

**软件设计说明书**

**智能与计算学部**

**软件工程**

**2016级**

**许宁馨(3016218074)**

**黄兴宇(3016218116)**

**许凌涵(3016218129)**

目录

[1引言 2](#_Toc532762628)

[1.1 编写目的 2](#_Toc532762629)

[1.2 背景 2](#_Toc532762630)

[1.3 参考资料 2](#_Toc532762631)

[2设计总体 2](#_Toc532762632)

[2.1 需求规定 2](#_Toc532762633)

[3接口设计 3](#_Toc532762634)

[3.1 Web API接口设计 3](#_Toc532762635)

[3.1.1 综述 3](#_Toc532762636)

[3.1.2用户模块接口设计 3](#_Toc532762637)

[3.1.3申请模块接口设计 4](#_Toc532762638)

[3.2 数据库设计说明 6](#_Toc532762639)

[3.2.1 数据库要求 6](#_Toc532762640)

[3.2.2 数据库概念设计 6](#_Toc532762641)

[3.2.3模块E-R图设计 8](#_Toc532762642)

[4模块设计 9](#_Toc532762643)

[4.1用户登录模块 9](#_Toc532762644)

[4.2用户注册模块 10](#_Toc532762645)

[4.3申请查看模块 10](#_Toc532762646)

[4.4用户信息管理模块 11](#_Toc532762647)

# 1 引言

## 编写目的

1. 编写软件设计说明书的目的在于配合用户需求说明书，描述系统的需求模型、功能和性能特点，为后期的软件开发工作提供依据。
2. 软件设计说明书地预期读者为用户、系统设计员及其他开发人员和相关审核人员。

## 背景

社团举办活动都需要审批一定数量的资金。由于职务不同，各个级别职务能管理的资金数额不同，且社团管理制度直接决定了资金申请的发起人和审批人之间具有固定的上下级关系。为了加强资金管理透明度，杜绝越级审批乱象，资金重复申请以及资金申请状况混乱等问题的出现，需要严格规范软件的业务流程，使得社团资金审批有条不紊地进行。

## 参考资料

《火球——UML大战需求分析》

# 2设计总体

## 2.1 需求规定

本软件系统的各种用户是惟一的参与者，参与者通过使用事件与系统进行交互，所有的时间综合起来构成用户的功能需求。参照本软件系统的需求规格说明书，详细设计系统的各个运行状态。

**2.2 运行环境**

本软件系统基于angular框架和json-server开发，可在chrome等常用浏览器上正常运行。

# 3 接口设计

## 3.1 Web API接口设计

### 3.1.1 综述

统一的输入输出参数

Request

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 用户ID |

（不需要登陆就能调用的接口、注册接口、登录接口，id是一个不需要输入的参数）

Response

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| code | Int | N | 错误码 |
| message | String | N | 错误消息 |

必须登录才能访问的接口

|  |  |
| --- | --- |
| 接口名称 | 描述 |
| users | 获取用户信息，修改用户信息 |
| applications | 获取、修改、增加、删除申请信息 |

错误返回码列表

|  |  |
| --- | --- |
| error\_code | error\_desc |
| 200 | 服务器成功返回用户请求的数据 |
| 201 | 用户新建或修改数据成功 |
| 202 | 表示一个请求已经进入后台排队 |
| 204 | 用户删除数据成功 |
| 400 | 用户发出的请求有错误，服务器没有进行新建或修改数据的操作 |
| 401 | 表示用户没有权限（令牌、用户名、密码错误） |
| 403 | 表示用户得到授权（与401错误相对），但是访问是被禁止的 |
| 404 | 用户发出的请求针对的是不存在的记录，服务器没有进行操作 |
| 406 | 用户请求的格式不可得 |
| 410 | 用户请求的资源被永久删除，且不会再得到的 |
| 422 | 当创建一个对象时，发生一个验证错误 |
| 500 | 服务器发生错误，用户将无法判断发出的请求是否成功 |

