[1 概述 2](#_Toc481843735)

[2 开发使用流程 2](#_Toc481843736)

[2.1 团队协作- 2](#_Toc481843737)

[2.2 2](#_Toc481843738)

[2.3 技术框架- 3](#_Toc481843739)

[2.4 前端功能设计 3](#_Toc481843740)

[2.4.1 登录 3](#_Toc481843741)

[2.4.2 超级管理员 4](#_Toc481843742)

[2.4.3 社区管理员 4](#_Toc481843743)

[2.4.4 普通用户 5](#_Toc481843744)

[2.5 前后端接口设计 5](#_Toc481843745)

[3 ER图设计 6](#_Toc481843746)

[4 关系模式分析 6](#_Toc481843747)

[5 界面展示 7](#_Toc481843748)

[5.1 登录 7](#_Toc481843749)

[5.1.1 登录页面表单 7](#_Toc481843750)

[5.1.2 修改密码表单 7](#_Toc481843751)

[5.2 超级管理员 7](#_Toc481843752)

[5.2.1 首页 8](#_Toc481843753)

[5.2.2 流水 8](#_Toc481843754)

[5.2.3 任务 9](#_Toc481843755)

[5.2.4 订单 10](#_Toc481843756)

[5.2.5 产品 10](#_Toc481843757)

[5.2.6 人员 12](#_Toc481843758)

[5.2.7 社区 12](#_Toc481843759)

[5.3 社区管理员 13](#_Toc481843760)

[5.3.1 首页 13](#_Toc481843761)

[5.3.2 流水 13](#_Toc481843762)

[5.3.3 任务 14](#_Toc481843763)

[5.3.4 订单 15](#_Toc481843764)

[5.3.5 产品 15](#_Toc481843765)

[5.3.6 人员 15](#_Toc481843766)

[5.4 普通用户 16](#_Toc481843767)

[5.4.1 首页 16](#_Toc481843768)

[5.4.2 任务 16](#_Toc481843769)

[5.4.3 产品 16](#_Toc481843770)

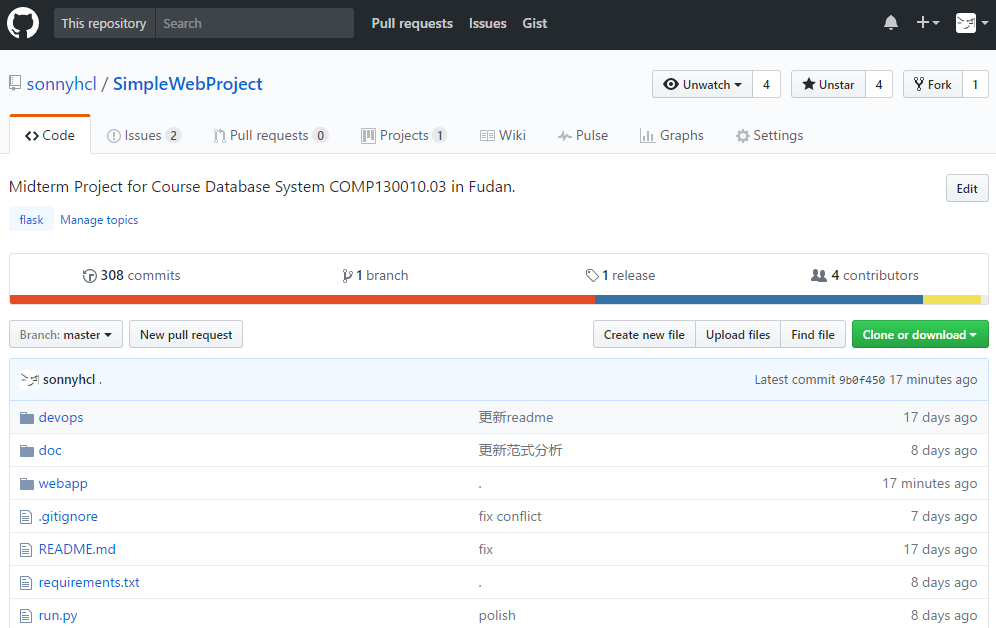
# 概述

阐述项目来源、项目需求分析、后面会用到的所有关键字概念解释

# 开发使用流程

## 团队协作-

本次开发采用GitHub作为团队协作的辅助工具，Git使得我们得以快速提交代码、比较前后代码不同、定位bug位置、发布issue、分配任务到个人，可以说是极大地提高了我们的开发效率，我们的项目地址为<https://github.com/sonnyhcl/SimpleWebProject>。



