，提高效率。

1.2 问题与机会

问题

- 如何正确的设计界面避免可能部分文字和信息太过于密集
- 满足一些用户使用时的习惯，如需要使用 `markdown` 来书写文字，夜间模式等
- 开发与运维成本还是一个未知数
- 市场推广方面，当前市场方向倾向于大公司流量推荐和应用商店独立推荐
- 如何布局全平台
- 应用社交模式的方向
- 分享的模式，如何通过分享吸引更多的用户
- 如何让软件更加智能，使其符合用户习惯

机会

- 市面上关于这种时间线类型应用比较少，可以抢占市场先机
- 添加用户群组，使得组织者可以查看其他人员的时间安排
- 经调查发现，存在着多数人群对记事软件的需求
- 部分企业寻求这样的管理员工时间线软件
- 现在大部分的记事 APP 都是个人开发，少部分是大公司开发，如果能够系统地去开发，可以抢占市场

1.3 项目介绍

时间线应用是什么？

一个以时间线和核心的应用，在这里你可以管理你的多条时间线。

只需要几步，你就能把你的时间线变成笔记、待办、日程、日记等，并以时间轴为载体进行记录。

时间线应用有哪里不同？

帮助大家经营人生琐碎，串成一条时间的轴线，以下是部分功能：

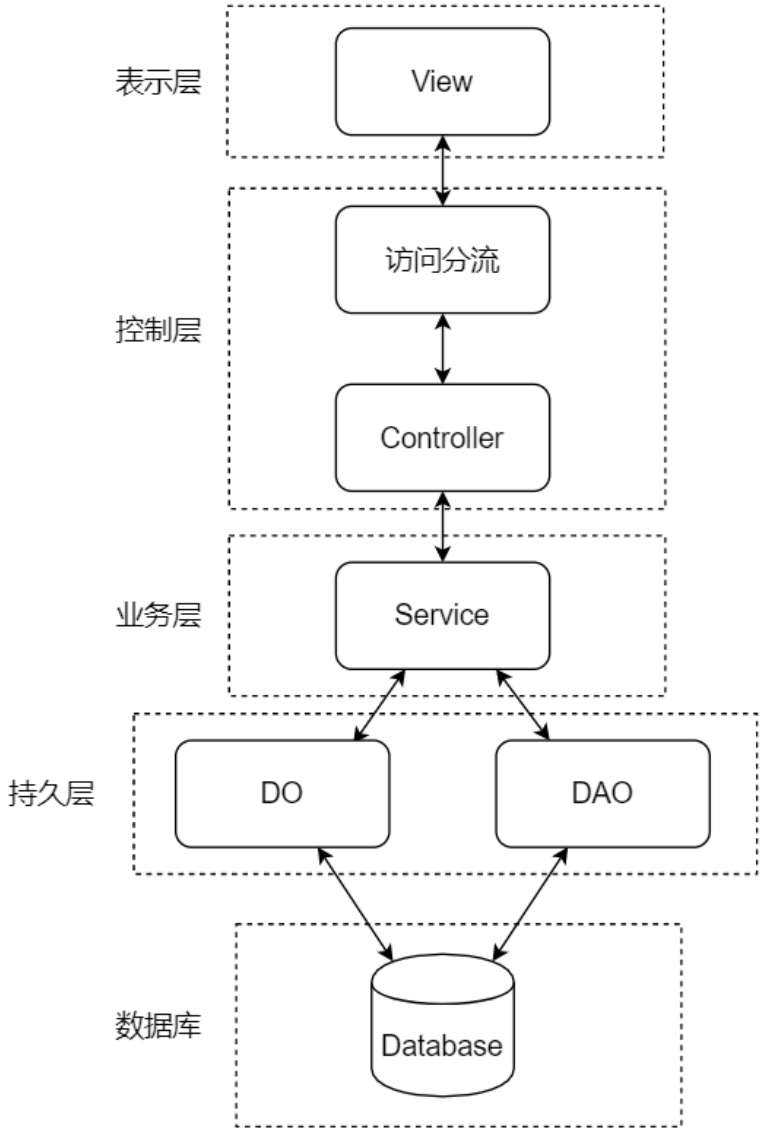
- 多条时间线相互独立，记录一切
- 智能日程安排和提醒
- 代办事项与自定义模版
- 与大家分享时间线，与组织共享时间安排
- 全平台通用，随时随地同步
- 社交，在相同的时间遇见不同的人
- 等等……

2 系统概要设计

2.1 总体架构设计

系统根据不同功能划分成不同模块，各模块在进一步划分，各个模块可以独立并行开发，为不同的子模块之间降低耦合性，各模块之间通过接口进行关联调用。衡量模块独立性的标准是耦合性和内聚性。耦合性，也称块间联系，用来度量软件系统结构中各模块之间相互联系的紧密程度。模块之间联系越紧密，耦合性就越强，模块的独立性就越差。内聚性，又称为块内联系，用来衡量模块的功能强度，也就是一个模块内部的各个元素之间彼此结合的紧密程度。如果一个模块内各元素联系得越紧密，则它的内聚性就越高。为软件系统进行模块划分时，要尽量做到高内聚和低耦合，从而提高模块的独立性。

系统总体架构图如下：



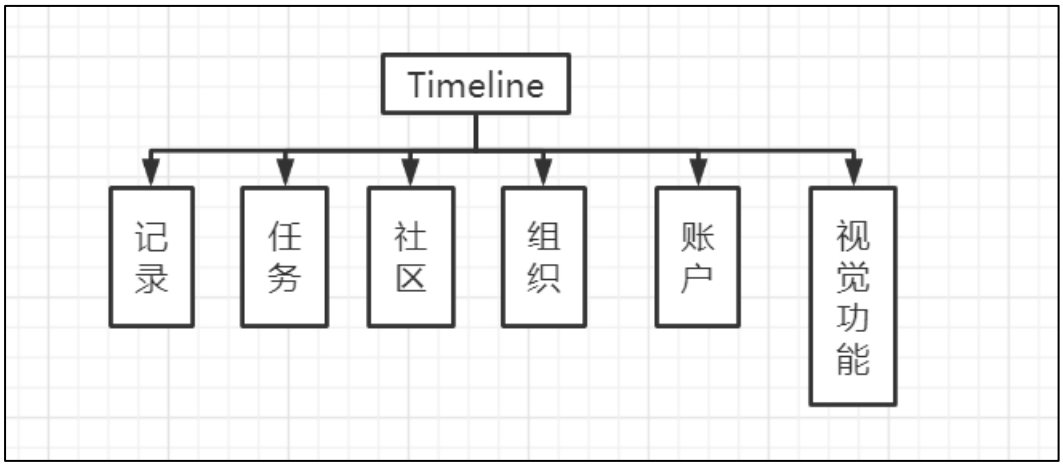
架构说明如下：

1. 表示层：网页端的表示层主要为使用 Vue.js 制作的界面，用户可以通过浏览器进行访问，访问时表示层向控制层传送参数，将信息传递给控制层。业务逻辑处理后，处理结果将返回给表示层进行展示，实现交互。
2. 控制层：控制层接受来自表示层的信息和请求，将请求分发到业务层不同的模块进行处理，处理后返回结果。同时控制层也承担着对表示层访问请求的分流工作。
3. 业务层：业务层在控制层和持久层中间，调用持久层，又被控制层调用，起一个承上启下的作用，非常关键。系统的结构耦合度较弱，层与层之间的依赖是向下的，上层设计的改变对底层的调用不会有影响。
4. 持久层：持久层采用 Mybatis 框架。Mybatis 提供强大的动态 SQL 功能，将 JDBC 接口封装起来，并通过 mapper 配置文件与数据库相连接，利用 Java 的注解和映射功能进行数据库实例化。Mybatis 可以通过 SqlSessionFactory 对象生成 SqlSession 实例，执行 mapper 中的 SQL

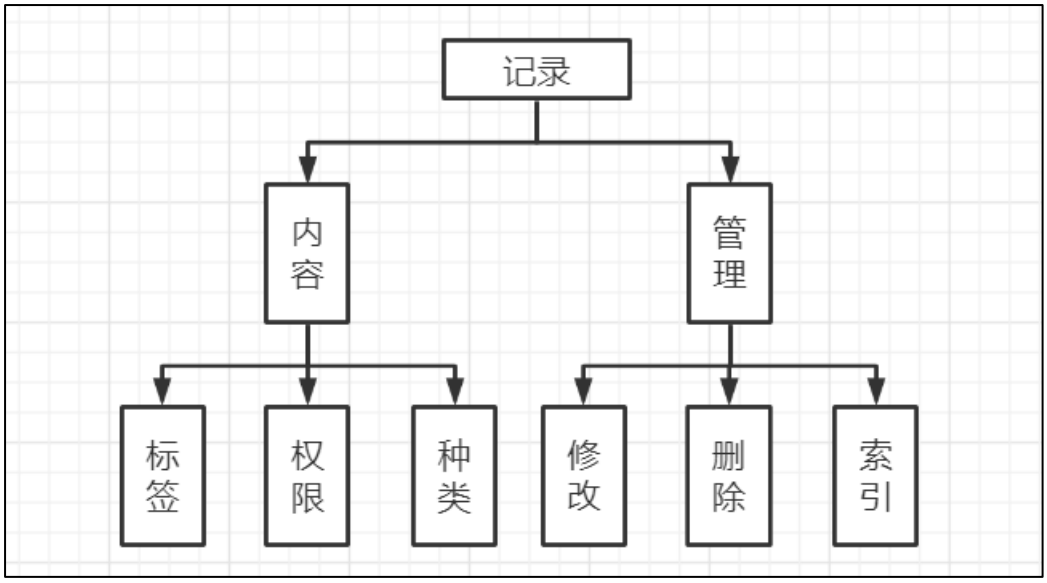
语句。SqlSessionFactory 对象可以通过 XML 配置文件生成或通过 Mybatis 的 Resource 对象构建。

