**代加工管理系统项目报告**

[1 概述 3](#_Toc481960439)

[1.1 项目简介 3](#_Toc481960440)

[1.2 需求分析 3](#_Toc481960441)

[1.3 名词说明 3](#_Toc481960442)

[2 数据库设计与分析 4](#_Toc481960443)

[2.1 ER图设计 4](#_Toc481960444)

[2.2 关系模式分析 4](#_Toc481960445)

[3 开发使用流程 5](#_Toc481960446)

[3.1 团队协作- 5](#_Toc481960447)

[3.2 6](#_Toc481960448)

[3.3 技术框架- 6](#_Toc481960449)

[3.4 前端功能设计 6](#_Toc481960450)

[3.4.1 登录 7](#_Toc481960451)

[3.4.2 主管理员 7](#_Toc481960452)

[3.4.3 社区管理员 8](#_Toc481960453)

[3.4.4 普通用户 8](#_Toc481960454)

[3.5 前后端接口设计 9](#_Toc481960455)

[4 界面展示 11](#_Toc481960456)

[4.1 登录 11](#_Toc481960457)

[4.1.1 登录页面表单 11](#_Toc481960458)

[4.1.2 修改密码表单 11](#_Toc481960459)

[4.2 超级管理员 11](#_Toc481960460)

[4.2.1 首页 12](#_Toc481960461)

[4.2.2 流水 12](#_Toc481960462)

[4.2.3 任务 13](#_Toc481960463)

[4.2.4 订单 14](#_Toc481960464)

[4.2.5 产品 14](#_Toc481960465)

[4.2.6 人员 16](#_Toc481960466)

[4.2.7 社区 16](#_Toc481960467)

[4.3 社区管理员 17](#_Toc481960468)

[4.3.1 首页 17](#_Toc481960469)

[4.3.2 流水 17](#_Toc481960470)

[4.3.3 任务 18](#_Toc481960471)

[4.3.4 订单 19](#_Toc481960472)

[4.3.5 产品 19](#_Toc481960473)

[4.3.6 人员 19](#_Toc481960474)

[4.4 普通用户 20](#_Toc481960475)

[4.4.1 首页 20](#_Toc481960476)

[4.4.2 任务 20](#_Toc481960477)

[4.4.3 产品 20](#_Toc481960478)

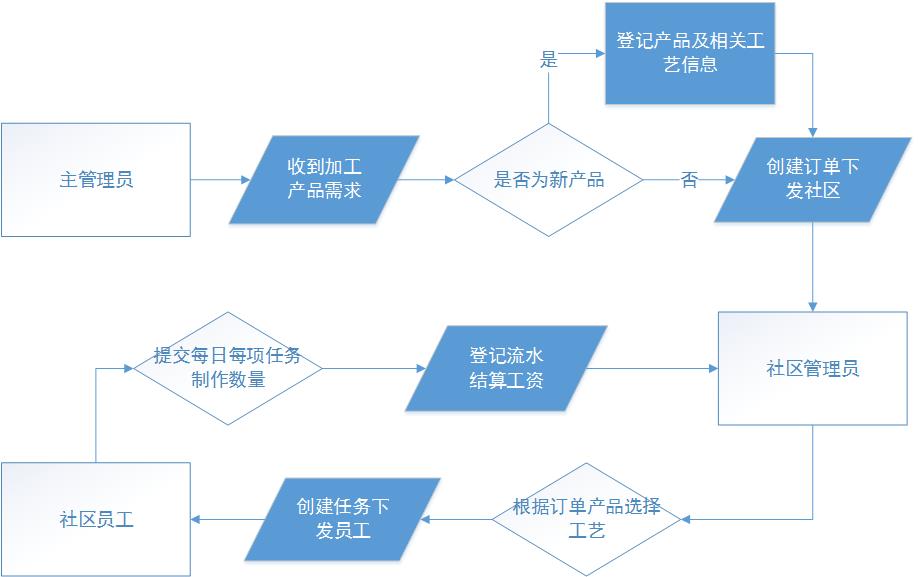
# 概述

## 项目简介

本项目以上海市彩虹桥公益社的社区代加工项目为背景，根据课程要求进行了相应的抽象与修改后完成的。本项目实现了信息管理系统、权限管理系统、订单分配系统与工资结算系统等功能模块，较好的满足了使用需求，期望以数据库管理系统来摆脱过去纸上作业所带来的不便，在保持准确性的情况下提升代加工项目的管理效率。