## 

开发Python Web应用当然需要一个好的IDE来提高开发效率，Pycharm Pro(学生可免费使用)除了支持python等各项功能之外，还支持html/css/js混合文件的语法高亮，实乃开发利器。

## 技术框架-

在技术框架的选择上，我们希望做到前后端分离得非常清晰，从而可以方便地将任务划分到每个人头上，因此我们最后采用的是轻量型组合。

后端方面，flask相比Django更容易上手，无需复杂的第一步配置就可以快速搭建一个轻量级的应用，同时又有丰富的插件可供选择。

数据库层面，我们选择了Sqlite而不是mysql，主要看中的是它的私有数据全部保存在一个.db文件中，方便我们直接在github上push/pull，无需额外配置数据库就可以轻易在不同成员之间迁移，保证了不同成员之间开发的时候，每个人的demo数据库是一致的。

至于前端我们选择的是BootStrap，这是当下最流行的前端基本框架，尤其是它对不同大小设备的支持较好，适合我们后期移植到微信公众号上。

## 前端功能设计

我们本次要做的是一个人员管理系统、产品管理系统和流水管理系统的大杂烩，其中人员管理包括社区管理和社区人员管理，产品管理系统包括产品管理和工艺管理，流水管理系统包括订单管理、任务管理和详细流水管理。使用的用户可以分类为三种角色，主管理员、社区分管理员和普通用户，他们各自对应的权限也各不相同，所有人需要登录才能查看相应的信息，其中主管理员可以增删查改所有内容，社区管理员只能增删查改跟本社区相关的信息，普通用户则只能查看跟自己相关的信息。

### 登录



|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 页面 | 功能 | 备注 |
| ALL | 登录 | 登录 |  |
|  |  | 修改密码 |  |

### 超级管理员



|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 页面 | 功能 | 备注 |
| Root | 首页 | 查看流水 | 查看所有社区所有人员的流水 |
|  | 流水 | 查看流水 | 查看所有社区所有人员的流水 |
|  |  | 修改流水 |  |
|  |  | 删除流水 |  |
|  |  | 增加流水 |  |
|  | 任务 | 查看任务 | 所有社区分配给所有人员的任务 |
|  |  | 修改任务 |  |
|  |  | 删除任务 |  |
|  |  | 增加任务 |  |
|  | 订单 | 查看订单 | 分配给所有社区的订单 |
|  |  | 修改订单 | 修改某个订单的内容 |
|  |  | 删除订单 | 删除某个订单 |
|  |  | 增加订单 | 新分配一个订单给社区 |
|  | 产品 | 查看产品和工艺 |  |
|  |  | 增加产品 |  |
|  |  | 删除产品 |  |
|  |  | 增加某产品的工艺 |  |
|  |  | 修改某产品的工艺 |  |
|  |  | 删除某产品的工艺 |  |
|  | 人员 | 查看人员 | 所有社区的人员 |
|  |  | 增加人员 |  |
|  |  | 删除人员 |  |
|  |  | 修改人员 |  |
|  | 社区 | 查看社区 | 此页面仅root可见 |
|  |  | 增加社区 |  |
|  |  | 删除社区 |  |
|  |  | 修改社区 |  |

### 社区管理员



|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 页面 | 功能 | 备注 |
| Admin | 首页 | 查看流水 | 查看该社区所有人员流水 |
|  | 流水 | 查看流水 | 查看该社区所有人员流水 |
|  |  | 修改流水 |  |
|  |  | 删除流水 |  |
|  |  | 增加流水 |  |
|  | 任务 | 查看任务 | 查看该社区分配给所有人员的任务 |
|  |  | 修改任务 |  |
|  |  | 删除任务 |  |
|  |  | 增加任务 |  |
|  | 订单 | 查看订单 | 分配到该社区的订单 |
|  | 产品 | 查看产品和工艺 | 只读 |
|  | 人员 | 查看人员 | 查看该社区的人员 |
|  |  | 增加人员 |  |
|  |  | 删除人员 |  |
|  |  | 修改人员 |  |