### 3.1.2用户模块接口设计

用户注册

|  |  |
| --- | --- |
| 接口 | users |
| 描述 | 用户注册 |
| 方法 | POST |

Request

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| username | String | Y | 用户名 |
| password | String | Y | 用户密码 |

Response

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 用户ID |

用户登录

|  |  |
| --- | --- |
| 接口 | users |
| 描述 | 用户登录 |
| 方法 | GET |

Request

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| username | String | Y | 用户名 |
| password | String | Y | 用户密码 |

Response

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| username | String | Y | 用户名 |
| password | String | Y | 用户密码 |
| position | String | Y | 用户职务 |
| department | String | Y | 用户部门 |
| id | Int | Y | 用户ID |

用户档案管理

|  |  |
| --- | --- |
| 接口 | users |
| 描述 | 用户登录 |
| 方法 | GET/PUT |

Request

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户ID |

Response

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| username | String | Y | 用户名 |
| password | String | Y | 用户密码 |
| position | String | Y | 用户职务 |
| department | String | Y | 用户部门 |
| id | Int | Y | 用户ID |

### 3.1.3申请模块接口设计

所有申请

|  |  |
| --- | --- |
| 接口 | applications |
| 描述 | 查看申请 |
| 方法 | GET |

Request

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 用户ID |

Response

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 申请ID |
| appTime | String | Y | 申请发起时间 |
| amount | Int | Y | 申请资金数量 |
| reason | String | Y | 申请原因 |
| fromId | String | Y | 申请发起者ID |
| toId | Int | Y | 申请受理者姓名 |
| type | String | Y | 申请类型 |
| status | String | Y | 申请状态 |
| statusUpdateTime | String | Y | 申请状态最近更新时间 |
| comment | String | Y | 申请审批意见 |

用户申请

|  |  |
| --- | --- |
| 接口 | applications |
| 描述 | 查看申请 |
| 方法 | GET/DELETE |

Request

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 用户ID |

Response

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 申请ID |
| appTime | String | Y | 申请发起时间 |
| amount | Int | Y | 申请资金数量 |
| reason | String | Y | 申请原因 |
| fromId | String | Y | 申请发起者ID |
| toId | Int | Y | 申请受理者姓名 |
| type | String | Y | 申请类型 |
| status | String | Y | 申请状态 |
| statusUpdateTime | String | Y | 申请状态最近更新时间 |
| comment | String | Y | 申请审批意见 |
| username | String | Y | 用户名 |
| password | String | Y | 用户密码 |
| position | String | Y | 用户职务 |
| department | String | Y | 用户部门 |

添加申请

|  |  |
| --- | --- |
| 接口 | applications |
| 描述 | 查看申请 |
| 方法 | GET/DELETE/POST |

Request

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 用户ID |

Response

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| id | Int | Y | 申请ID |
| appTime | String | Y | 申请发起时间 |
| amount | Int | Y | 申请资金数量 |
| reason | String | Y | 申请原因 |
| fromId | String | Y | 申请发起者ID |
| toId | Int | Y | 申请受理者姓名 |
| type | String | Y | 申请类型 |
| status | String | Y | 申请状态 |
| statusUpdateTime | String | Y | 申请状态最近更新时间 |
| comment | String | Y | 申请审批意见 |
| username | String | Y | 用户名 |
| password | String | Y | 用户密码 |
| position | String | Y | 用户职务 |
| department | String | Y | 用户部门 |

## 3.2 数据库设计说明

### 3.2.1 数据库要求

（1）安全性：数据备份，以防数据丢失。业务对象只能在自己的权限范围

内操作数据库。

（2）减少冗余：利用模式求精与范式设计合理数据结构，尽量避免数据冗

余。

（3）完善功能：保证管业务对象权限范围内的查询、添加、删除、修改功

能的实现。

### 3.2.2 数据库概念设计

用户注册表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 数据名称 | 数据类型 | 数据说明 |
| 基本数据 |  |  |  |
| 输入数据 | username | String | 用户名 |
| password | String | 登陆密码 |
| 输出数据 | id | Int | 用户ID |