2.2 功能模块设计

根据需求分析，将系统前台分为六个大模块：记录功能、任务功能、社区功能、组织功能、账户相关、视觉功能。功能模块图下图所示。



2.2.1 记录模块



内容

种类：可以上传文字描述，图片和音频等多种形式记录，VIP 用户有更多使用权限。

权限：用户可以设置记录查看的权限。

标签：每条记录都会有标签，包含心情、地点等种类，用于事件索引。

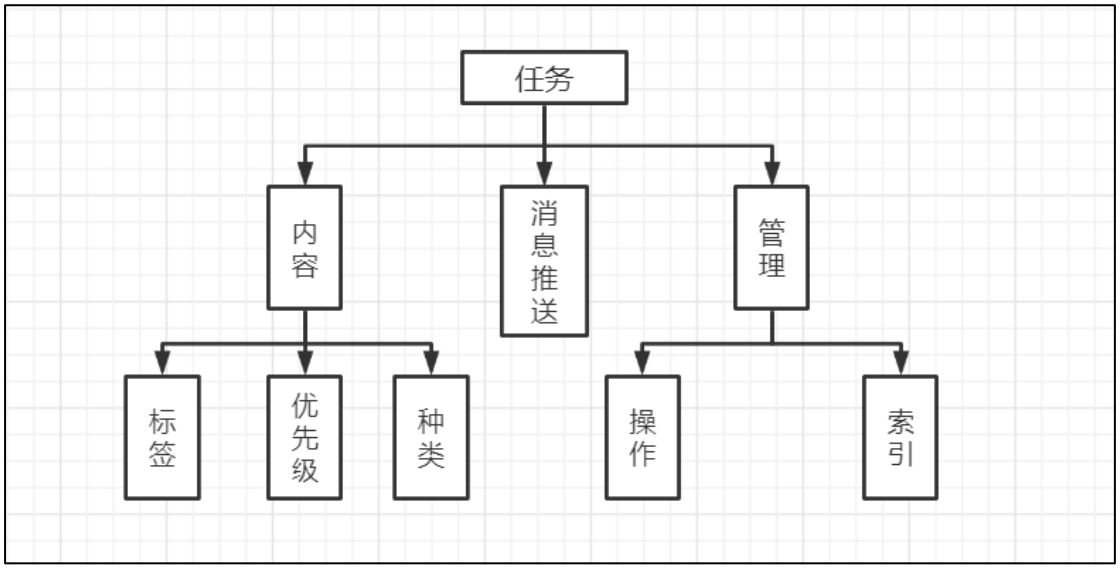
其他：记录标题、记录时间点等通用内容。

管理

修改/删除：用户可以修改删除已创建的记录。

索引：可根据时间，关键字，标签等进行记录索引。

2.2.2 任务模块



内容

标签：对内容优先级或者地点等特征值描述，可以用来建立索引。

优先级：任务的优先级。

种类：不同类型的任务，有多种分类标准，如重复与否等。

其他：任务标题，任务时间地点等相关内容。

消息推送

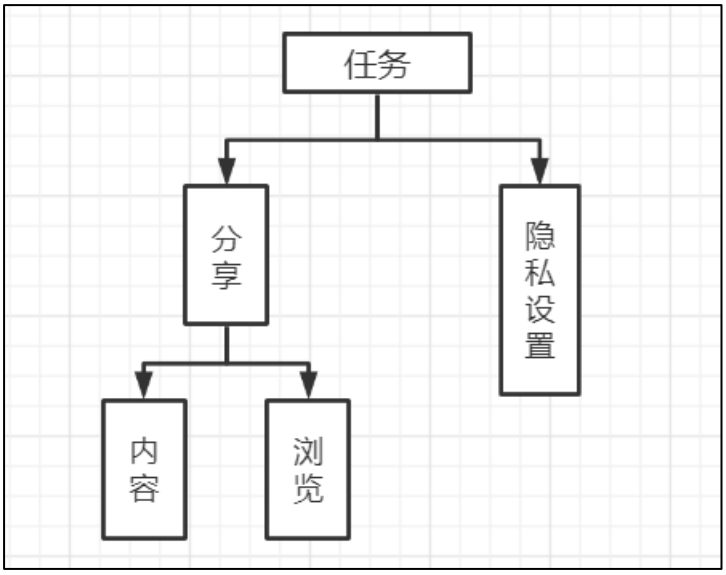
用户可以开启消息推送以便及时获得任务进度的通知。

管理

操作：对任务创建/修改/删除等操作。

索引：用户可以根据索引来寻找到特定范围的内容，例如限制了时间、地点等范围。

2.2.3 社区模块



分享：

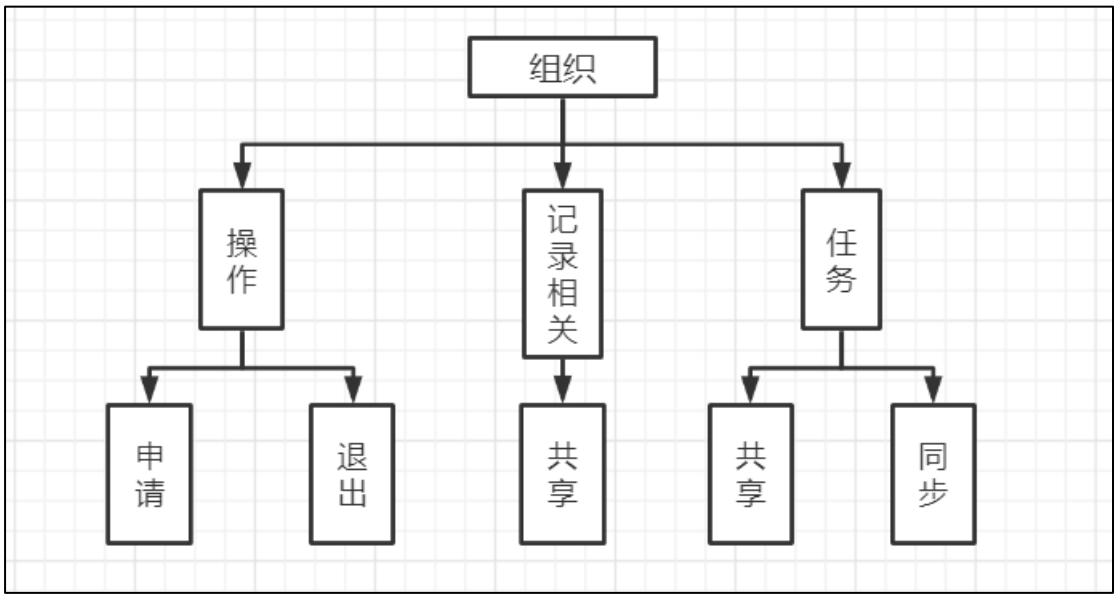
内容：用户可以分享自己的任务和记录等多种内容到社区。

浏览：用户可以浏览查看其他用户分享到社区的内容。

隐私：

用户可以设置自己的分享内容查看权限，分为开启、限制和关闭三个等级。

2.2.4 组织模块



操作

申请：用户可以申请加入组织。

退出：用户可以退出已经加入的组织。

记录相关

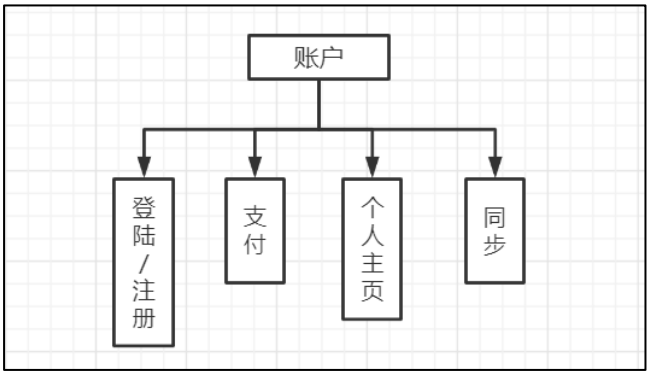
用户可以共享自己的记录，组内其他用户可以查看。

任务相关

共享：用户可以向组内成员共享自己的任务。

同步：小组发布的任务可以同步到每个小组成员的任务列表（时间线）中。

2.2.5 账户相关



登录\注册：

用户使用注册的账户登录到软件;用户使用自己的个人信息进行帐号注册。

支付：

对于一些付费的功能，用户可以通过付费对这些功能进行解锁。

个人主页：

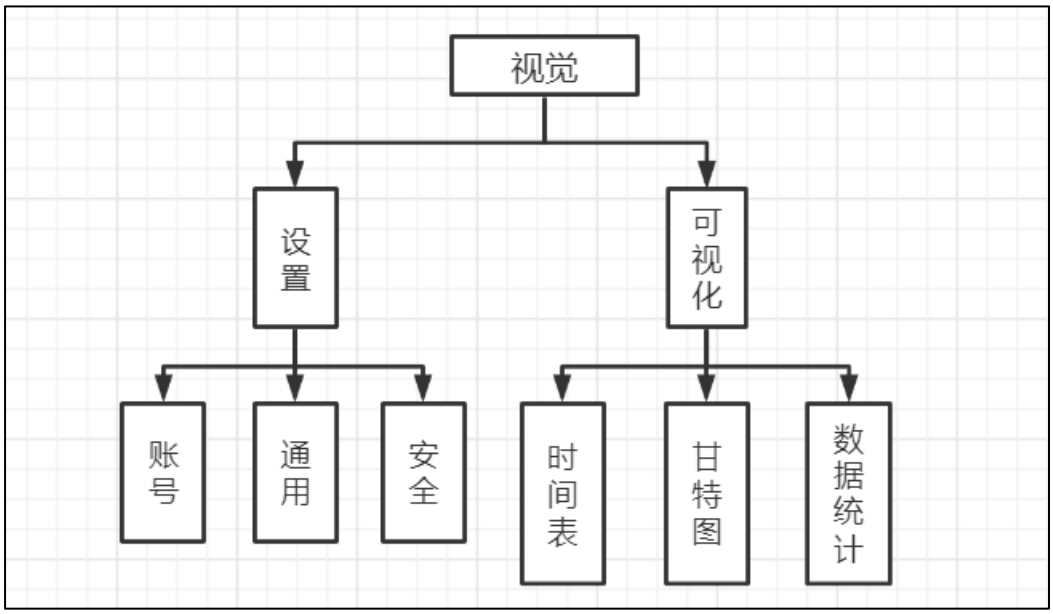
浏览：对用户个人主页上的信息进行浏览和查看。

修改：对用户个人主页上的信息进行修改。

多终端同步：

将本地的时间线将自动与云端同步，保证本地与云端数据一致。

2.2.6 视觉相关



设置：

账号：用户可以进行账号相关的设置，同时可以进行登陆退出等操作。

安全：用户进行安全性相关的设置，如设置软件权限等。

通用：其他通用设置。

可视化：

时间表：根据用户输入的任务生成包含时间+任务简要描述的时间表，单位为每日/每周。生成时间表后，用户可以按照不同的时间尺度查看。如，用户的任务或记录可按日/月/年为单位生成缩略图。

