桂林航天工业学院

课程设计存档材料

https://github.com/ALI1416/Network-Voting-System/

课程名称： 软件建模课程设计

开课学期： **2018—2019学年第二学期**

专 业： 软件工程

班 级：

姓 名：

学 号：

指导老师：

报告日期：**2019年6月21日**

桂林航天工业学院课程设计任务书

设计题目： 网络投票管理系统设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生姓名 | |  | | |
| 课程名称 | | 软件建模技术课程设计 | 专业班级 |  |
| 地 点 | | 巡天308 | 起止时间 | 2019.6.17-2019.6.20 |
| 设计内容 | 1、系统需求及问题域描述：（1）针对选择的系统对其问题域进行描述；(2)以文字性描述系统的功能需求、非功能需求，并列出功能需求。  2、用例图及用例的事件流描述  包括：(1)描述系统的参与者；(2)描述系统的用例；(3)画出用例图；(4)对每个用例采用事件流进行描述；  3、静态结构模型：(1)分析系统的类，包括实体类和界面类，分析每个类中包含的主要属性及操作；(2)建立类图，用多个类图来描述系统；  4、动态行为模型：(1)用顺序图描述用例的功能（5-6个）；(2)用通讯图描述系统的功能（3-4个）；  5、动态行为模型：(1)用活动图描述用例的功能（5-6个）；(2)用状态机图描述复杂的对象的生命周期过程（3-4个）；  6、数据库设计：用power designer 工具设计项目的数据库部分的E-R模型，并转换到具体的数据库管理系统，如SQL Server2012等。  7、物理模型：描述系统各个节点之间的关系。  8、程序设计：每个功能的界面、功能说明及主要代码。 | | | |
| 设计参数 | 1、本次课设遵循软件开发的一般过程，主要包含需求建模、静态结构建模、动态行为建模、数据库设计、物理建模及实现。  2、利用所学的软件建模技术为每一个阶段采用UML建模语言进行建模。 | | | |
| 设计进度 | 第一次课：指导学生完成系统需求及问题域描述和用例图及用例的事件流描述两部分的内容。  第二次课：指导学生完成静态结构模型和动态行为模型两部分的内容。  第三次课：指导学生完成数据库设计和部分功能的界面、功能主要代码的程序设计。  第四次课：指导学生完成系统的物理建模和剩下的部分功能的界面、功能主要代码的程序设计。 | | | |
| 设计成果 | 本次课程设计最终产生一份课程设计报告和课设设计所实现的系统，且系统能够正常运行。 | | | |
| 参考资料 | [1] 冀振燕编著．UML系统分析与设计教程(第2版).人民邮电出版社，2014  （二）教学参考书  [2] 布奇 (Grady Booch) ．UML用户指南（第2版）.人民邮电出版社，2013  [3] Hassan Gomaa 著. 软件建模与设计. 机械工业出版社，2014.08  [4] 冀振燕编著．《UML系统分析设计与应用案例》.人民邮电出版社，2003  [5] 吕云翔，赵天宇，丛硕 著. UML与Rose建模实用教程. 人民邮电出版社，2016.04 | | | |
| 说明 | 1．本表应在每次实施前由指导教师填写一式2份，审批后所在教学单位和指导教师各留1份。2．多名学生共用一题的，在设计内容、参数、要求等方面应有所区别。3．若填写内容较多可另纸附后。 | | | |

指导教师：

教研室主任：

年 月 日

****

**本科课程设计报告**

|  |  |
| --- | --- |
| 课程名称 | 软件建模技术 |
| 开课学期 | 2018-2019第2学期(2019春) |
| 学 院 | 计算机科学与工程学院 |
| 题 目 | 网络投票管理系统设计 |
| 专业班级 |  |
| 学 号 |  |
| 学生姓名 |  |
| 指导教师 |  |
| 报告日期 | 2019年6月21日 |

成绩：

**课程设计评分标准**

|  |  |  |  |
| --- | --- | --- | --- |
| 平时（30%） | 全勤、学习态度端正、实验认真、积极回答问题 | 优秀（90-100） |  |
| 偶有缺勤、实验认真、回答问题较积极 | 良好（80-89） |
| 旷课2次以内、偶有迟到、实验认真、回答问题较好 | 中（70-79） |
| 旷课2次以上、学习态度一般、基本能回答出问题 | 及格（60-69） |
| 经常旷课，实验过程不认真、问题回答不积极 | 不及格(<60) |
| 考核（70%） | 作品功能完全实现、作品演示操作熟练、问题回答准确；设计论文结构合理、内容符合要求、论文格式符合规范要求、参考文献新 | 优秀（90-100） |  |
| 作品功能基本完全实现、作品演示操作熟练、问题回答准确；设计论文结构合理、内容符合要求、论文格式符合规范要求、参考文献新 | 良好（80-89） |
| 作品功能实现70%、作品演示操作基较熟练、问题回答基本准确；设计论文结构基本合理、内容基本符合要求、论文格式基本符合规范要求、参考文献较新 | 中（70-79） |
| 作品功能实现50%、作品演示操作较熟练、部分问题回答准确；设计论文结构较合理、 论文格式基本符合规范要求、有参考文献 | 及格（60-69） |
| 作品功能不能实现、作品演示操作不熟练、问题回答效果差；设计论文结构不合理、 论文格式不符合规范要求、无参考文献 | 不及格(<60) |
| 总评 | | |  |

目 录

[第1章 系统需求 1](#_Toc12112851)

[1.1 问题域描述 1](#_Toc12112852)

[1.2 系统需求 1](#_Toc12112853)

[第2章 需求分析 2](#_Toc12112854)

[2.1 系统参与者 2](#_Toc12112855)

[2.2 系统用例 2](#_Toc12112856)

[2.3 用例图 3](#_Toc12112857)

[2.3.1 未登录者用例图 3](#_Toc12112858)

[2.3.2 登录者用例图 3](#_Toc12112859)

[2.3.3 管理员用例图 3](#_Toc12112860)

[2.4用例的事件流描述 3](#_Toc12112861)

[2.4.1 注册用例图事件流描述 3](#_Toc12112862)

[2.4.2 登录用例图事件流描述 4](#_Toc12112863)

[2.4.3 注销用例图事件流描述 4](#_Toc12112864)

[2.4.4 发布投票用例图事件流描述 5](#_Toc12112865)

[2.4.5 查看投票用例图事件流描述 6](#_Toc12112866)

[2.4.6 维护投票用例图事件流描述 7](#_Toc12112867)

[2.4.7 编辑个人信息用例图事件流描述 7](#_Toc12112868)

[第3章 系统静态模型 8](#_Toc12112869)

[3.1 实体类 8](#_Toc12112870)