用户登录表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 数据名称 | 数据类型 | 数据说明 |
| 基本数据 |  |  |  |
| 输入数据 | username | String | 用户名 |
| password | String | 用户密码 |
| 输出数据 | position | String | 用户职务 |
| department | String | 用户部门 |
| id | String | 用户ID |

用户档案管理表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 数据名称 | 数据类型 | 数据说明 |
| 基本数据 |  |  |  |
| 输入数据 | id | Int | 用户ID |
| 输出数据 | username | String | 用户名 |
| password | String | 用户密码 |
| position | String | 用户职务 |
| department | String | 用户部门 |

所有申请表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 数据名称 | 数据类型 | 数据说明 |
| 基本数据 |  |  |  |
| 输入数据 | id | Int | 申请ID |
| 输出数据 | appTime | String | 申请发起时间 |
| amount | Int | 申请资金数量 |
| reason | String | 申请原因 |
| fromId | String | 申请发起者ID |
| toId | Int | 申请受理者姓名 |
| type | String | 申请类型 |
| status | String | 申请状态 |
| statusUpdateTime | String | 申请状态最近更新时间 |
| comment | String | 申请审批意见 |

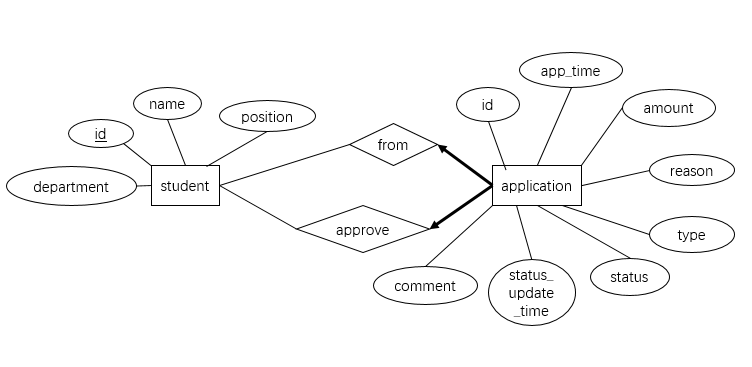
用户申请表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 数据名称 | 数据类型 | 数据说明 |
| 基本数据 |  |  |  |
| 输入数据 | id | Int | 申请ID |
| 输出数据 | appTime | String | 申请发起时间 |
| amount | Int | 申请资金数量 |
| reason | String | 申请原因 |
| fromId | String | 申请发起者ID |
| toId | String | 申请受理者姓名 |
| type | String | 申请类型 |
| status | String | 申请状态 |
| statusUpdateTime | String | 申请状态最近更新时间 |
| comment | String | 申请审批意见 |
| username | String | 用户名 |
| password | String | 用户密码 |
| position | String | 用户职务 |
| department | String | 用户部门 |

添加申请表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 数据名称 | 数据类型 | 数据说明 |
| 基本数据 |  |  |  |
| 输入数据 | id | Int | 申请ID |
| 输出数据 | appTime | String | 申请发起时间 |
| amount | Int | 申请资金数量 |
| reason | String | 申请原因 |
| fromId | String | 申请发起者ID |
| toId | String | 申请受理者姓名 |
| type | String | 申请类型 |
| status | String | 申请状态 |
| statusUpdateTime | String | 申请状态最近更新时间 |
| comment | String | 申请审批意见 |
| username | String | 用户名 |
| password | String | 用户密码 |
| position | String | 用户职务 |
| department | String | 用户部门 |