甘特图：根据用户制定的任务安排生成甘特图，利于用户根据甘特图查看自己的进度，并安排管理自己的时间。

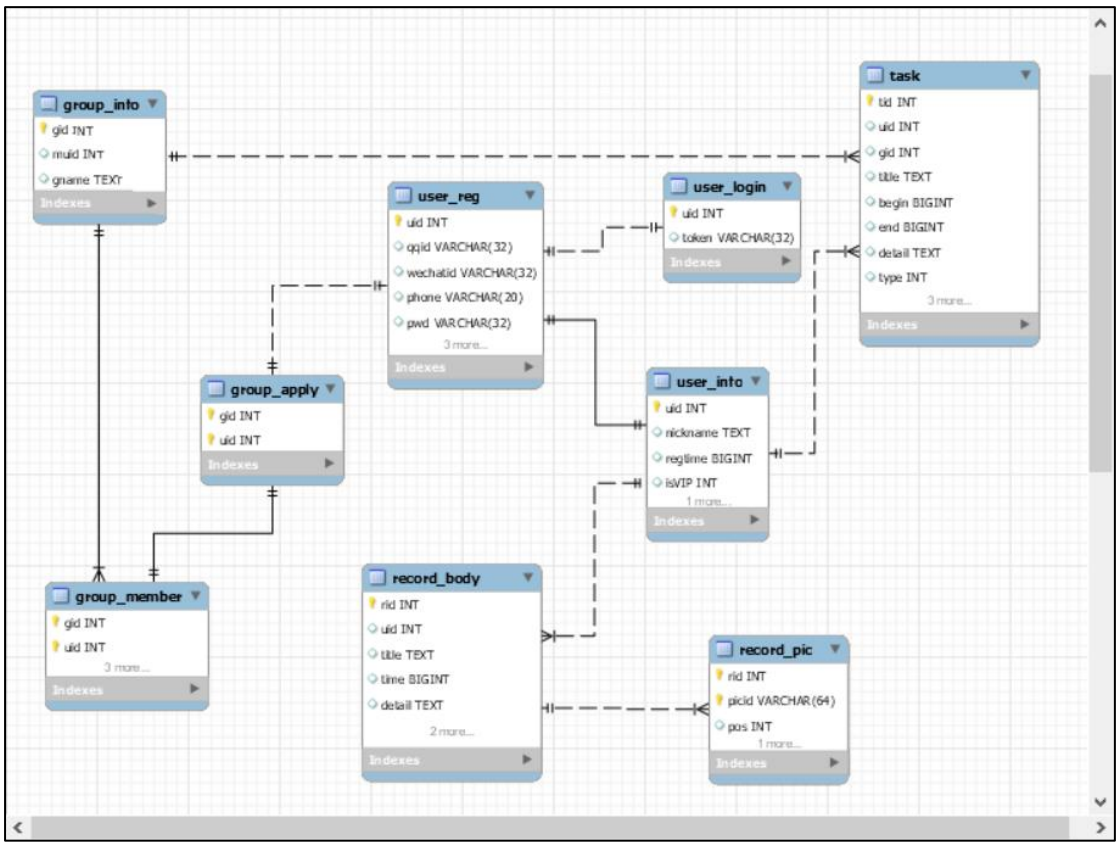
数据统计：对用户的任务/记录做数据统计。如该用户一年中心情统计是怎样的，一年中有多少天是开心的或悲伤的等。

2.3 数据库设计

2.3.1 概念结构

构成 E-R 图的 3 个基本要素是实体、属性和联系。实体之间有三种关系：一 对一、一对多、多对多。实体与实体之间的联系也可以有属性。

通过对系统的分析，抽象出的实体主要有：用户、任务、记录和组等。各实体之间的 E-R 关系图如图所示：（可能待调整）



2.3.2 逻辑结构

根据系统需求和功能模块设计以及数据库概念模型 E-R 图的设计，共设计了 9 张数据基本表。

后续还会对系统进行优化，开发更多新模块，因此在表的设计中需要保证未来可拓展性。 各模块的关系模型如下：

(1) 注册用户表（用户 ID，QQ 号，微信号，手机号，密码）

(2) 在线用户表（用户 ID，对应 token）

(3) 用户信息表（用户 ID，昵称，注册时间，是否 VIP）

(4) 任务表（任务 ID，用户 ID，组 ID，任务标题，任务开始时间，任务结束时间，任务描述，任务重复类别）

(2) 记录体表（记录 ID，用户 ID，记录标题，记录时间，记录描述）

(6) 记录图片表（记录 ID，图片 ID，图片排序位置）

(7) 组申请记录表（组 ID，用户 ID）

(8) 组成员表（组 ID，用户 ID）

(9) 组信息表（组 ID，组成员 ID，组名）

以上数据表中，各表主键均已在途中用金色钥匙🔑符号标出，由于外键的约束会导致复杂的 删除问题，降低效率，所以在数据表的设计中没有设置外键。

2.3.3 物理结构

各类表具体信息见下。

注册用户表

列名	类型	描述
uid	INT	用户 ID
qqid	VARCHAR(32)	用户注册使用的 QQ 号
wechatid	VARCHAR(32)	用户注册使用的微信号
phone	VARCHAR(20)	用户注册使用的手机号
pwd	VARCHAR(32)	用户的注册密码

在线用户表

列名	类型	描述
uid	INT	用户 ID

列名	类型	描述
token	VARCHAR(32)	和 uid 一起做身份验证

用户信息表

列名	类型	描述
uid	INT	用户 ID
nickname	INT	用户昵称
regtime	BIGINT	用户注册时间
isVIP	INT	是否为 VIP

任务表

列名	类型	描述
tid	INT	任务 ID
uid	INT	用户 ID
gid	INT	组 ID（-1 代表非组任务）
title	TEXT	任务标题
begin	BIGINT	任务开始时间
end	BIGINT	任务结束时间
detail	TEXT	任务描述
type	INT	任务重复类型（每日每周等）

记录体表

列名	类型	描述
rid	INT	记录 ID
uid	INT	用户 ID
title	TEXT	记录标题
time	BIGINT	记录创建的时间
detail	TEXT	记录描述

记录图片表

列名	类型	描述
rid	INT	记录 ID
picid	VARCHAR(64)	图片 ID
pos	INT	图片排序位置

组申请记录表

列名	类型	描述
gid	INT	组 ID
uid	INT	用户 ID

组成员表

列名	类型	描述
gid	INT	组 ID
uid	INT	用户 ID

组信息表

列名	类型	描述
gid	INT	组 ID
muid	INT	组成员用户 ID
gname	TEXT	组名

3 系统详细设计

3.1 方案总览

3.1.1 前台要求

地址	名称	描述
/login	登录页	登录/注册/找回密码
/home	框架页	导航/用户基本信息/边栏
/home/info	个人信息	个人信息详细页
/home/time	时间线	记录任务可视化/添加/删除/索引
/home/list	列表	记录任务/添加/删除/索引
/home/group	组织表	浏览管理组织
/home/analize	数据分析	数据分析/社交平台分享
/recinfo [pop]	记录信息	记录详细信息查看
/taskinfo [pop]	任务信息	任务详细信息查看
/groupinfo [pop]	组织信息	组织详细信息查看

```
--注:[pop]为覆盖层页面
```

登录页

```
登录：社交登录，手机密码登录，手机验证码登录
注册：手机密码注册，应接收验证码（注，社交平台直接登录即可）
找回密码：手机+验证码
```

框架页

```
为主功能页面提供基本框架
通俗来讲是个功能选择器，子功能套在里面
有五个子功能的入口，有>80%的区域显示相应功能
```

顶栏显示用户名，点击后有下拉聊表框，应提供退出功能

个人信息页

展示详细个人信息，可修改

时间线

展示用户时间线，事件和记录以块的形式显示在线上，可点击直接进入详情
可切换尺度（日，周，月，年）
可添加记录
可添加事件
有索引功能（文字，标签，时间等）

列表

将事件和记录以列表的形式呈现，是 2.4 的文字版，功能一致。

组织表

与用户相关的组织列表，包含我管理的和我加入的两部分
可创建组织（仅询问组织名即可）
可加入组织
点击具体组织后进入组织详情页

数据分析

呈现数据分析结果
提供社交分享入口

记录信息[pop]

查看某记录的详细信息 或 添加新纪录
记录的共享开关在此，开启后查看位置在此

编辑记录
删除记录

任务信息[pop]

查看某任务的详细信息 或 添加新任务
个人任务与组织任务
编辑任务（如果可以）
删除任务（如果可以）

组织信息[pop]

成员列表
组织信息查看与修改
组织任务列表
发布任务（仅创建者）
点击任务列表中的人物 pop 任务详细信息，在里面可修改删除
申请加入者列表（仅创建者）

3.1.2 后台要求

用户相关

地址	名称
/user/login	登录
/user/logout	登出
/user/reg	注册
/user/phone	发送验证码
/user/info	用户信息
/user/infoupd	修改用户信息

任务相关

地址	名称
/task/list	任务列表
/task/view	查看单个任务

地址	名称
/task/upd	修改任务
/task/add	添加任务
/task/del	删除任务

记录相关

地址	名称
/rcd/list	记录列表
/rcd/view	查看单个记录
/rcd/upd	修改记录
/rcd/add	添加记录
/rcd/del	删除记录
/rcd/picupload	上传图片

搜索相关

地址	名称
/search	联合搜索

组织相关

地址	名称
/group/list	组列表
/group/make	创建组
/group/del	删除组
/group/join	申请加入组
/group/manage	加人/T 人