[3.1.1 用户实体类 8](#_Toc12112871)

[3.1.2 发布投票实体类 9](#_Toc12112872)

[3.1.3 投票选项实体类 10](#_Toc12112873)

[3.1.4 投票记录实体类 10](#_Toc12112874)

[3.2 界面类 11](#_Toc12112875)

[3.2.1 主界面类 11](#_Toc12112876)

[3.2.2 注册界面类 11](#_Toc12112877)

[3.2.3 登录界面类 11](#_Toc12112878)

[3.2.4 查看投票界面类 12](#_Toc12112879)

[3.2.5 发布投票界面类 12](#_Toc12112880)

[3.2.6 查看投票详情界面类 12](#_Toc12112881)

[3.2.7 查看我发布的界面类 12](#_Toc12112882)

[3.2.8 编辑投票界面类 12](#_Toc12112883)

[3.2.9 查看我投的票界面类 12](#_Toc12112884)

[3.2.10 查看我的信息界面类 12](#_Toc12112885)

[3.3 类图 13](#_Toc12112886)

[3.3.1 用户界面类图 13](#_Toc12112887)

[3.3.2 实体类图 13](#_Toc12112888)

[第4章 系统动态模型 14](#_Toc12112889)

[4.1 顺序图 14](#_Toc12112890)

[4.1.1 登录顺序图 14](#_Toc12112891)

[4.1.2 查看发布顺序图 14](#_Toc12112892)

[4.1.3 发布投票顺序图 15](#_Toc12112893)

[4.1.4 投票顺序图 15](#_Toc12112894)

[4.1.5 修改个人信息顺序图 16](#_Toc12112895)

[4.2 通信图 16](#_Toc12112896)

[4.2.1 登录通信图 16](#_Toc12112897)

[4.2.2 查看发布通信图 17](#_Toc12112898)

[4.2.3 发布投票通信图 17](#_Toc12112899)

[4.2.4 投票通信图 17](#_Toc12112900)

[4.2.5 修改个人信息通信图 17](#_Toc12112901)

[4.3 活动图 18](#_Toc12112902)

[4.3.1 登录活动图 18](#_Toc12112903)

[4.3.2 查看发布活动图 18](#_Toc12112904)

[4.3.3 发布投票活动图 19](#_Toc12112905)

[4.3.4 投票活动图 19](#_Toc12112906)

[4.3.5 修改密码活动图 20](#_Toc12112907)

[4.4 状态机图 20](#_Toc12112908)

[4.4.1 用户状态机图 20](#_Toc12112909)

[4.4.2 发布的投票状态机图 20](#_Toc12112910)

[4.4.3 投票选项状态机图 20](#_Toc12112911)

[第5章 数据库设计 21](#_Toc12112912)

[5.1 E-R模型 21](#_Toc12112913)

[5.2 表清单 21](#_Toc12112914)

[5.2.1 用户表 21](#_Toc12112915)

[5.2.1 发布的投票表 22](#_Toc12112916)

[5.2.1 投票选项表 22](#_Toc12112917)

[5.2.1 投票记录表 22](#_Toc12112918)

[第6章 物理模型 23](#_Toc12112919)

[部署图 23](#_Toc12112920)

[第7章 程序实现 24](#_Toc12112921)

[7.1 注册、登录模块 24](#_Toc12112922)

[7.2 首页、查看投票模块 24](#_Toc12112923)

[7.3 查看投票详情模块 26](#_Toc12112924)

[7.4 发布新的投票模块 27](#_Toc12112925)

[7.5 查看我的发布模块 27](#_Toc12112926)

[7.6 查看投票详情模块 28](#_Toc12112927)

[7.7 查看我投的票模块 28](#_Toc12112928)

[7.8 查看我的信息、修改密码模块 29](#_Toc12112929)

[7.9 查看用户信息、批量导入导出用户模块 29](#_Toc12112930)

[总结 30](#_Toc12112931)

[附件 31](#_Toc12112932)

# 第1章 系统需求

## 1.1 问题域描述

在网络投票管理系统中，要为每一个发布者建立一个账户，其中账户中存储ID、账户名、密码、个人信息等。发布者可以发布投票，没有注册的不可以发布。发布时可以选择单选、多选和多次(多个选项，每个选项可以投多次)投票，也可以选择非实名投票(注册的也可以投票)和实名投票，还可以选择私密投票和公开投票，还可以选择不公开投票结果和公开投票结果。可以删除投票、暂停投票、修改发布信息，也可以查看、导出投票结果，还可以查看、搜索自己发布的所有投票。可以编辑自己的个人信息。

投票者可以不注册，但要获取IP地址和MAC地址(注册的也要)。单选和多选只可以投一次，多次投票不可超过最多投票数量。其中未注册者根据MAC地址判断，注册者根据ID和MAC地址判断(也不能与未注册的MAC地址相同)。

所有人都可以查看、搜索所有公开投票，并进行投票。

管理员可以查看所有投票，也可以删除投票、暂停投票、修改发布信息，也可以参看、导出投票结果，还可以修改注册者的个人信息。

## 1.2 系统需求

**未登录者：**

可以查看、搜索公开投票。

可以投非实名投票。

可以查看公开投票结果。

**登录者：**

可以查看、搜索公开投票。

可以投票。

可以查看公开投票结果。

可以发布投票。

可以查看、搜索、编辑(删除、暂停、修改)、导出自己发布的投票。

可以查看自己投过的票。

可以修改个人信息。

**管理员：**

可以查看、搜索所有投票。

可以投票。

可以查看所有投票结果。

可以发布投票。

可以查看、搜索、编辑(删除、暂停、修改)、导出所有投票。

可以查看所有人投过的票。

可以修改所有注册者信息。

**备注：**

私密投票只可以通过链接访问，访问时，变成公开投票。搜索投票无法找到。

# 第2章 需求分析

## 2.1 系统参与者

1. **未登录者：**

**描述：**可以查看、搜索公开投票。可以投非实名投票。可以查看公开投票结果。

**示例：**未登录用户。

1. **登录者：**

**描述：**可以查看、搜索公开投票。可以投票。可以查看公开投票结果。可以发布投票。可以查看、搜索、编辑(删除、暂停、修改)、导出自己发布的投票。可以查看自己投过的票。可以修改个人信息。

**示例：**已登录用户。

1. **管理员：**

**描述：**可以查看、搜索所有投票。可以投票。可以查看所有投票结果。可以发布投票。可以查看、搜索、编辑(删除、暂停、修改)、导出所有投票。可以查看所有人投过的票。可以修改所有注册者信息。

**示例：**管理员。

## 2.2 系统用例

1. **注册**

本用例提供用户注册功能。

1. **登录**

本用例提供注册者登录功能。

1. **注销**

本用例提供登录者注销功能。

1. **发布投票**

本用例提供注册者发布投票功能。

1. **查看投票**

本用例提供未登录者查看、搜索公开投票，投非实名公开投票；登录者查看、搜索公开投票，投公开投票，查看自己投过的票；管理员查看、搜索所有投票，投票，查看所有人投过的票。