### 3.2.3模块E-R图设计



# 4 模块设计

## 4.1用户登录模块

模块功能：该模块完成用户登录

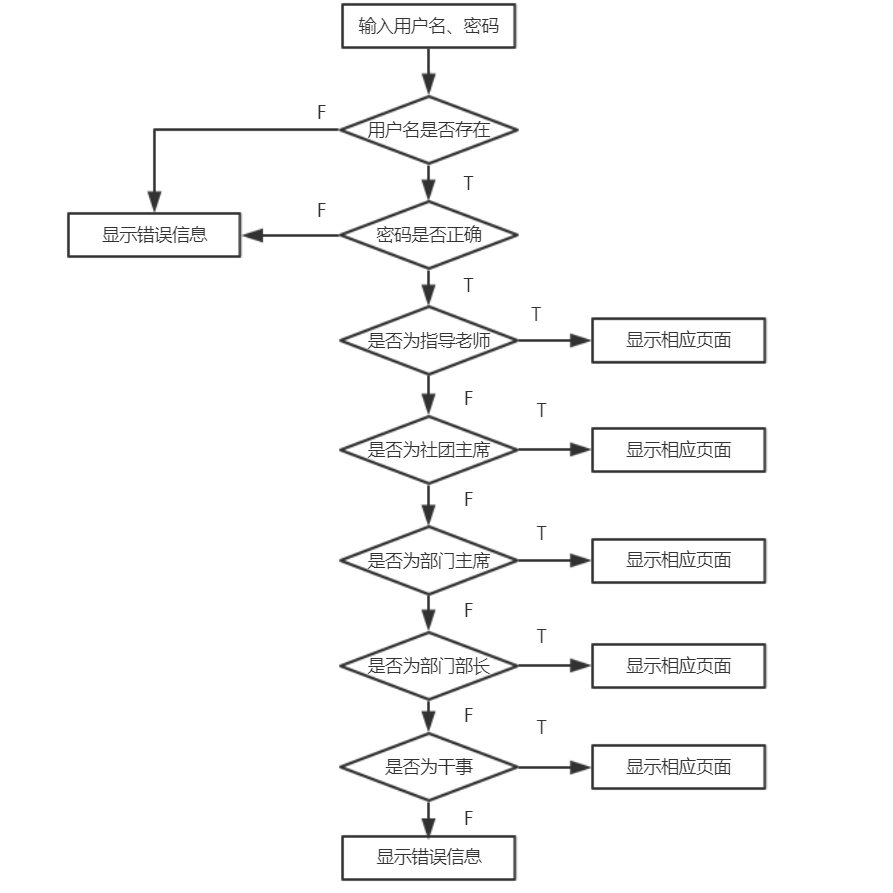
输入信息：用户名及密码

输出信息：页面跳转或错误提示

程序逻辑：输入用户名密码查询用户名是否存在，若存在查询是否与密码匹配。若匹配，判断用户类型。并根据用户类型跳转至不同页面。若不匹配，输出错误提示信息。

存储分配：数据库，插入参数。

测试要点：数据库连接是否正常、事务跳转是否正常、非法输入提示、非法操作提示。



## 4.2用户注册模块

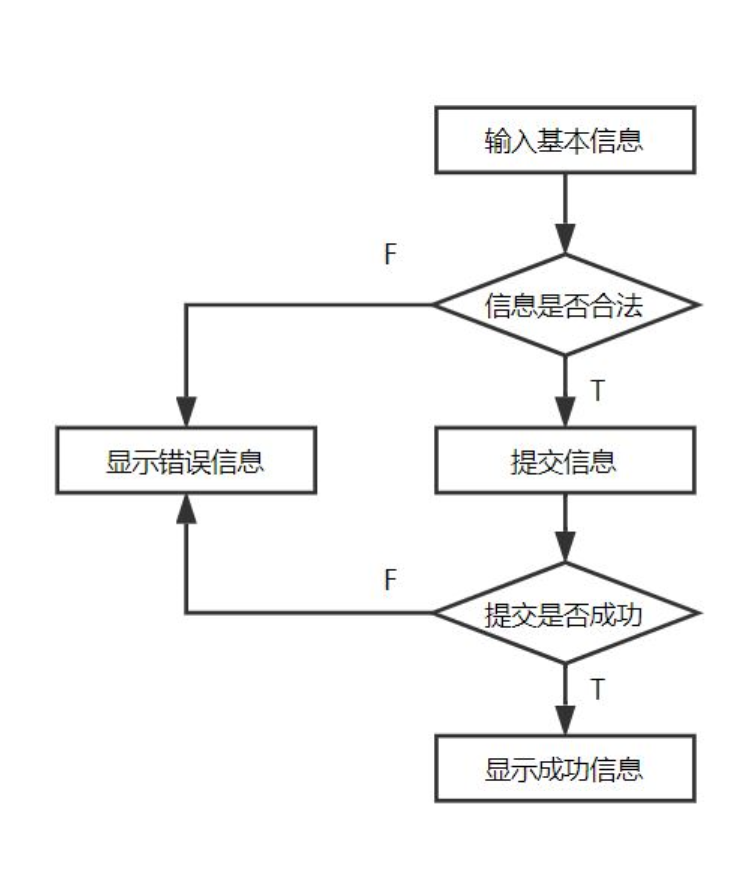
模块功能：该模块完成用户登录

输入信息：用户名密码以及其他基本信息

输出信息：页面跳转或错误提示

程序逻辑：显示基本信息填写页面。完成操作后提交，若提交失败，输出失败信息，若成功，输出成功信息。