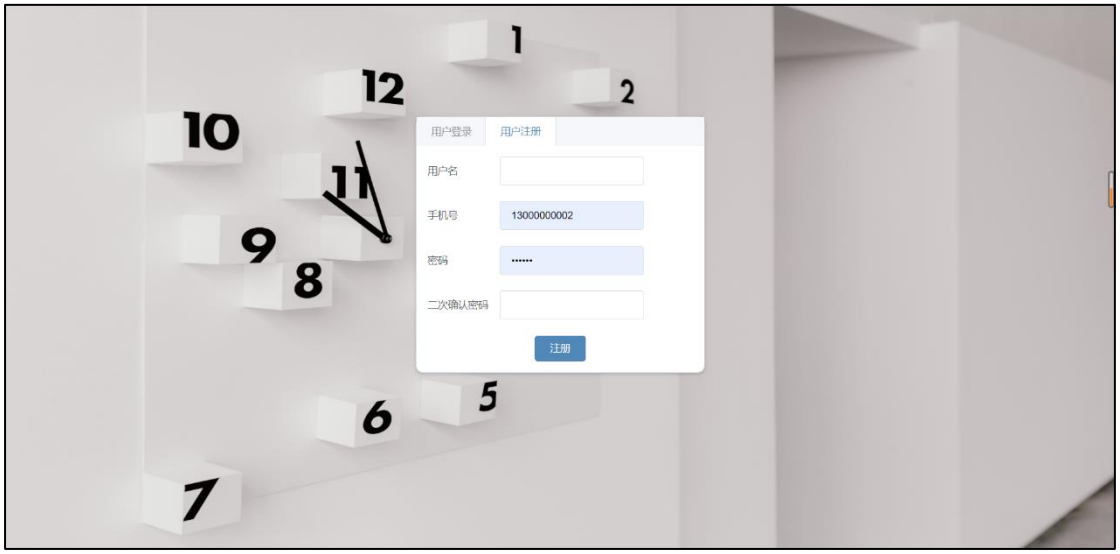
地址	名称
/group/info	组信息
/group/upd	修改组信息

3.2 前端

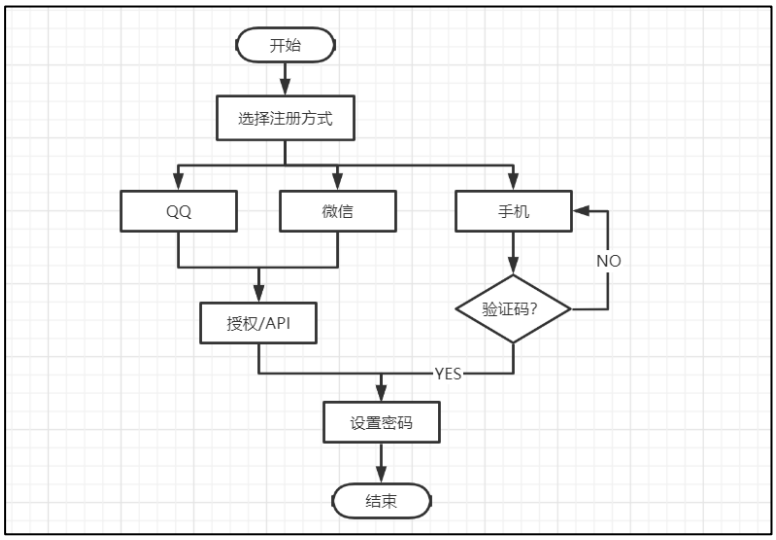
3.2.1 用户相关

3.2.1.1 注册页

界面



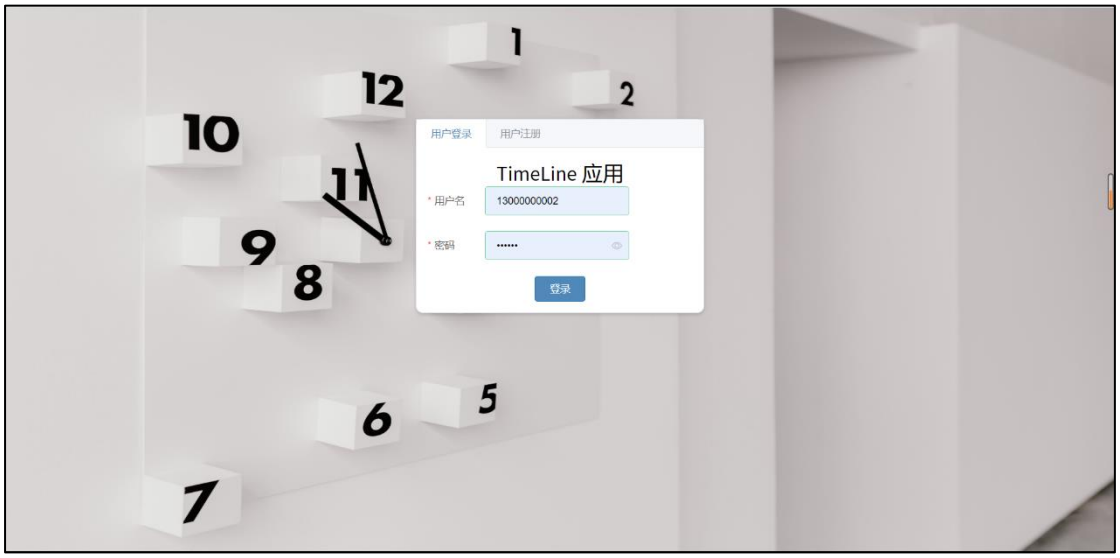
业务逻辑流程图



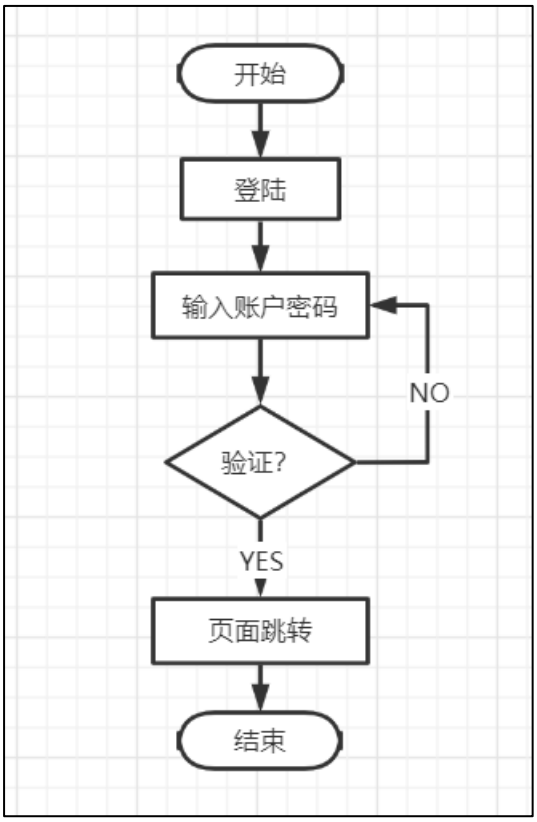
用户可以选择 QQ、微信和手机三种注册方式，选择前两种会调用官方 API 进行登录授权，选择手机注册，后台会发送验证码，验证正确后即可设置账户密码，注册成功。

3.2.1.2 登录页

界面



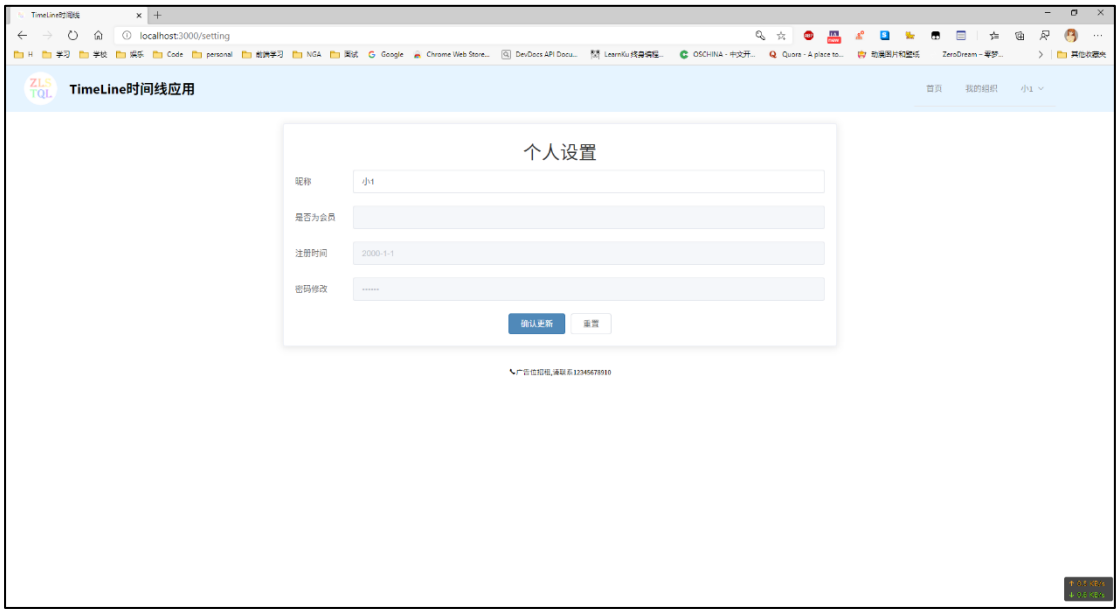
业务逻辑流程图



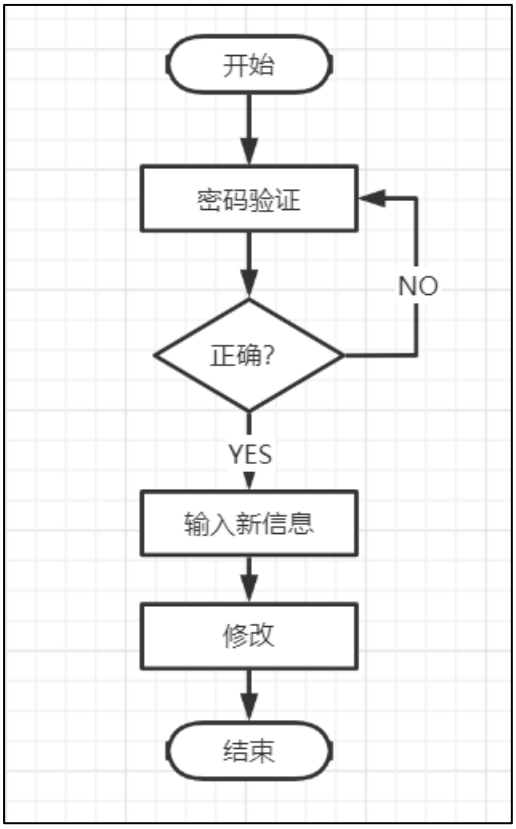
用户进入登陆页面，输入账户密码，前端会请求后端查询已注册用户的信息并进行账户密码验证，成功后登陆，跳转至主页。