1. **维护投票**

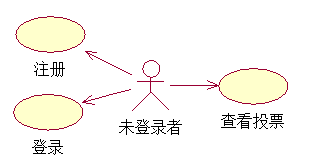
本用例提供登录者删除、暂停、修改、导出自己发布的投票；管理员删除、暂停、修改、导出所有发布的投票。

1. **编辑个人信息**

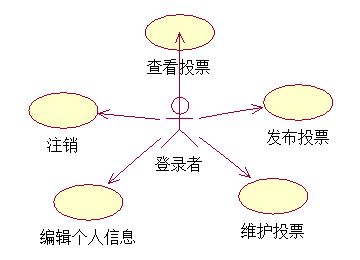
本用例提供登录者修改个人信息；管理员修改所有注册者信息。

## 2.3 用例图

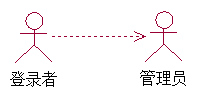
### 2.3.1 未登录者用例图



### 2.3.2 登录者用例图



### 2.3.3 管理员用例图



## 2.4用例的事件流描述

### 2.4.1 注册用例图事件流描述

1. **前置条件**

没有。

1. **后置条件**

如果这个用例成功，系统添加注册者信息。反之，系统状态不变。

1. **扩充点**

没有。

1. **事件流**
   1. **基流**

当用户希望注册账号时，用例启动。

1. 系统提示用户输入用户名、密码、确认密码、个人信息等。
2. 系统判断输入信息是否合法，以及密码和确认密码是否一致，若合法且一致(E-1)。
3. 系统验证用户名是否存在，若不存在(E-2)，注册成功。
   1. **分支流**

没有。

* 1. **替代流**

E-1:如果不合法，系统显示输入信息不合法；如果不一致，系统显示密码和确认密码不一致。用户可以选择返回基流的起点，重新输入正确的信息；或者取消注册，用例结束。

E-2:如果存在，系统显示该用户名已存在，请重新输入。用户可以选择返回基流的起点，重新输入正确的信息；或者取消注册，用例结束。

### 2.4.2 登录用例图事件流描述

1. **前置条件**

没有。

1. **后置条件**

如果这个用例成功，参与者可以启动系统并使用系统所提供的功能。反之，系统状态不变。

1. **扩充点**

没有。

1. **事件流**
   1. **基流**

当用户希望登录账号时，用例启动。

1. 系统提示用户输入用户名、密码。
2. 系统判断输入信息是否合法，若合法(E-1)。
3. 系统验证用户名和密码是否正确，若正确(E-2)，登录成功。
   1. **分支流**

没有。

* 1. **替代流**

E-1:如果不合法，系统显示输入信息不合法。用户可以选择返回基流的起点，重新输入正确的信息；或者取消登录，用例结束。

E-2:如果不正确，系统显示账号或密码错误，请重新输入。用户可以选择返回基流的起点，重新输入正确的信息；或者取消登录，用例结束。

### 2.4.3 注销用例图事件流描述

1. **前置条件**

在这个用例开始之前，用户必须登录到系统中。

1. **后置条件**

如果这个用例成功，关闭本次会话。

1. **扩充点**

没有。

1. **事件流**
   1. **基流**

当用户希望注销时，用例启动。

1. 系统显示注销成功。
2. 系统自动跳转到登录页面。
   1. **分支流**

没有。

* 1. **替代流**

没有。

### 2.4.4 发布投票用例图事件流描述

1. **前置条件**

在这个用例开始之前，用户必须登录到系统中。

1. **后置条件**

如果这个用例成功，投票发布成功。

1. **扩充点**

没有。

1. **事件流**
   1. **基流**

当用户希望发布投票时，用例启动。

如果用户选择“单选投票”，则执行分支流S-1：单选投票。

如果用户选择“多选投票”，则执行分支流S-2：多选投票。

如果用户选择“多次投票”，则执行分支流S-3：多次投票。

* 1. **分支流**

S-1：单选投票

1. 填写投票标题、介绍。
2. 填写每一项的内容(可自行添加选项数量)。
3. 填写投票模式(非实名、实名，私密、公开，不公开结果、公开结果)。
4. 发布投票。
5. 检验是否合法，合法(E-1)，发布成功。

S-2：多选投票

1. 填写投票标题、介绍。
2. 填写每一项的内容(可自行添加选项数量)。
3. 填写最多选择数量。
4. 填写投票模式(非实名、实名，私密、公开，不公开结果、公开结果)。
5. 发布投票。
6. 检验是否合法，合法(E-1)，发布成功。

S-3：多次投票

1. 填写投票标题、介绍。
2. 填写每一项的内容(可自行添加选项数量)。
3. 填写最多投票数量。
4. 填写投票模式(非实名、实名，私密、公开，不公开结果、公开结果)。
5. 发布投票。
6. 检验是否合法，合法(E-1)，发布成功。
   1. **替代流**