## 需求分析

本代加工系统的工作流程如下图所示：



流程大致如下：主管理员收到一个产品的制作需求后，设置订单下发给社区管理员。社区管理员收到订单后，根据订单相关产品的工艺信息分别制作成任务下发给员工。员工收到任务后开始制作，并每日提交当日的制作情况给社区管理员，经其审核后记录流水并结算应得工资。

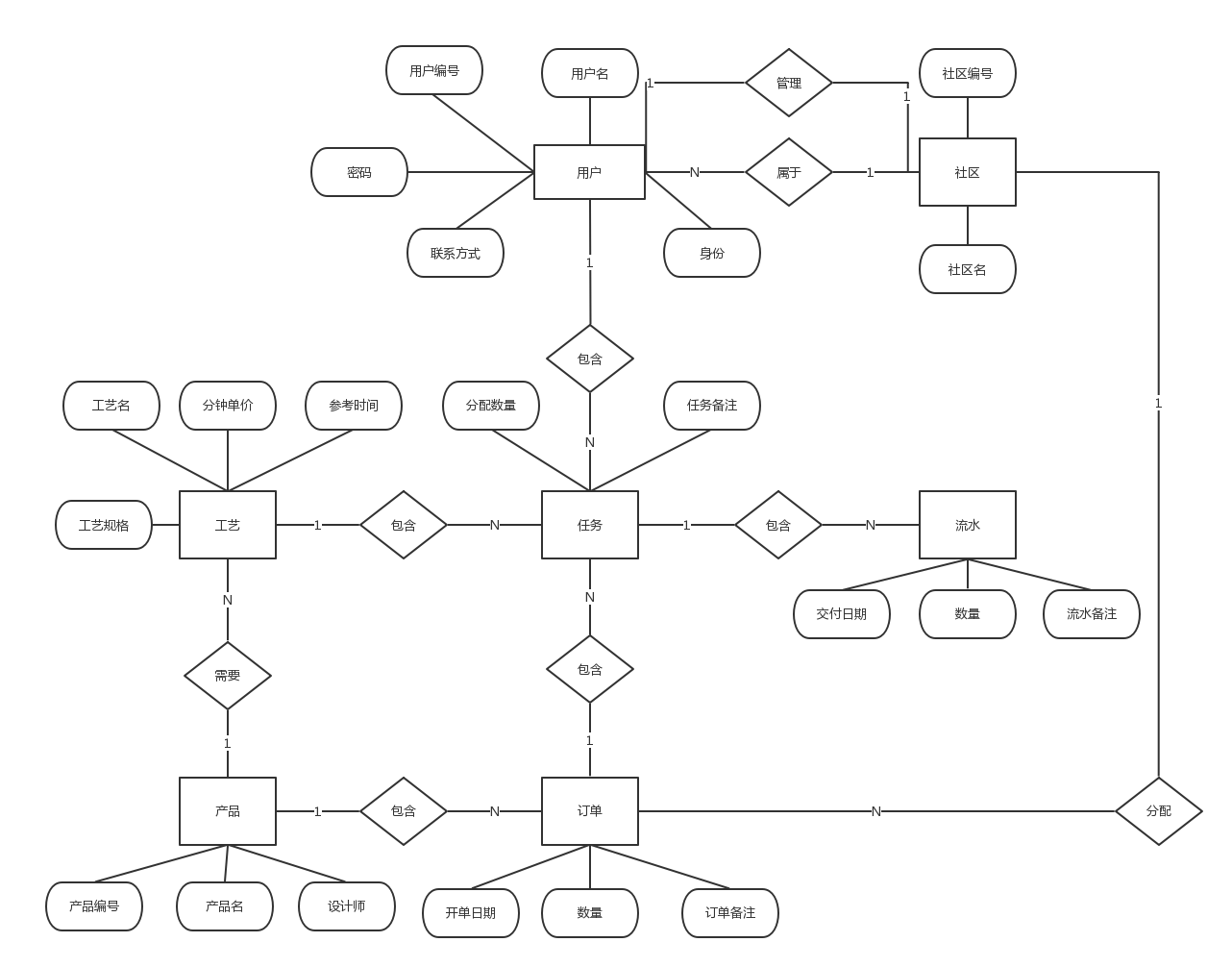
因此，本系统需要实现几个功能模块：权限管理模块、社区信息管理模块、人员信息管理模块、产品信息管理模块、订单信息管理模块、任务信息管理模块、流水信息管理模块。每个模块对应一个操作页面，并根据用户的身份控制其操作权限。

