# **星云6项目管理系统**概要设计文档

## 概述

## 系统简述

本系统是一个运行在Web端的项目管理系统，内容包括项目进度的监控管理、人员角色的分配、相关文档的分类及存储、任务日程的进度管理与审核、成员之间在线交流等。

## 软件设计目标

本项目目标是构建上述的项目管理系统，能够满足小型团队（10人以下）的项目管理需求。

## 参考资料

## 修订版本记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **修订内容** | **版本** | **修订人** | **修订日期** |
| 1 | 创建本文档框架，编写系统简述和设计目标 | 0.1 | 利俊安 | 2019/9/8 |
| 2 | 编写系统逻辑架构内容 | 0.2 | 利俊安 | 2019/9/9 |
| 3 | 编写系统架构设计和接口设计内容 | 1.0 | 利俊安 | 2019/9/25 |
| 4 | 更新系统架构设计内容 | 1.1 | 利俊安 | 2019/9/25 |
| 5 | 更新系统物理架构内容 | 1.2 | 利俊安 | 2019/10/8 |
| 6 | 编写非功能设计内容 | 1.3 | 利俊安 | 2019/10/14 |

## 术语表

|  |  |  |
| --- | --- | --- |
| **术语** | **解释** | **备注** |
| **项目** | 软件工程中的一个项目，由一个项目经理和若干个普通成员组成。 | 项目是本项目的基本单位 |
| **任务** | 项目中包含的若干任务，任务由项目经理布置和指派人员。任务分为4层，最后一层无法再包含子任务。 |  |
| **进度请求** | 一个项目的推进由成员发起进度请求，项目经理批准后才生效。项目经理亦可直接更改任务的完成状态，不经过进度请求步骤。 |  |

## 设计概述

## E:\Study\项目管理\项目管理系统\最终文档\QQ图片20191127080240.pngQQ图片20191127080240系统结构设计

## 系统逻辑架构

**前端逻辑架构：**

**网络服务层：**前端中将网络处理方面的函数封装成全局类形成网络交互层，用于向后端发起网络请求以及对响应进行通用处理，将注入到业务服务层中的服务中。包括HTTP服务和WebSocket服务，处理不同类型的网络请求；

**业务服务层：**在前端中声明所需的数据结构，主要用于储存和转化后台响应得到的数据；声明一些服务类来组织和管理这些数据和进行全局逻辑操作。这些数据结构和管理类构成数据逻辑层，通过调用上一层的函数来发起和处理特定功能的请求；

**用户界面层：**前端声明的所有Angular组件、模块构成组件层，负责页面的展示与接收处理用户的事件；

**后台逻辑架构：**

**数据持久层：**后端声明的所有Django模型构成数据持久层，用于与数据库交互；

**视图层：**具体进行业务操作的函数，包括两个公用的工具类和一些函数封装而来的视图类，接收前端传来的数据并进行处理；

**接口管理层：**对接口进行统一管理的函数和数据构成接口管理层，对Django原有的urls.py进行封装，用于路由配置和接口参数校验和类型转化，调用视图层的函数来进行相应的业务操作；

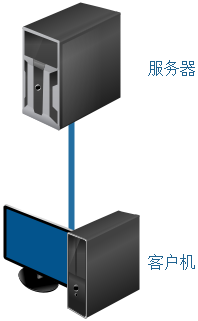
**Consumer：**是用于负责处理WebSocket业务的类；

注1：Django框架采用的是**MVT（模型-视图-模板）模式**而不是传统的MVC模式。在该模式中，V充当了控制器的功能，而T则类似于传统MVC模式中的V。而在本系统中没有使用T。