E-1:如果不合法，系统显示输入信息不合法。用户可以选择返回基流的起点，重新输入正确的信息；或者取消发布，用例结束。

### 2.4.5 查看投票用例图事件流描述

1. **前置条件**

在这个用例开始之前，用户可以不登录到系统中。

1. **后置条件**

如果这个用例成功，查看投票或投票成功。

1. **扩充点**
2. 筛选已发布投票。
3. 投票。
4. **事件流**
   1. **基流**
5. 当用户希望查看(可筛选)投票时，用例启动。

如果用户状态不为“管理员”，则执行分支流S-1：查看公开投票。

如果用户状态为“管理员”，则执行分支流S-2：查看全部投票。

1. 当用户希望投票时，用例启动。

如果用户状态为“未登录”且投票模式为“非实名”或用户状态为“已登录”，则执行分支流S-3：可投票。

如果用户状态为“未登录”且投票模式为“实名”，则执行分支流S-4：不可投票。

* 1. **分支流**

S-1：查看公开投票

根据搜索条件(可筛选)查询已发布的公开投票。

S-2：查看全部投票

根据搜索条件(可筛选)查询已发布的所有投票。

S-3：可投票

1. 选择选项。
2. 投票。
3. 判断是否已经投票或投票次数超出限制，否(E-1)，投票成功。

S-4：不可投票

执行替代流E-1。

* 1. **替代流**

E-1:如果为是，系统显示已经投票或投票次数超出限制。返回基流的起点，用例结束。

### 2.4.6 维护投票用例图事件流描述

1. **前置条件**

在这个用例开始之前，用户必须登录到系统中。

1. **后置条件**

如果这个用例成功，删除、暂停、修改、导出投票成功。

1. **扩充点**

没有。

1. **事件流**
   1. **基流**

当用户希望维护投票时，用例启动。

如果用户选择“删除”，则执行分支流S-1：删除。

如果用户选择“暂停”，则执行分支流S-2：暂停。

如果用户选择“修改”，则执行分支流S-3：修改。

如果用户选择“导出”，则执行分支流S-4：导出。

* 1. **分支流**

S-1：删除

1. 提示用户是否确认删除，如果为是(E-1)。
2. 删除本次投票记录。

S-2：暂停

1. 提示用户是否确认暂停，如果为是(E-1)。
2. 暂停本次投票。

S-3：修改

1. 显示原本信息
2. 用户更改信息
3. 提示用户是否确认修改，如果为是(E-1)。
4. 修改本次投票。
5. 检验是否合法，合法(E-2)，修改成功。

S-4：导出

1. 导出全部投票结果。
   1. **替代流**

E-1:如果为否，什么也不做，用例结束。

E-2:如果不合法，系统显示输入信息不合法。用户可以选择返回基流的起点，重新输入正确的信息；或者取消修改，用例结束。

### 2.4.7 编辑个人信息用例图事件流描述

1. **前置条件**

在这个用例开始之前，用户必须登录到系统中。

1. **后置条件**

如果这个用例成功，用户信息修改成功。

1. **扩充点**

修改密码。

1. **事件流**
   1. **基流**

当用户希望编辑个人信息时，用例启动。

1. 显示用户信息。
2. 用户修改个人信息，
3. 提示用户是否确认修改，如果为是(E-1)。
4. 检验是否合法，合法(E-2)，修改成功。

当用户希望修改密码时，用例启动。

1. 提示用户输入原密码和新密码。
2. 用户点击修改按钮。
3. 检验原密码是否正确，正确(E-3)，修改成功。
   1. **分支流**

没有。

* 1. **替代流**

E-1: 如果为否，什么也不做，用例结束。

E-2: 如果不合法，系统显示输入信息不合法。用户可以选择返回基流的起点，重新输入正确的信息；或者取消修改，用例结束。

E-3:如果不正确，系统显示密码错误。用户可以选择返回基流的起点，重新输入信息；或者取消修改密码，用例结束。

# 第3章 系统静态模型

## 3.1 实体类

### 3.1.1 用户实体类

**私有属性：**

Integer id

用户ID

String name

用户名

String pwd

密码

Integer gender

性别 0女 1男

Integer year

出生年

List<Post> post

发布的投票

List<Record> record

投票的记录

**公共操作：**

User selectUserById(Integer id)

查询用户通过id

List<User> selectUser(User user)

查询用户(模糊)

Integer selectUserCount(User user)

查询用户总数(模糊)

Integer insertUser(User user)

新增用户

Integer updateUser(User user)

修改用户

Integer deleteUser(Integer id)

删除用户

### 3.1.2 发布投票实体类

**私有属性：**

Integer id

发布ID

Integer userId

用户ID

String title

标题

String introduce

内容

Integer pollType

投票类型 0单选 1多选 2多次

Integer openShow

是否可以搜索到 0公开显示 1不公开显示

Integer openPoll

是否可以不登录投票 0都可投票 1登录后可投票

Integer openResult

是否公开投票结果 0公开投票结果 1不公开投票结果

Integer max\_checkbox

多选投票的最多选择数量

Integer max\_poll

多次投票的最多投票数量

Integer postState

发布状态 0正常 1暂停 2删除

Date postDate

发布时间

List<Option> option

发布的投票选项

**公共操作：**

List<Post> selectPostByUserId(Integer id)

查询发布通过用户id

Integer selectPostCountByUserId(Integer id)

查询发布总数通过用户id

List<Post> selectPost(Post post)

查询发布(模糊)

Integer selectPostCount(Post post)

查询发布总数(模糊)

Integer insertPost(Post post)

新增发布

Integer updatePost(Post post)

修改发布

Integer deletePost(Integer id)

删除发布

### 3.1.3 投票选项实体类

**私有属性：**

Integer id

投票选项ID

String optionText

选项描述

Integer optionState

选项状态 0正常 1已删除

**公共操作：**

List<Option> selectOptionByPostId(Integer id)

查询选项通过发布id

Integer selectOptionCountByPostId(Integer id)

查询发布总数通过用户id

Integer insertOption(Option option)

新增选项

Integer updateOption(Option option)

更新选项

Integer deleteOption(Integer id)

删除选项

### 3.1.4 投票记录实体类

**私有属性：**

Integer id

记录ID

Integer userId

投此票用户的ID

Integer optionId

投票选项ID

Integer pollNum

投票数量

Date pollDate

投票时间

Integer pollIP

投票的IP地址

String pollMAC

投票的MAC地址

**公共操作：**

List<Record> selectRecordByUserId(Integer id)

查询投票记录通过用户id

Integer selectRecordCountByUserId(Integer id)

查询投票记录总数通过用户id

List<Record> selectRecord(Record record)

查询投票记录(模糊)

Integer selectRecordCount(Record record)

查询投票记录总数(模糊)

Integer insertRecord(Record record)

新增投票记录

Integer updateRecord(Record record)

更新投票记录

Integer deleteRecord(Integer id)

删除投票记录

## 3.2 界面类

### 3.2.1 主界面类

包含类：注册

包含类：注销

包含类：查看投票

包含类：查看我发布的投票

包含类：查看我投的票

包含类：查看我的信息

### 3.2.2 注册界面类

Register()

注册用户

### 3.2.3 登录界面类

Login()

用户登录

### 3.2.4 查看投票界面类

ShowPoll()

查看投票

SearchPoll()

搜索投票

包含类：发布投票

包含类：查看投票详情

### 3.2.5 发布投票界面类

PostPoll()

发布投票

### 3.2.6 查看投票详情界面类

ShowPollInf()

查看投票详情

Poll()

投票

### 3.2.7 查看我发布的界面类

ShowMyPost()

查看我发布的投票

包含类：查看投票详情

包含类：编辑投票

### 3.2.8 编辑投票界面类

ChangePoll ()

发布投票

ChangePoll()

暂停投票

ChangePoll ()

删除投票

### 3.2.9 查看我投的票界面类

包含类：查看投票详情

### 3.2.10 查看我的信息界面类

ChangeMyInf()