测试要点：数据库连接是否正常、事务跳转是否正常、非法输入提示、非法操作提示。



## 4.3申请查看模块

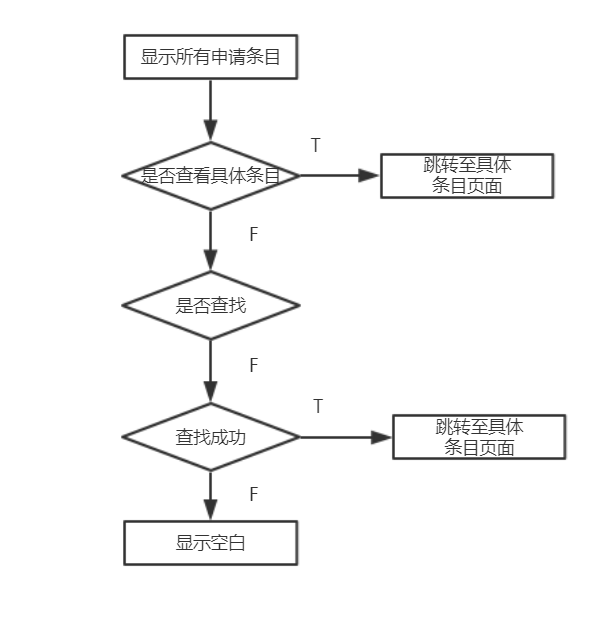
模块功能：该模块完成资金申请展示和查找

输入信息：查询关键词

输出信息：页面跳转或无查找目标提示

程序逻辑：显示所有资金申请条目，单击展示条目具体内容。若在查找框内输入关键词，则按关键词在数据库中查找相应条目并显示，单击展示条目跳转至条目具体内容。

测试要点：数据库连接是否正常、事务跳转是否正常。



## 4.4用户信息管理模块

模块功能：完成用户仅对自身用户信息的查看和修改

输入信息：需要修改的信息

输出信息：修改后的信息即时显示在屏幕上方

程序逻辑：进入用户档案界面后，可以在下方输入修要修改的信息，同时上方将即时显示修改后的信息。如果点击保存按钮，将能够保存已修改的信息。同时，可以在用户信息管理界面进行登出。

测试要点：是否正常修改了数据库里的对应数据，是否点击登出后回到正常界面。