### 普通用户



|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 页面 | 功能 | 备注 |
| USER | 首页（流水） | 查看流水 | 查看自己的流水 |
|  | 任务 | 查看任务 | 查看自己的任务 |
|  | 产品 | 查看产品和工艺 | 只读 |

## 前后端接口设计

根据上面罗列出来的前端功能列表，我们可以设计出下面的前后端接口。其中权限从低到高有四个级别，GUEST->USER->ADMIN->ROOT，如权限一栏填的是ADMIN则意味着至少ADMIN级别的人才有权限使用。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 页面 | url | methods | 权限 | 功能 |
| 登录 | /login | GET | GUEST | 登录页面 |
|  | /log\_in | POST | GUEST | 登入 |
|  | /log\_out | POST | GUEST | 登出 |
|  | /reset\_password | POST | GUEST | 重置密码 |
| 首页 | / | GET | USER | 跳转到/index |
|  | /index | GET | USER | 首页页面 |
| 流水 | /transaction/table | POST | USER | 获取流水信息表 |
|  | /transaction/add | POST | ADMIN | 增加流水 |
|  | /transaction/delete | POST | ADMIN | 删除流水 |
|  | /transaction/modify | POST | ADMIN | 修改流水 |
| 任务 | /mission | GET | USER | 任务页面 |
|  | /mission/table | POST | USER | 获取任务信息表 |
|  | /mission/add | POST | ADMIN | 增加任务 |
|  | /mission/delete | POST | ADMIN | 删除任务 |
|  | /mission/modify | POST | ADMIN | 修改任务 |
| 订单 | /order | GET | ADMIN | 订单页面 |
|  | /order/table | POST | ADMIN | 获取订单信息表 |
|  | /order/add | POST | ROOT | 增加订单 |
|  | /order/delete | POST | ROOT | 删除订单 |
|  | /order/modify | POST | ROOT | 修改订单 |
| 产品 | /product | GET | USER | 产品页面 |
|  | /product/table | POST | USER | 获取产品信息表 |
|  | /product/add | POST | ROOT | 增加产品 |
|  | /product/delete | POST | ROOT | 删除产品 |
| (工艺) | /product/modify | POST | ROOT | 修改某产品的某工艺 |
|  | /product/add\_item | POST | ROOT | 为某产品增加工艺 |
|  | /product/delete\_item | POST | ROOT | 为某产品删除工艺 |
| 人员 | /user | GET | ADMIN | 人员页面 |
|  | /user/table | POST | ADMIN | 获取用户信息表 |
|  | /user/add | POST | ADMIN | 增加人员 |
|  | /user/delete | POST | ADMIN | 删除人员 |
|  | /user/modify | POST | ADMIN | 修改人员 |
| 社区 | /community | GET | ROOT | 社区页面 |
|  | /community/add | POST | ROOT | 增加社区 |
|  | /community/delete | POST | ROOT | 删除社区 |
|  | /community/modify | POST | ROOT | 修改社区 |
| 其它 | /show\_db\_log | GET | ROOT | 获取后台日志 |
|  | /show\_web\_log | GET | ROOT | 获取访问日志 |
|  | /error | GET | USER | 错误页面 |
|  | 404 | - | USER | 404 not found |

# ER图设计

参考书上第五章，画一个visio的图

# 关系模式分析

进行一波范式分析

# 界面展示

在我们的系统中所有人员被划分为三种角色，因此下面的界面展示以角色为划分点，展示登录页面和登陆后各个不同角色所能看到的不同界面

## 登录

### 登录页面表单



### 修改密码表单



## 超级管理员

下面将罗列出该角色所能看到的所有页面，还有他能进行的所有操作。

### 首页



### 流水





### 任务







### 订单







### 产品











### 人员







### 社区







## 社区管理员

### 首页



### 流水







### 任务







### 订单



### 产品



### 人员





## 普通用户

### 首页



### 任务



### 产品