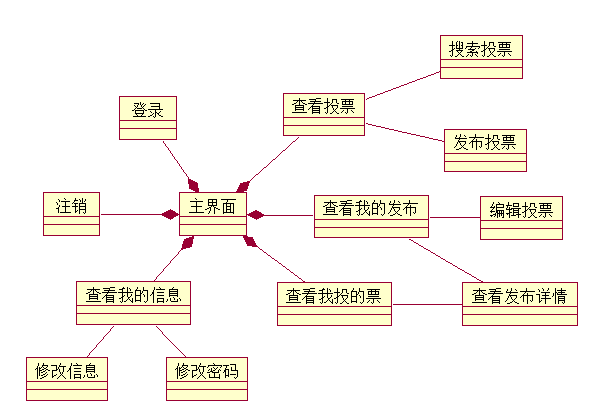
修改个人信息

ChangePwd()

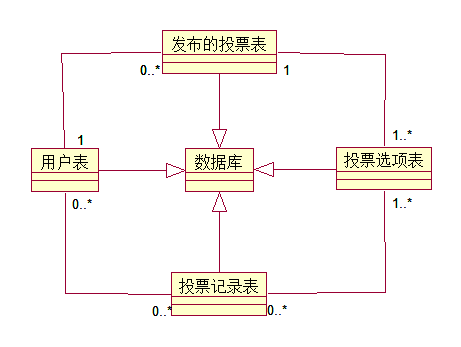
修改密码

## 3.3 类图

### 3.3.1 用户界面类图



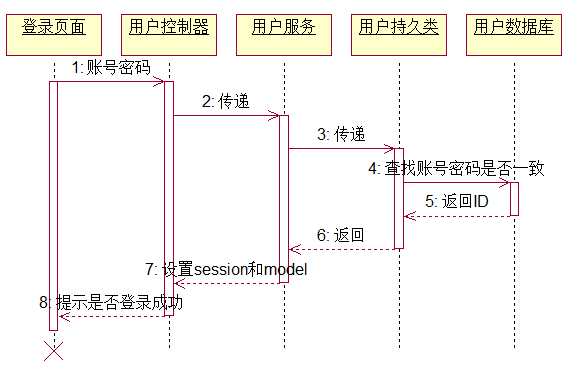
### 3.3.2 实体类图



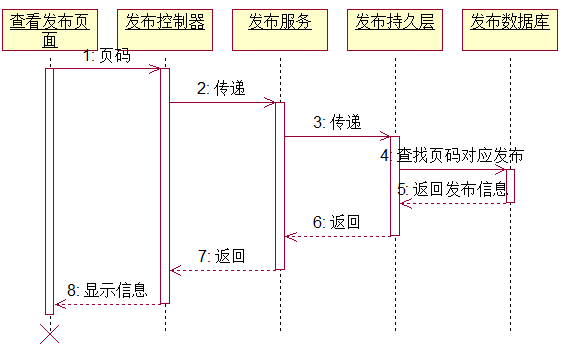
# 第4章 系统动态模型

## 4.1 顺序图

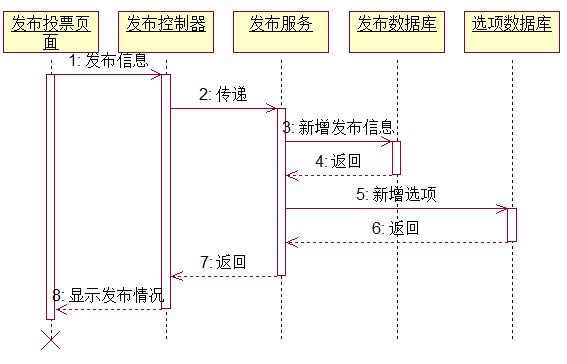
### 4.1.1 登录顺序图



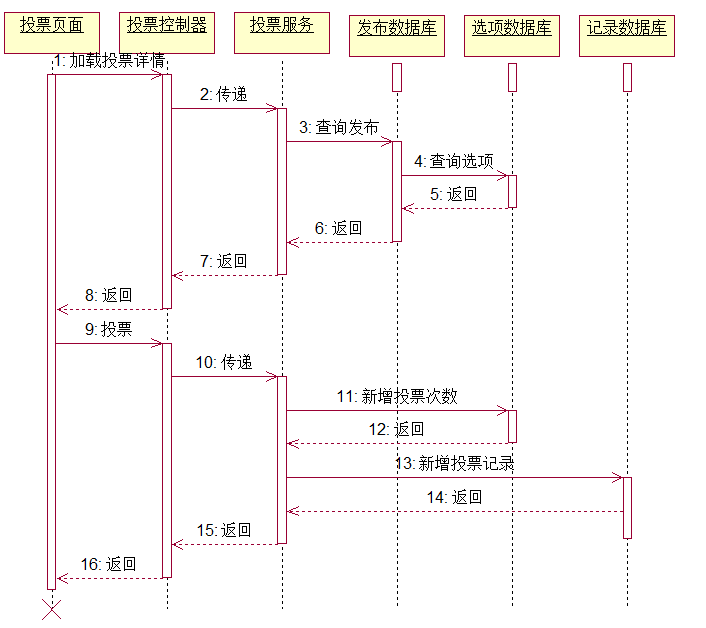
### 4.1.2 查看发布顺序图



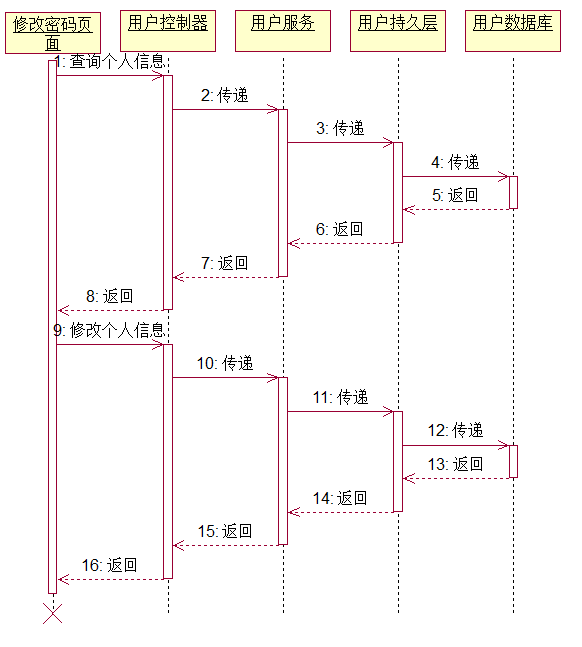
### 4.1.3 发布投票顺序图



### 4.1.4 投票顺序图

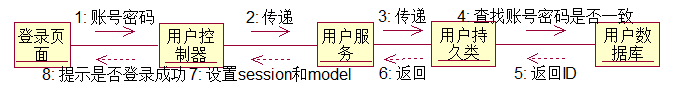


### 4.1.5 修改个人信息顺序图

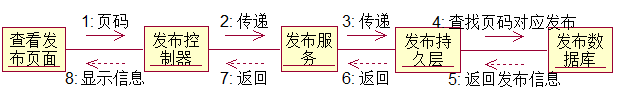


## 4.2 通信图

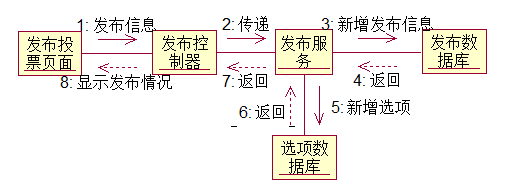
### 4.2.1 登录通信图



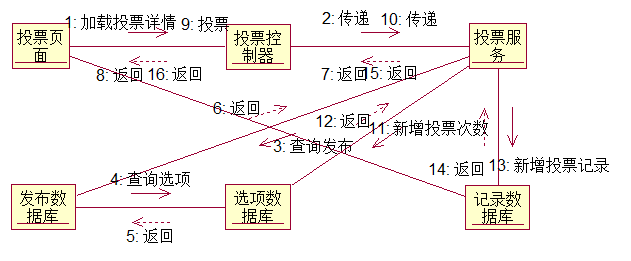
### 4.2.2 查看发布通信图



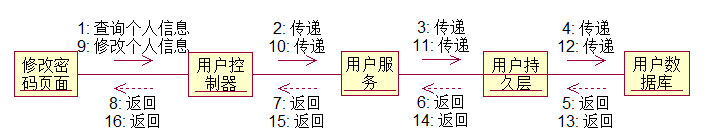
### 4.2.3 发布投票通信图



### 4.2.4 投票通信图

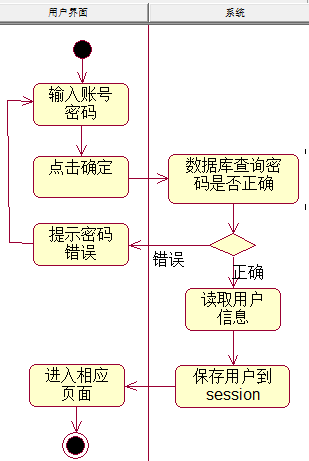


### 4.2.5 修改个人信息通信图

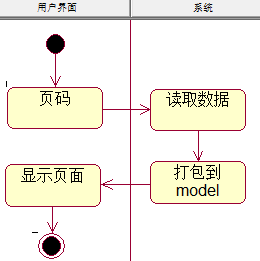