3.2.1.3 个人设置页

界面



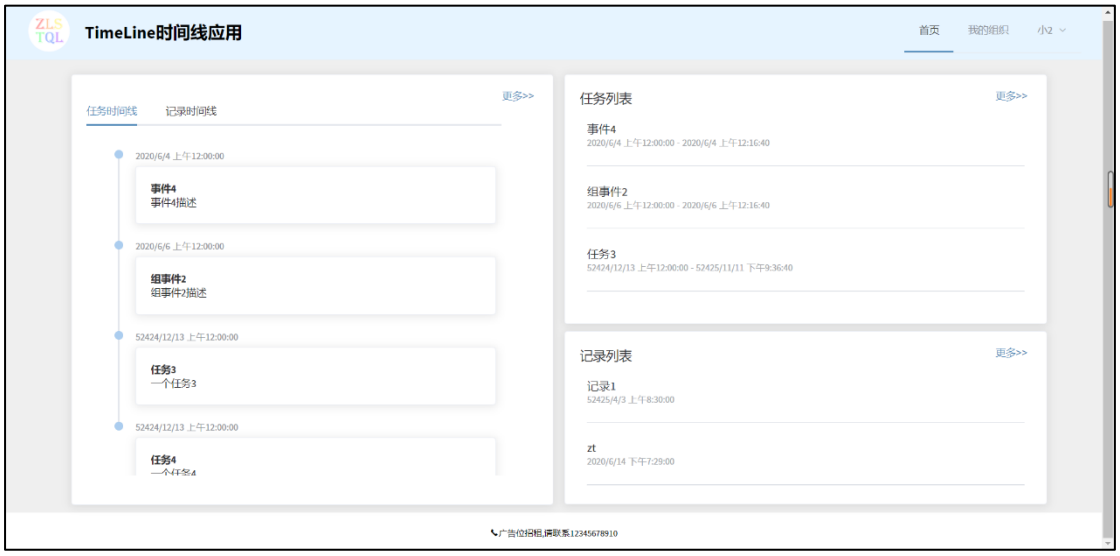
业务逻辑流程图

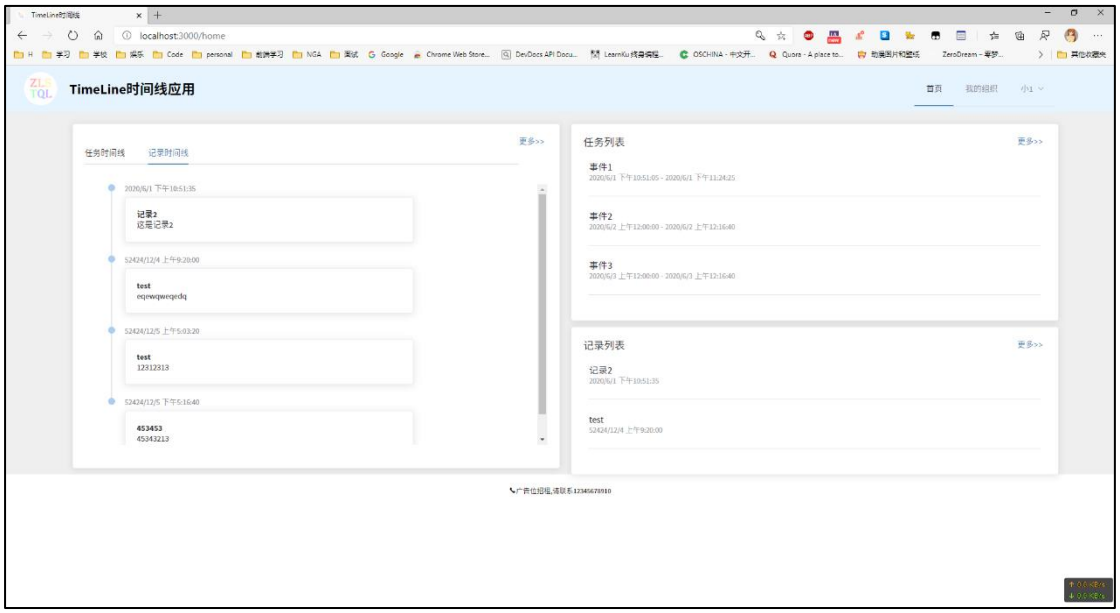


用于可以进行个性化个人信息设置，输入密码验证成功后，输入新的个人信息，前端会将数据加密传输至后端，并写入数据库。

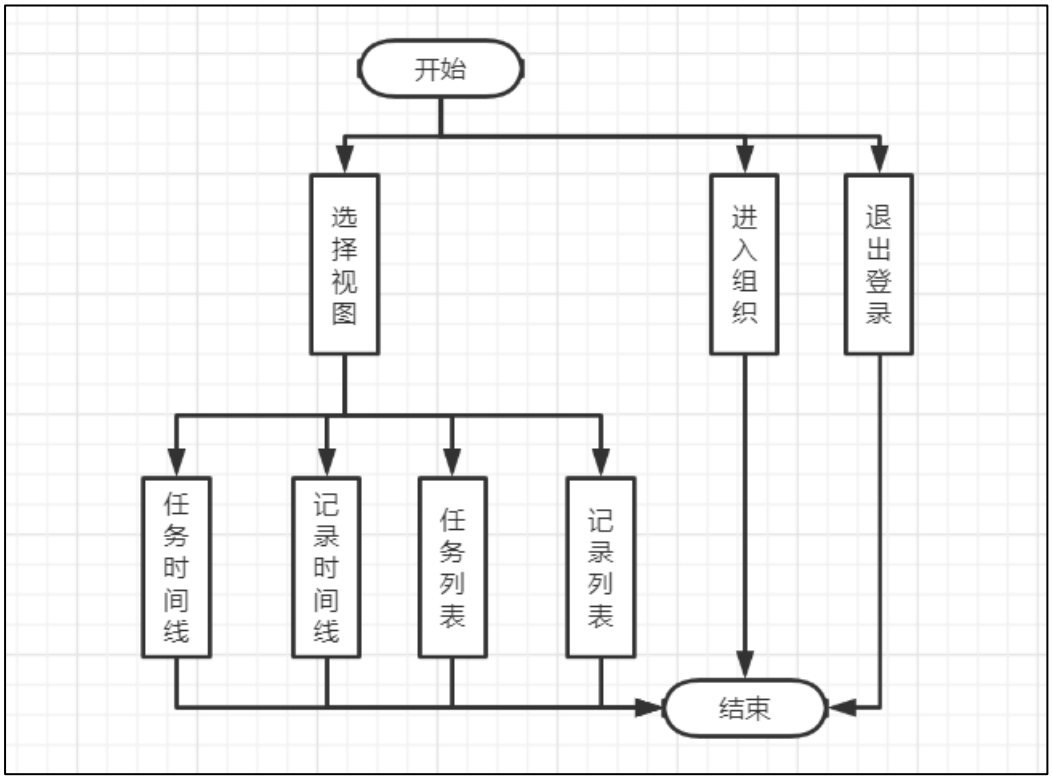
3.2.2 首页

界面





业务逻辑流程图

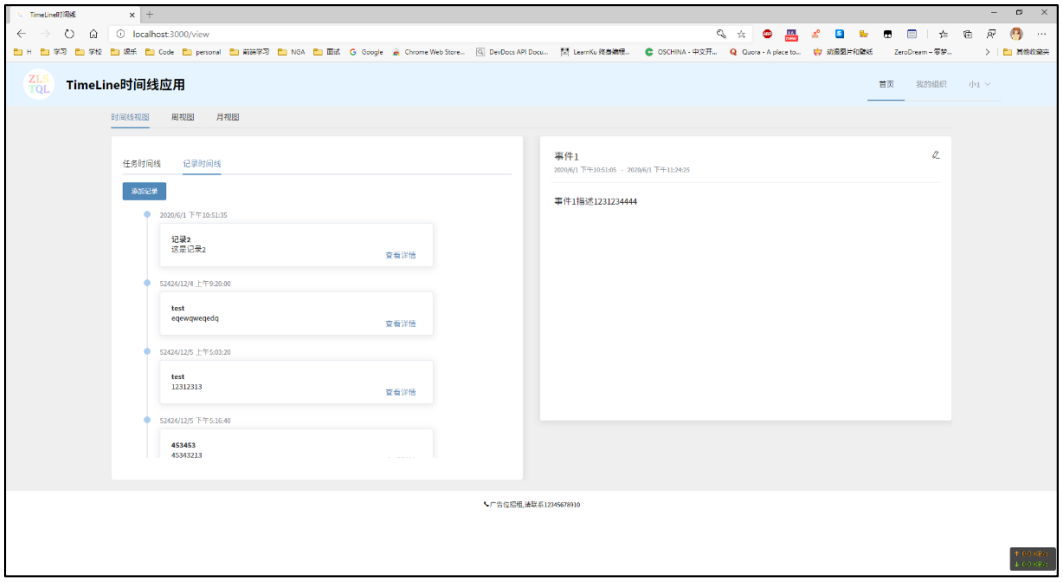


用户从首页可以选择进入任务时间线视图，列表视图，也可以进入组织视图，或者退出登录。

3.2.3 视图类

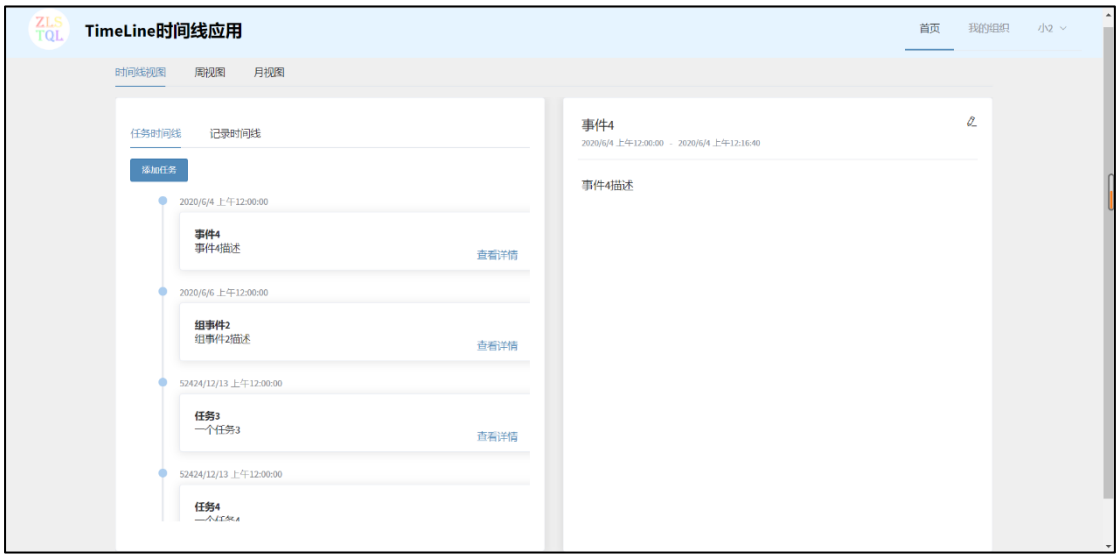
3.2.3.1 记录时间线视图

界面

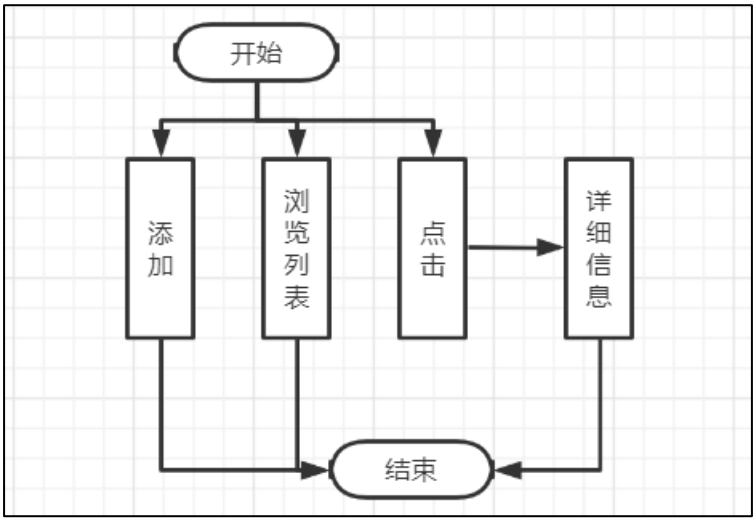


3.2.3.2 任务时间线视图

界面