具体可以参考：<https://blog.csdn.net/sinat_34166518/article/details/89212518>

注2：箭头表示**调用关系**，实线表示**数据流向**。

## 系统物理架构



服务器（阿里云服务器）：

操作系统：CentOS 7.6 64位

网络带宽能力：0.2GB/s

网络收发包能力：6万PPS

内存：1GB 磁盘大小：40GB

处理器：2.5 GHz主频的Intel ® Xeon ®

软件部署：Django & Channels & MySQL & Redis

客户机：

操作系统：Windows 7 以上

浏览器：

Chrome 74.0.3729.169

360浏览器10.0

## 系统数据模型

另附文件

## 系统接口设计

## 外部接口设计

无

## 内部接口设计

注1：对于所有登录后执行的接口，前端都需要在请求头加上token用作身份验证，下列表格中不再列出。

注2：所有前端传来的数据后台都需要校验，下面只写明校验成功后的情况。校验失败则返回错误信息到前端。

|  |  |  |  |
| --- | --- | --- | --- |
| **所属** | **接口** | **功能** | **备注** |
| **用户模块** | 用户注册 | 前端发送用户名、密码、邮箱和验证码，后台执行注册 | 不需要token |
| 发送注册验证码 | 前端发送邮箱，后台生成验证码并发送 | 不需要token |
| 用户登陆 | 前端发送用户名和密码，后台执行登陆 | 不需要token |
| 忘记密码 | 前端发送用户名、邮箱、新密码和验证码，后台重置密码 | 不需要token |
| 发送忘密验证码 | 前端发送邮箱，后台生成验证码并发送 | 不需要token |
| 更新TOKEN | 后台生成一个新的token并传回前端 |  |
| 身份再次确认 | 前端发送密码，后台生成一个新的token并传回前端 | 用于修改敏感信息项时确认 |
| 修改密码 | 前端发送密码和新密码，后台修改密码 |  |
| 修改个人信息 | 前端发送全部个人信息，后台执行修改 |  |
| 添加好友 | 前端发送双方用户ID，后台生成一个好友请求，并实时通知对方（若在线） |  |
| 删除好友 | 前端发送双方用户ID，后台删除好友记录 |  |
| **项目模块** | 创建项目 | 前端发送用户ID、项目名、项目类型和项目描述，后台执行创建项目 |  |
| 获取项目概要 | 前端发送用户ID和项目ID，后台返回项目的概要信息 |  |
| 获取项目任务 | 前端发送用户ID和项目ID，后台返回项目的任务摘要数据列表 |  |
| 获取项目通知 | 前端发送用户ID和项目ID，后台返回该项目的通知摘要信息列表 |  |
| 获取项目成员 | 前端发送用户ID和项目ID，后台返回项目的成员摘要数据列表 |  |
| 邀请加入项目 | 前端发送双方用户ID和项目ID，后台自动将对方加入该项目中 | 项目经理权限 |
| 踢出项目成员 | 前端发送双方用户ID和项目ID，后台自动将对方从该项目中踢出 | 项目经理权限 |
| 获取成员角色 | 后台返回本系统中预设的角色数据列表 | 项目经理权限 |
| 分配成员角色 | 前端发送成员角色数据，后台执行角色的分配 | 项目经理权限 |
| 转让项目经理 | 前端发送双方用户ID，后台执行项目经理转让 | 项目经理权限 |
| 发布通知 | 前端发送用户ID、项目ID、通知标题和通知内容，后台对该项目新增一个通知，并实时通知在线成员 | 项目经理权限 |
| 删除通知 | 前端发送用户ID、项目ID和通知ID，后台对该通知进行删除 | 项目经理权限 |
| 查询通知 | 前端发送用户ID和通知ID，后台返回该通知的标题和内容 |  |
| 查询全部通知 | 前端发送用户ID，后台返回该用户的全部接收到的通知摘要信息列表 |  |
| **任务模块** | 布置任务 | 前端发送用户ID、项目ID、任务名和任务描述等，后台为该项目添加一个任务 | 项目经理权限 |
| 修改任务 | 前端发送用户ID、项目ID和任务数据，后台更新此数据 | 项目经理权限 |
| 删除任务 | 前端发送用户ID、项目ID和任务ID，后台对该项目的该任务进行删除 | 项目经理权限 |
| 查看任务详情 | 前端发送用户ID和任务ID，后台返回该任务的数据 | 项目经理权限 |
| 请求任务进度 | 前端发送用户ID、项目ID、任务ID、完成度和附加描述，后台发起一个新的进度请求，并实时通知给该项目项目经理（若不在线则发送邮件） |  |
| 获取进度请求 | 前端发送用户ID，后台返回与该用户有关的任务进度请求的数据列表 |  |
| 审批任务进度 | 前端发送用户ID、进度请求ID和审批结果，后台应用审批，并实时告知结果给任务请求方（若在线） | 项目经理权限 |
| 分配任务 | 前端发送用户ID、项目ID、任务ID和成员ID列表，后台更新任务成员 | 项目经理权限 |
| 查看我的任务 | 前端发送用户ID，后台返回该用户参与的所有任务ID数据 |  |

## 非功能性设计

## 安全性

数据库中，储存用户的密码通过以下方式加密：

密文 = SHA1(随机字符串+明文+随机字符串)

## 性能

使用Nginx提高服务器性能。

## 可靠性

由于Django中与MySQL的连接8小时内没有访问就会自动断开。后续的MySQL访问就会报错。为了防止该情况，后台程序里需每隔一定时间（1小时）对数据库进行一次操作，可通过另起一个线程来实现。

## 易用性

通过在多种浏览器上测试，确保兼容性。