## 4.3 活动图

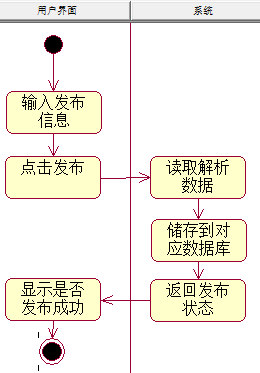
### 4.3.1 登录活动图



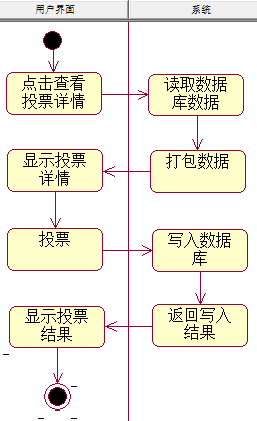
### 4.3.2 查看发布活动图



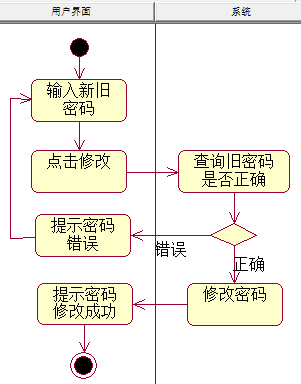
### 4.3.3 发布投票活动图



### 4.3.4 投票活动图

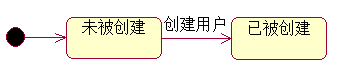


### 4.3.5 修改密码活动图

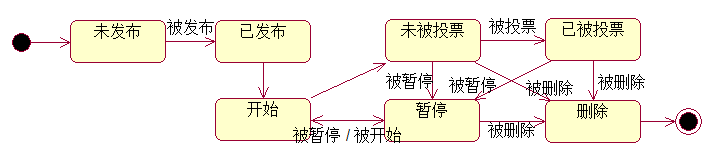


## 4.4 状态机图

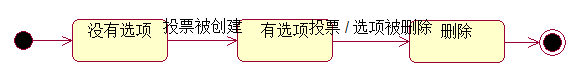
### 4.4.1 用户状态机图



### 4.4.2 发布的投票状态机图

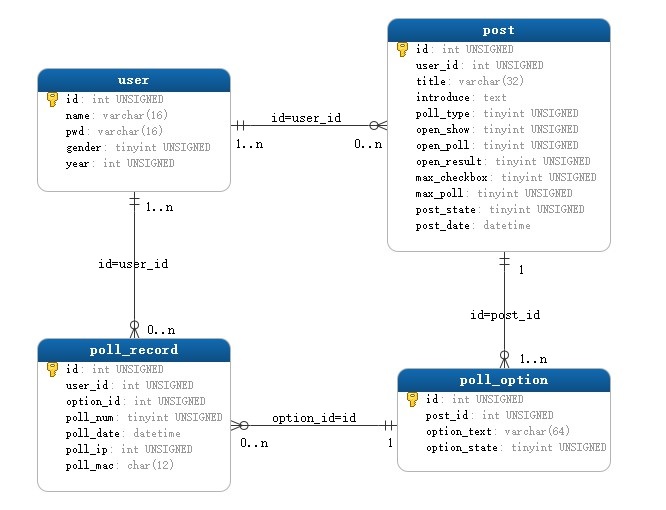


### 4.4.3 投票选项状态机图



# 第5章 数据库设计

## 5.1 E-R模型



## 5.2 表清单

### 5.2.1 用户表

表名user

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 备注 |
| id | int(11) unsigned | NOT NULL  AUTO\_INCREMENT | 编号 |
| name | varchar(16) | NOT NULL | 用户名 |
| pwd | varchar(16) | NOT NULL | 密码 |
| gender | tinyint(4) unsigned | NOT NULL  DEFAULT '1' | 性别  0女 1男 |
| year | int(11) unsigned | NOT NULL  DEFAULT '1990' | 出生年 |
| 约束 | PRIMARY KEY (`id`) | | |
| UNIQUE KEY `NAME` (`name`) | | |

