

Web编程技术(含分布式计算) **期末项目**

|  |  |
| --- | --- |
| 项目技术级别 | 单选，在对应位置打勾 |
| DAO设计模式 |  |
| MVC设计模式 |  |
| Spring+MyBatis框架 |  |
| SSM框架 | √ |
| 其他： |  |

班 级 软件1912 学 号 201921122050

姓 名 游嘉升 自评成绩 A

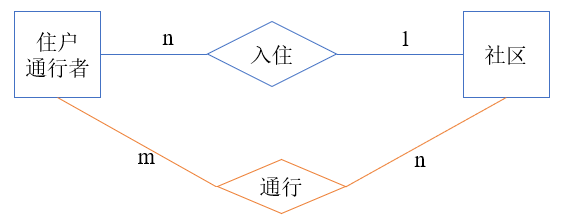
教师评定成绩

2021年 7 月 4 日

**（1）项目开发环境**

|  |  |
| --- | --- |
| 软件 | 版本 |
| JDK | 13.0.2 |
| Tomcat | 8.5.65 |
| IntelliJ IDEA | 2020.2.4 |
| Maven | 3.8.1 |
| MySQL | 8.0.23 |

**(2)数据设计（E-R图，各表介绍）**



**社区表community**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 属性 | 约束 | 备注 |
| 1 | cId | int | primary key auto\_increment | id |
| 2 | cName | varchar(30) | not null | 社区名称 |
| 3 | cProvince | varchar(30) |  | 省份 |
| 4 | cCity | varchar(30) |  | 城市 |
| 5 | cStreet | varchar(30) |  | 街道 |
| 6 | cTel | varchar(30) |  | 电话 |

**用户表members**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 属性 | 约束 | 备注 |
| 1 | mId | int | primary key auto\_increment | id |
| 2 | mName | varchar(30) | not null | 用户姓名 |
| 3 | mPhone | varchar(30) | not null | 手机号 |
| 4 | mSex | varchar(10) |  | 性别 |
| 5 | mAge | int |  | 年龄 |
| 6 | mHouseNumber | Varchar(50) |  | 楼号+房号 |
| 7 | mWorkUnit | varchar(50) |  | 工作单位名称 |
| 8 | mCarNumber | Varchar(20) |  | 车牌号 |
| 9 | mIsHousehold | bool | default true | 是否为业主 |
| 10 | cId | int |  | 外键，社区id |

**通行表record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 属性 | 约束 | 备注 |
| 1 | rId | int | primary key auto\_increment | id |
| 2 | mId | int | not null | 外键，用户id |
| 3 | cId | int | not null | 外键，社区id |
| 4 | rIsOutCity | bool | default false | 一周内是否离开本城 |
| 5 | rIsFromForei | bool | default false | 一月内是否来自境外 |
| 6 | rIsHosehold | bool | default true | 是否是业主 |
| 7 | rNowTime | datetime |  | 通行进入时间 |

**(3)主要功能**

**社区功能：**

|  |
| --- |
| 插入新社区信息 |
| 查找所有社区记录 |
| 修改某社区信息 |

**用户表功能**

|  |
| --- |
| 为某社区添加住户 |
| 修改某住户信息，如出境、出城信息等 |
| 查询所有用户信息 |

**通行记录功能**

|  |
| --- |
| 插入一条通行信息 |
| 删除一条通行信息 |
| 查询某用户在某社区的通行记录 |

**关联查询功能**

|  |
| --- |
| 查询一周内离开本城用户信息，包括其所在社区的名称、街道、社区电话信息。 |

**(4)亮点技术补充介绍（前端、后端皆可）**

前端没有什么亮点技术，稍微有亮点的就是用了iframe标签实现了部分局部刷新让用户没有那么强的割裂感。

后端使用了ssm技术框架，将模型层，控制层和数据访问层很好的切割开来。完成了在一定程度上对功能的解耦。

**(5)开发中遇到什么难点，如何解决的，举两个例子说明。**

1、关于ssm框架，从最简单的框架的配置到后面bean以及mapper的配置，通通绕不过xml，简直就是配置地狱。

解决方法：了解了ssm整合后，也了解了一些注解之后不用写bean也不用写mapper了，各种环境的配置也随着上手的程度简单了不少。

2、关于各种页面之间跳转的，为了实现在一个页面中多个功能的前端形式，不得不导致了页面和controller之间跳转混乱，进而导致页面命名混乱。

解决方法：硬着头皮写，在没有学习ajax局部刷新之前用iframe实现部分局部刷新，没有学习js的进入新页面只能先调用controller来进行预先传参给页面显示，导致效果很差，后来学习了ajax和jQuery也没有应用其中，勉强算是将期末作业做好了但是很多地方混乱的不想再去看，也没有再去修改，后面课设的时候多加注意吧。