## 名词说明

|  |  |
| --- | --- |
| 名词 | 释义 |
| 社区 | 员工**工作的地点** |
| 产品 | **需要代加工的产品**，由公司或个人以需求的形式提交给主管理员。**每个产品需要由若干道工艺**来完成，如：一款双肩包的制作包含前整理、裁剪、车缝面、车缝里等八道工艺 |
| 工艺 | **制作一个产品所需要的步骤**，如上所示。每道工艺有相应的**制作时间（如40min）**以及**制作的单价（如0.5元/min）**，制作单价**可能会随着人力成本的变化而变化**，制作时间是基本固定的。 |
| 主管理员 | **总负责人**，拥有对人员、社区、产品、订单、任务、流水等所有信息增删改查的权限 |
| 社区管理员 | **某一社区的负责人**，拥有对所管理社区订单、任务、流水、人员的增删改查权限 |
| 员工 | 社区内的员工，每日**完成对产品工艺的制作**并**提交制作情况**，拥有对自己流水信息的查看权限 |
| 订单 | 主管理员收到制作产品的需求后，选择**产品与数量**（如双肩包，100个），以订单的形式下发给社区 |
| 任务 | 社区管理员收到订单后，根据其所对应**产品的工艺以及数量**分别制作任务（如双肩包的裁剪，50个）下发给员工 |
| 流水 | 员工每日提交任务的**制作情况**（如裁剪，10个）给社区管理员，由社区管理员检验确认后**登记流水并结算相应的工资金额** |

# 数据库设计与分析

## ER图设计



## 关系模式分析

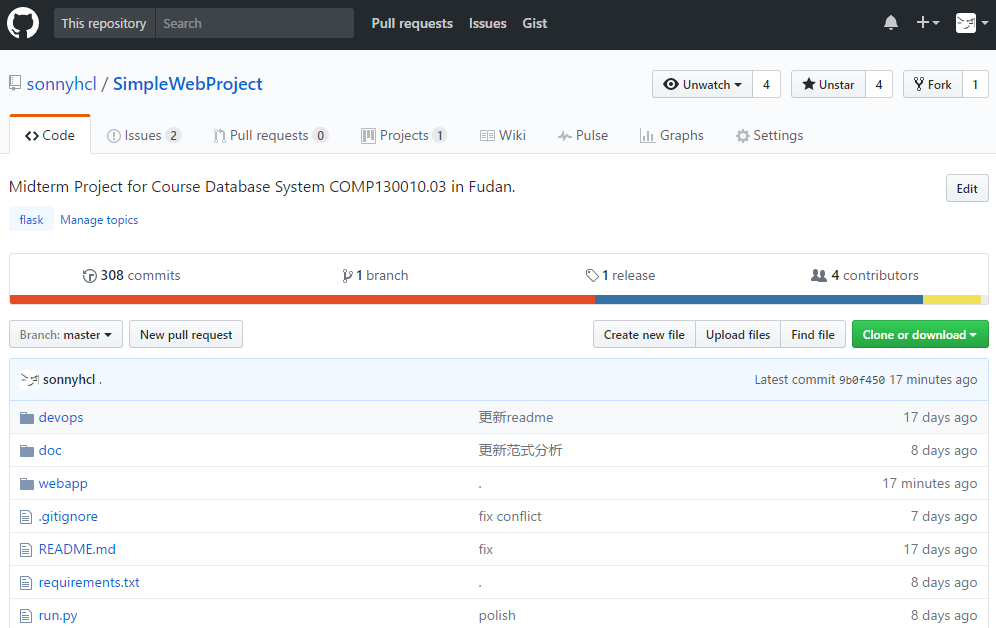
本数据库系统共有八个关系模式，现一一分析范式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 关系模式 | 主键 | 外键 | 满足范式 |
| 产品 | （产品编号，产品名，设计师） | 产品编号 |  | BCNF |
| 工艺 | （产品编号, 工艺名，分钟单价，参考时间, 工艺规格） | 产品编号,  工艺名 | 产品编号 | BCNF |
| 社区 | （社区编号，社区名，管理员编号） | 社区编号 | 管理员编号 | BCNF |
| 用户 | （用户编号，用户名，身份，密码，联系方式，社区编号） | 用户编号 | 社区编号 | BCNF |
| 订单 | （订单编号，产品编号，开单日期，数量，订单备注） | 订单编号 | 产品编号 | BCNF |
| 分配 | （任务编号，用户编号，工艺编号，订单编号） | 任务编号 | 用户编号  工艺编号  订单编号 | 3NF |
| 任务 | （用户编号，工艺编号，订单编号，分配数量，任务备注） | 用户编号  工艺编号  订单编号 |  | BCNF |
| 流水 | （任务编号, 交付日期，数量，流水备注) | 任务编号  交付日期 |  | BCNF |

# 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 |  |

### 主管理员



|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

# 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