### 5.2.1 发布的投票表

表名post

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 备注 |
| id | int(11) unsigned | NOT NULL  AUTO\_INCREMENT | 编号 |
| user\_id | int(11) unsigned | NOT NULL | 用户id |
| title | varchar(32) | NOT NULL | 标题 |
| introduce | text | NOT NULL | 内容 |
| poll\_type | tinyint(4) unsigned | NOT NULL | 投票类型  0单选 1多选 2多次 |
| open\_show | tinyint(4) unsigned | NOT NULL | 是否可以搜索到  0公开显示 1不公开显示 |
| open\_poll | tinyint(4) unsigned | NOT NULL | 是否可以不登录投票  0都可投票 1登录后可投票 |
| open\_result | tinyint(4) unsigned | NOT NULL | 是否公开投票结果  0公开投票结果 1不公开投票结果 |
| max\_checkbox | tinyint(4) unsigned | NOT NULL | 多选投票的最多选择数量 |
| max\_poll | tinyint(4) unsigned | NOT NULL | 多次投票的最多投票数量 |
| post\_state | tinyint(4) unsigned | NOT NULL | 发布状态  0正常 1暂停 2删除 |
| post\_date | datetime | NOT NULL | 发布时间 |
| 约束 | PRIMARY KEY (`id`) | | |
| KEY `user\_id` (`user\_id`) | | |
| FOREIGN KEY (`user\_id`) REFERENCES `user` (`id`) | | |

### 5.2.1 投票选项表

表名poll\_option

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 备注 |
| id | int(11) unsigned | NOT NULL  AUTO\_INCREMENT | 编号 |
| post\_id | int(11) unsigned | NOT NULL | 发布投票id |
| option\_text | varchar(64) | NOT NULL | 选项描述 |
| option\_state | tinyint(4) unsigned | NOT NULL | 选项状态  0正常 1已删除 |
| 约束 | PRIMARY KEY (`id`) | | |
| KEY `poll\_option\_ibfk\_1` (`post\_id`) | | |
| FOREIGN KEY (`post\_id`) REFERENCES `post` (`id`) | | |

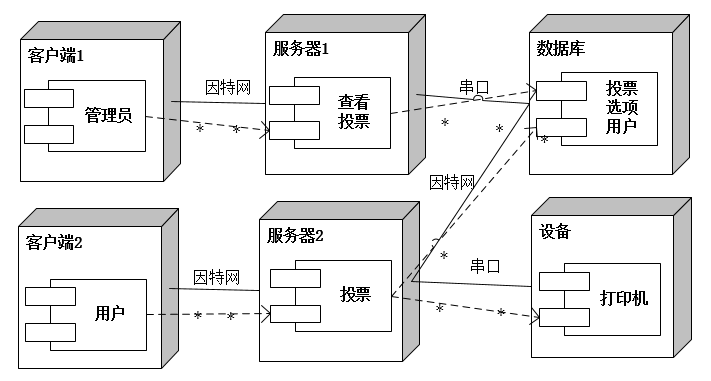
### 5.2.1 投票记录表

表名poll\_record

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 备注 |
| id | int(11) unsigned | NOT NULL  AUTO\_INCREMENT | 编号 |
| user\_id | int(11) unsigned | NOT NULL | 投此票用户的id |
| option\_id | int(11) unsigned | NOT NULL | 投票选项id |
| poll\_num | tinyint(4) unsigned | NOT NULL | 投票数量 |
| poll\_date | datetime | NOT NULL | 投票时间 |
| poll\_ip | int(4) unsigned | NOT NULL | 投票的IP |
| poll\_mac | char(12) | NOT NULL | 投票的MAC |
| 约束 | PRIMARY KEY (`id`) | | |
| KEY `user\_id` (`user\_id`) | | |
| KEY `option\_id` (`option\_id`) | | |
| FOREIGN KEY (`user\_id`) REFERENCES `user` (`id`) | | |
| FOREIGN KEY (`option\_id`) REFERENCES `poll\_option` (`id`) | | |