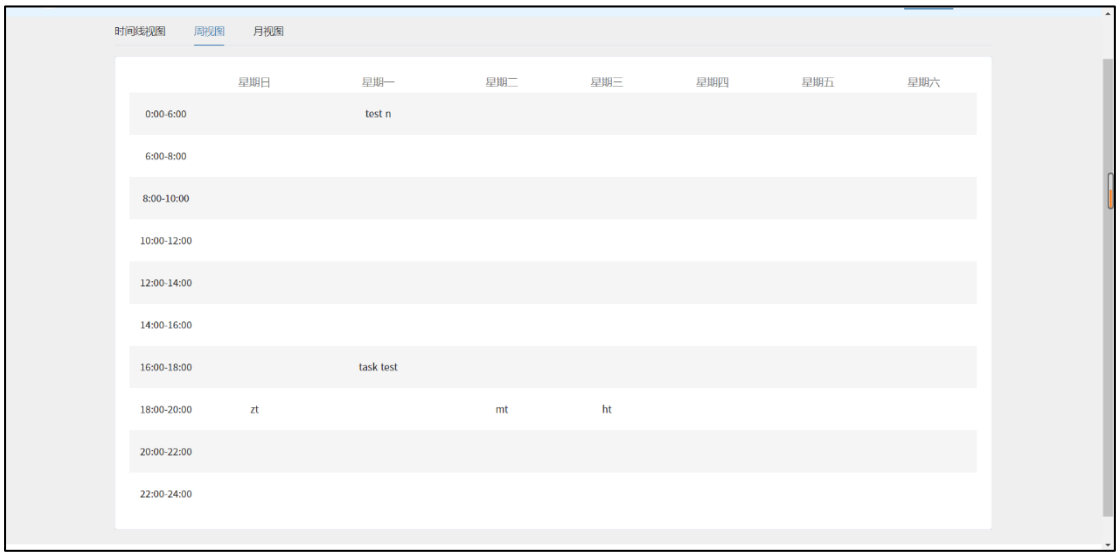
时间线业务逻辑流程图



在时间线视图，用户可以浏览列表中已创建的记录/任务，可以添加记录/任务，也可以点击具体的任务/记录从而查看其详细信息。

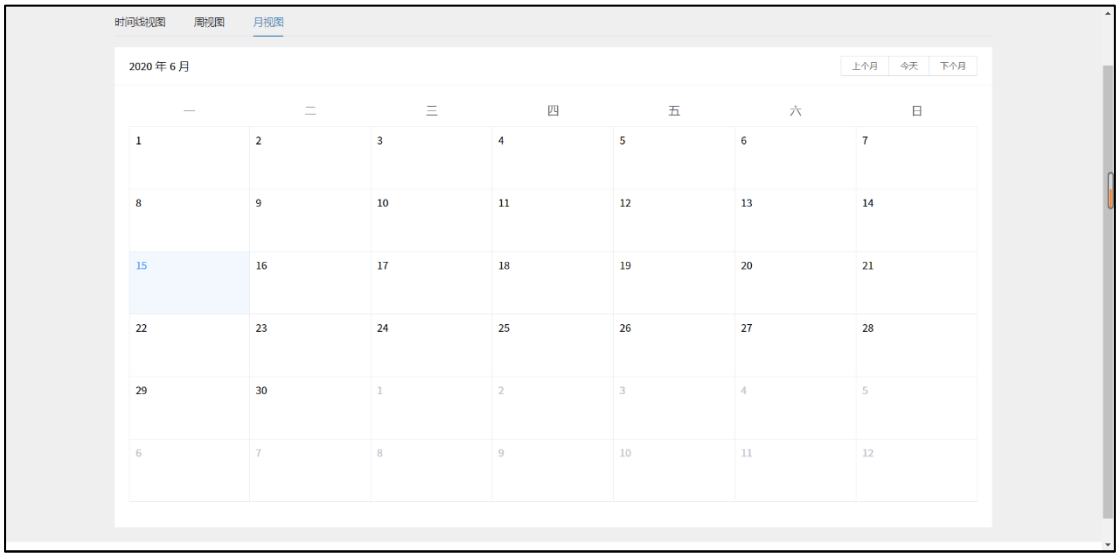
3.2.3.3 周视图

界面

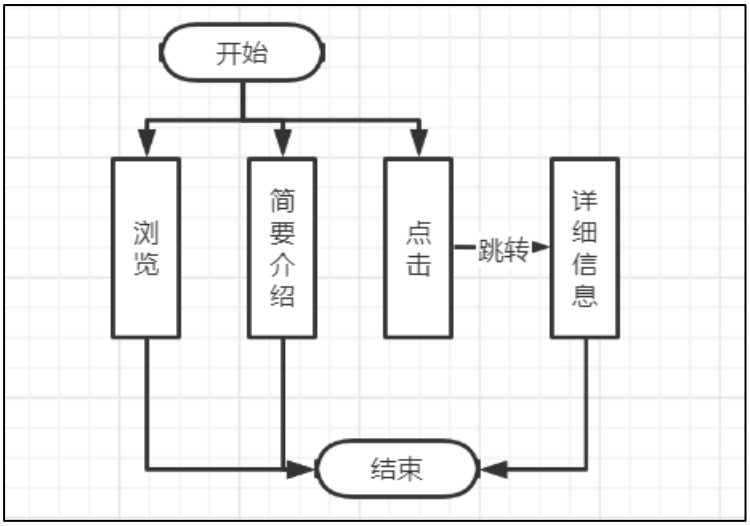


3.2.3.4 月视图

界面



时间视图业务逻辑流程图

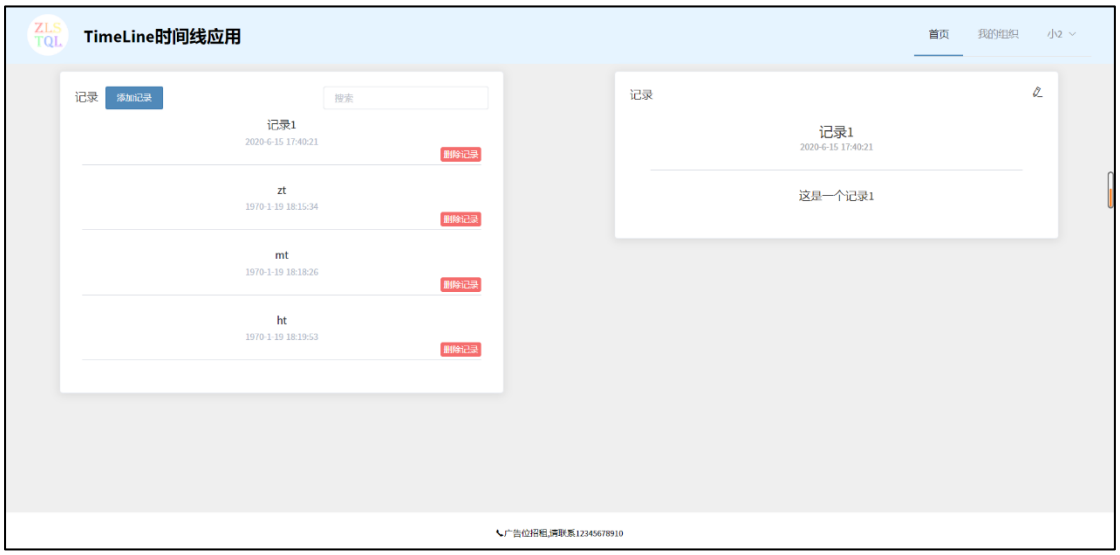


在月/周试图内，用户可以清楚地按照时间单位浏览已经创建的任务/记录，并可以看到简要介绍，点击后跳转至相关页面（任务/记录列表视图）进行详细信息查看。

3.2.4 记录相关

3.2.4.1 记录列表页

界面

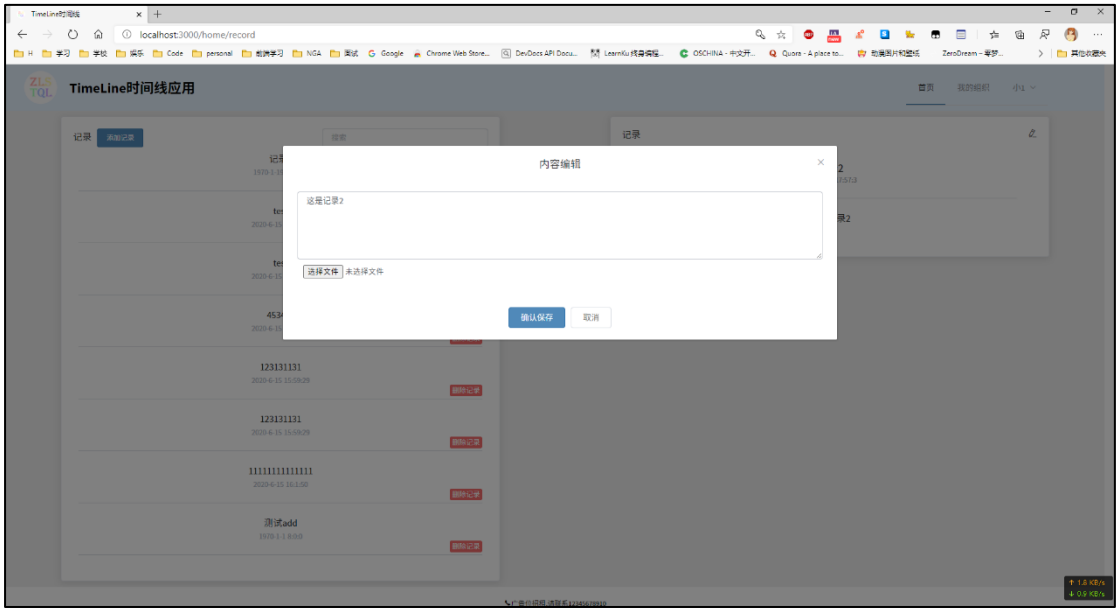


3.2.4.2 添加记录页

界面

3.2.4.3 修改记录页

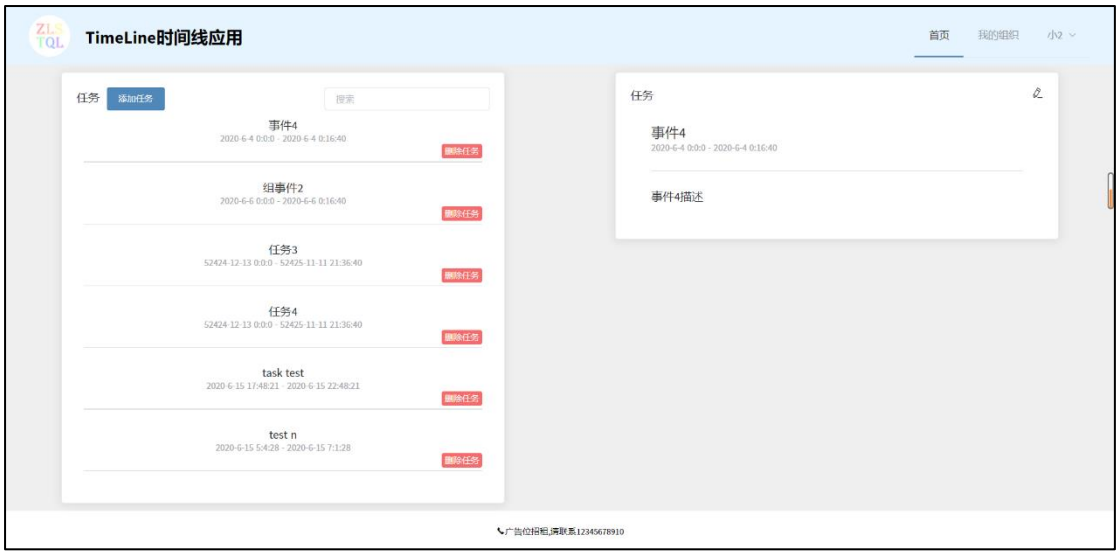
界面



3.2.5 任务相关

3.2.5.1 任务列表页

界面

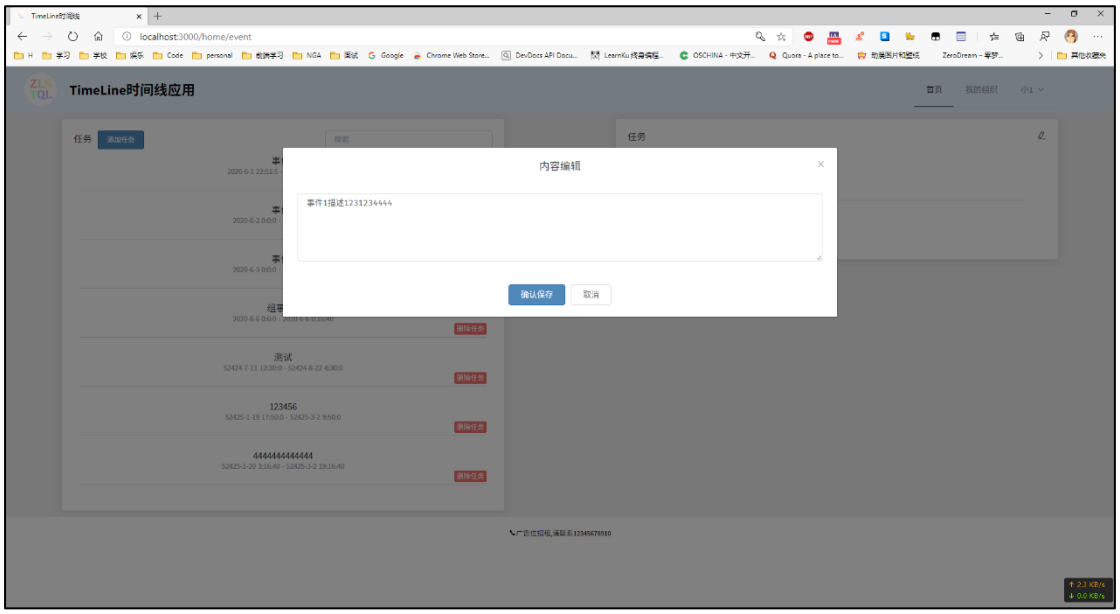


3.2.5.2 添加任务页

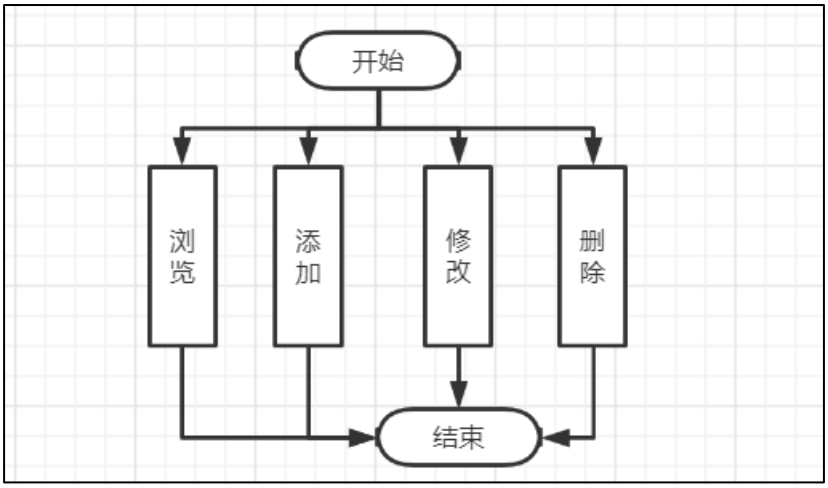
界面

3.2.5.3 修改任务页

界面



记录&任务相关业务逻辑流程图

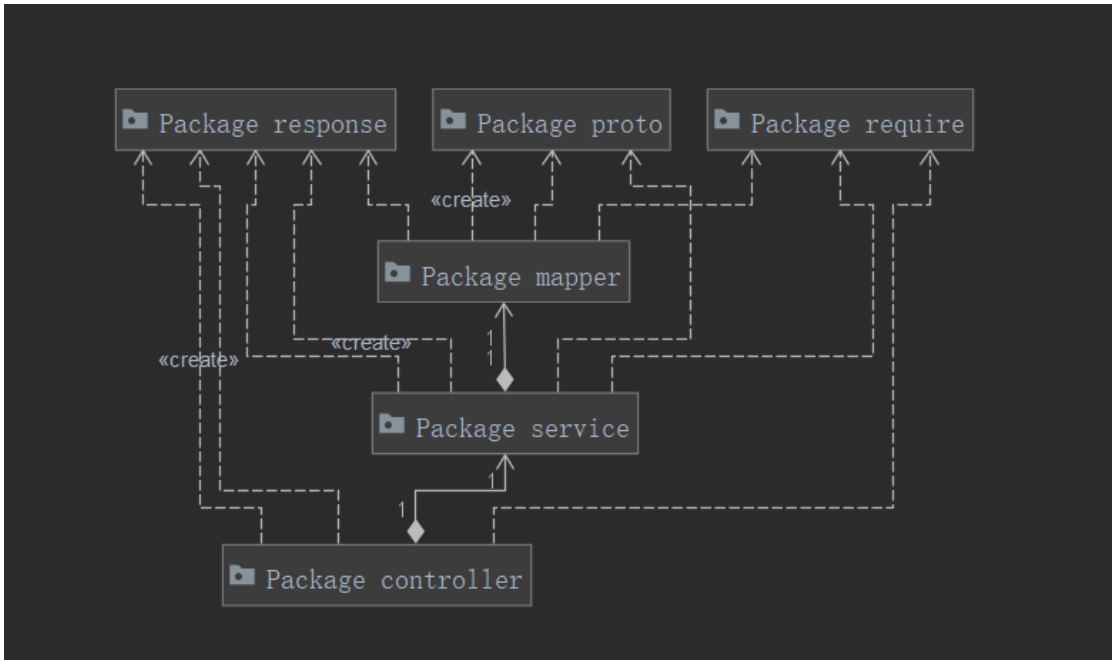


在任务/记录列表视图内，用户可以统一浏览已经创建的任务/记录，并可以进行添加、修改或删除等操作。

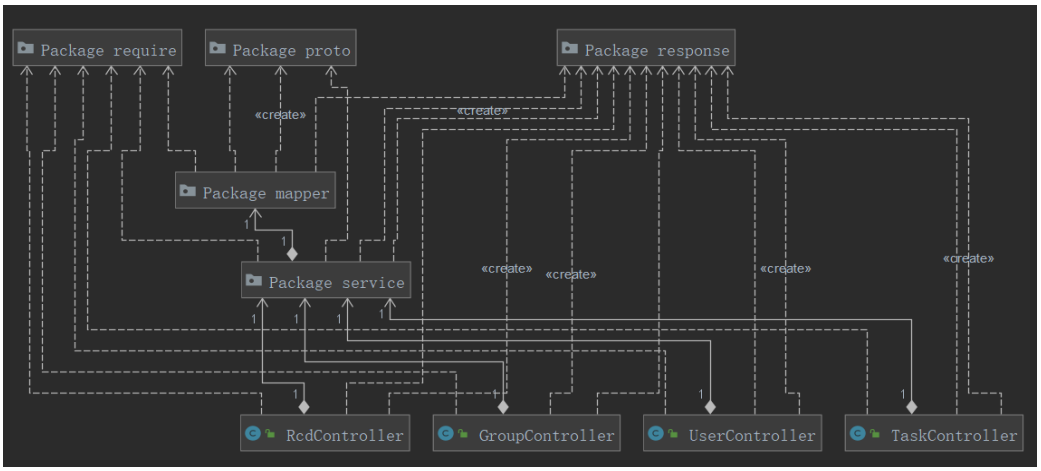
3.3 后端

3.3.1 总览

后端采用该 Spring Boot 框架，MySQL 作为数据库存储。包之间的依赖关系见下图。Controller 层负责接收请求，Service 层对接受的请求进行逻辑处理，并调用 Dao 层（mapper）进行数据相关的操作。



3.3.2Controller 层



UserController：负责接收来自前端的用户相关的请求。

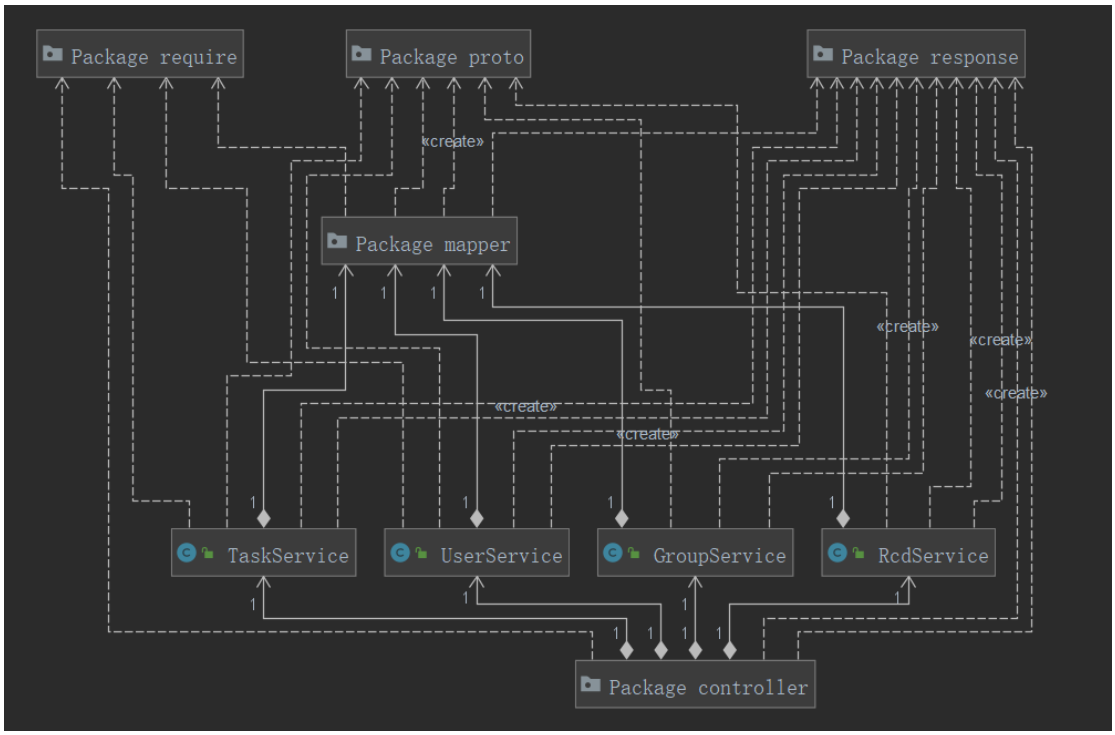
TaskController：负责接收来自前端的任务相关的请求。

RcdController：负责接收来自前端的记录相关的请求。

GroupController：负责接收来自前端的组织相关的请求。

相关的请求连接请见 3.1.2 后台要求

3.3.3Service 层



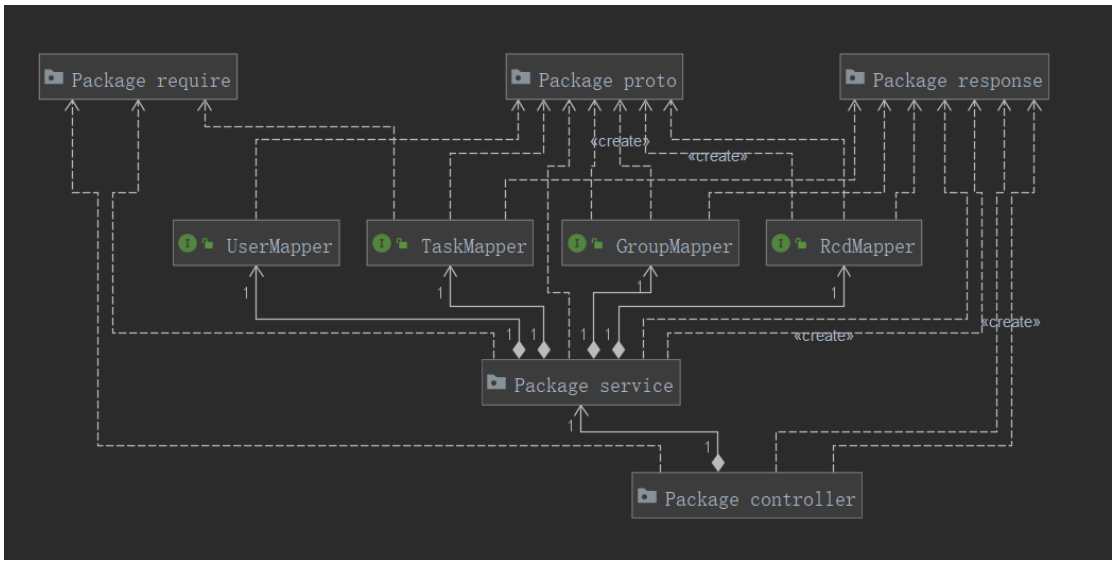
UserService：负责对来自前端的用户相关的请求进行逻辑处理。

TaskService：负责对来自前端的任务相关的请求进行逻辑处理。

RcdService：负责对来自前端的记录相关的请求进行逻辑处理。

GroupService：负责对来自前端的组织相关的请求进行逻辑处理。

3.3.4Dao 层



UserMapper：负责对来自前端的用户相关的请求进行数据库操作。

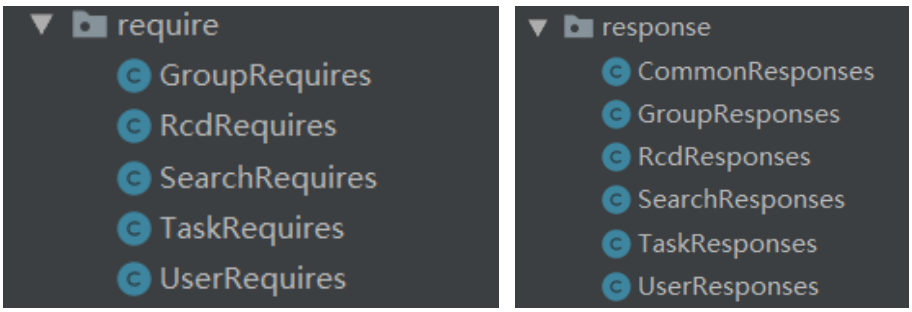
TaskMapper：负责对来自前端的任务相关的请求进行数据库操作。

RcdMapper：负责对来自前端的记录相关的请求进行数据库操作。

GroupMapper：负责对来自前端的组织相关的请求进行数据库操作。

3.4API

系统对来自前端的请求和产生的响应进行了抽离封装。



详细的接口设计请见接口文档“api.md".

4 系统测试

在完成了系统功能开发之后，为了在产品上线前消除潜在的 bug，我们需要根据系统测试理论对系统的相关功能和性能指标进行测试。系统测试是将已

经集成好的软件系统，作为计算机系统的一个元素，与计算机硬件、某些支持软件、数据和人员等其他系统元素结合在一起，在实际运行环境下，对计算机系统进行一系列的集成测试和确认测试。

因为时间与成本原因，本项目只融合了黑盒与白盒测试技术，对系统进行了简单必要的测试。预期结果为通过的测试用例均已通过。以下列出了预期结果为不通过的功能测试样例，即可能会导致系统出错的样例。性能测试均已通过。

4.1 用户相关

注册

测试用例	预期结果	结果
无效手机号	注册失败	通过
错误验证码	注册失败	通过

登陆

测试用例	预期结果	结果
正确账号，错误密码	登陆失败	通过
错误账号，正确密码	登陆失败	通过
错误账号，错误密码	登陆失败	通过

登出

测试用例	预期结果	结果
错误 uid(不在线)	登出失败	通过
错误 token	登出失败	通过

4.2 任务相关

操作（创建/修改/删除）

测试用例	预期结果	结果
过长标题	操作失败	通过
乱码内容	操作失败	通过
未来时间	操作失败	通过

消息推送

测试用例	预期结果	结果
开始时间阶梯测试	定时提醒	通过
结束时间接替测试	定时提醒	通过

4.3 记录相关

内容

测试用例	预期结果	结果
过长标题	内容操作失败	通过
过大图片	内容操作失败	通过
多种格式图片	内容操作失败	通过
乱码内容	内容操作失败	通过

4.4 组织相关

操作（创建/删除组，申请/T 人等）

测试用例	预期结果	结果
创建大于人数限制的组	创建失败	通过
T 出无效用户	T 人失败	通过
申请加入不存在的组	申请失败	通过

组信息修改

测试用例	预期结果	结果
过长组名	修改失败	通过
组成员 ID 修改	修改失败	通过

4.5 其他

搜索

测试用例	预期结果	结果
过长搜索关键字	无法搜索	通过
无效关键字	无法搜索	通过

共享

测试用例	预期结果	结果
隐私模式共享	共享失败	通过
查看无权限信息	查看失败	通过

同步

测试用例	预期结果	结果
不支持机型	同步失败	通过
过低版本	同步失败	通过
断网同步	同步失败	通过
不属于组的用户同步	同步失败	通过

可视化

测试用例	预期结果	结果
空数据	无法可视化	通过
错误范围限定	无法可视化	通过

4.6 性能测试

测试描述	期望标准	测试结果
用户同时操作	支持	通过
处理的文件数和记录数	10000	未知
数据库大小限制	不超过硬盘容量	通过
单条记录大小限制	不超过数据库大小限制	通过
存取单条记录时间限制	1s	通过
按照索引获取记录时间限制	1s	通过