# 第6章 物理模型

## 部署图



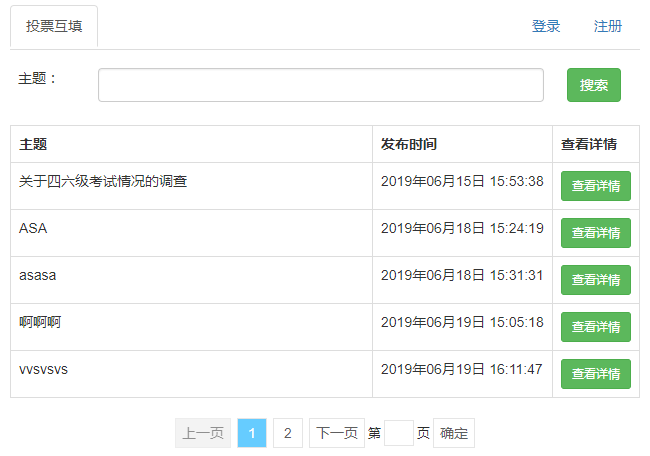
# 第7章 程序实现

## 7.1 注册、登录模块

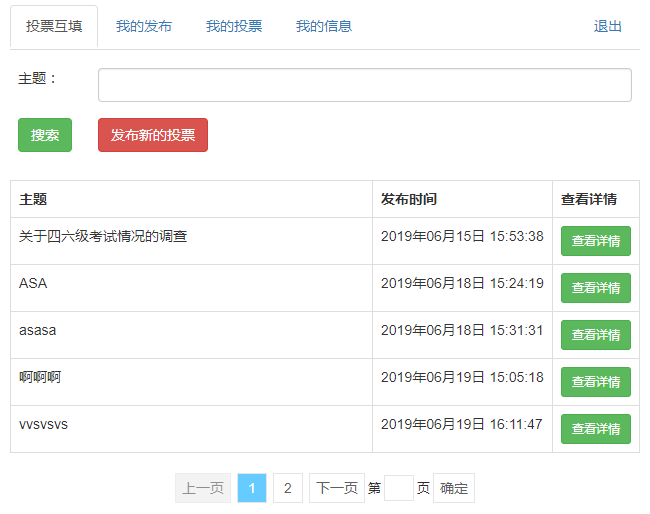


注册界面 登录界面

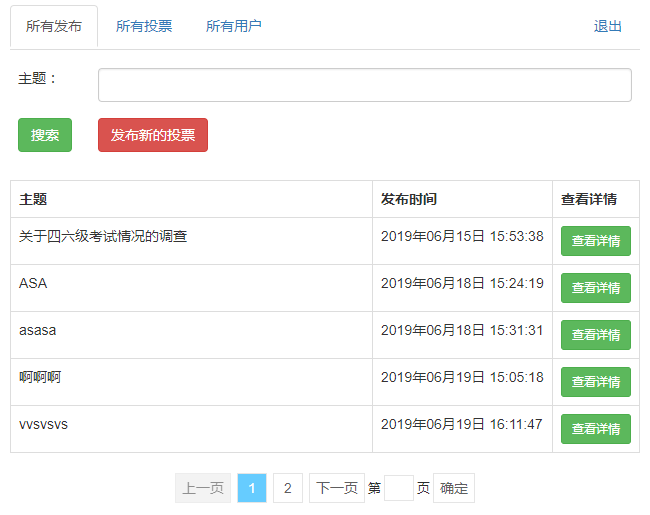
## 7.2 首页、查看投票模块



未登录者首页和查看投票界面



登录者(非管理员)首页和查看投票界面



管理员首页和查看投票界面



搜索投票页面

## 7.3 查看投票详情模块



查看投票详情界面

## 7.4 发布新的投票模块



发布新的投票界面

## 7.5 查看我的发布模块



查看我的发布界面

## 7.6 查看投票详情模块



查看投票详情界面

## 7.7 查看我投的票模块



查看我投的票界面

## 7.8 查看我的信息、修改密码模块



查看我的信息、修改密码界面

## 7.9 查看用户信息、批量导入导出用户模块



查看用户信息、批量导入导出用户界面(管理员才可以看到)

# 总结

通过本次投票系统的建模设计，我懂得了完整的从设计到开发的流程，对我以后在企业工作有很大的帮助。其中设计阶段，最难的是需求分析阶段，因为不知道自己分析出的需求是否合理，也不知别人是否满意自己做的系统，所以需要征集到大多数用户的需求，还需要整理归纳出可以实现的需求。其次是事件流的描述，他需要详细的描述事件的走向、分支、中断和结束等，要分析出各个状态以及操作。还有数据库的储存问题，需要合理、易储存、节省空间、易读取、方便查看等，需要斟酌考虑各个字段的类型和联系等。

对该模型进行设计，花费了大概一周的时间，期间还对以前的文档进行了细微的更改，对一些现在我的技术做不到的地方进行了删除或更改。总之，还是顺利完成了对该系统的设计，对自己的成品还是比较满意的。

学生签名：

年 月 日

# 附件

**数据库代码：**

CREATE TABLE `poll\_option` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT,

`post\_id` int(11) unsigned NOT NULL COMMENT '发布投票id',

`option\_text` varchar(64) NOT NULL COMMENT '选项描述',

`option\_num` int(11) unsigned NOT NULL DEFAULT '0' COMMENT '被投票数量',

`option\_state` tinyint(4) unsigned NOT NULL DEFAULT '0' COMMENT '选项状态 0正常 1已删除',

PRIMARY KEY (`id`),

KEY `poll\_option\_ibfk\_1` (`post\_id`),

CONSTRAINT `poll\_option\_ibfk\_1` FOREIGN KEY (`post\_id`) REFERENCES `post` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `poll\_record` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT,

`user\_id` int(11) unsigned NOT NULL COMMENT '投此票用户的id',

`option\_id` int(11) unsigned NOT NULL COMMENT '投票选项id',

`poll\_num` tinyint(4) unsigned NOT NULL DEFAULT '1' COMMENT '投票数量',

`poll\_date` datetime NOT NULL COMMENT '投票时间',

`poll\_ip` int(11) unsigned NOT NULL COMMENT '投票的IP',

`poll\_mac` char(12) NOT NULL COMMENT '投票的MAC',

`poll\_state` tinyint(4) NOT NULL DEFAULT '0' COMMENT '投票状态 0正常 1取消',

PRIMARY KEY (`id`),

UNIQUE KEY `user\_id\_2` (`user\_id`,`option\_id`) USING BTREE,

KEY `option\_id` (`option\_id`),

KEY `user\_id` (`user\_id`),

CONSTRAINT `poll\_record\_ibfk\_3` FOREIGN KEY (`option\_id`) REFERENCES `poll\_option` (`id`),

CONSTRAINT `poll\_record\_ibfk\_4` FOREIGN KEY (`user\_id`) REFERENCES `user` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `post` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT,

`user\_id` int(11) unsigned NOT NULL COMMENT '用户id',

`title` varchar(32) NOT NULL COMMENT '标题',

`introduce` text NOT NULL COMMENT '内容',

`poll\_type` tinyint(4) unsigned NOT NULL COMMENT '投票类型 0单选 1多选 2多次',

`open\_show` tinyint(4) unsigned NOT NULL DEFAULT '0' COMMENT '是否可以搜索到 0公开显示 1不公开显示',

`open\_poll` tinyint(4) unsigned NOT NULL DEFAULT '0' COMMENT '是否可以不登录投票 0都可投票 1登录后可投票',

`open\_result` tinyint(4) unsigned NOT NULL DEFAULT '0' COMMENT '是否公开投票结果 0公开投票结果 1不公开投票结果',

`max\_poll` tinyint(4) unsigned NOT NULL DEFAULT '0' COMMENT '最多选择/投票数量',

`post\_state` tinyint(4) unsigned NOT NULL DEFAULT '0' COMMENT '发布状态 0正常 1暂停 2删除',

`post\_date` datetime NOT NULL COMMENT '发布时间',

PRIMARY KEY (`id`),

KEY `user\_id` (`user\_id`),

CONSTRAINT `post\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `user` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `user` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT,

`name` varchar(16) NOT NULL COMMENT '用户名',

`pwd` varchar(16) NOT NULL COMMENT '密码',

`gender` tinyint(4) unsigned NOT NULL DEFAULT '1' COMMENT '性别 0女 1男',

`year` int(11) unsigned NOT NULL DEFAULT '1900' COMMENT '出生年',

PRIMARY KEY (`id`),

UNIQUE KEY `NAME` (`name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**部分关键代码：**

public class User {

private Integer id;

private String name; // 用户名

private String pwd; // 密码

private Integer gender; // 性别 0女 1男

private Integer year; // 出生年

private List<Post> post; // 发布的投票

private List<Record> record; // 投票的记录

public class Post {

private Integer id;

private Integer userId; // 用户id

private String title; // 标题

private String introduce; // 内容

private Integer pollType; // 投票类型 0单选 1多选 2多次

private Integer openShow; // 是否可以搜索到 0公开显示 1不公开显示

private Integer openPoll; // 是否可以不登录投票 0都可投票 1登录后可投票

private Integer openResult; // 是否公开投票结果 0公开投票结果 1不公开投票结果

private Integer maxPoll; // 最多投票数量

private Integer postState; // 发布状态 0正常 1暂停 2删除

private Date postDate; // 发布时间

private List<Option> option; // 发布的投票选项

public class Option {

private Integer id;

private Integer postId; // 发布投票id

private String optionText; // 选项描述

private Integer optionNum; // 被投票数量

private Integer optionState; // 选项状态 0正常 1已删除

public class Record {

private Integer id;

private Integer userId; // 投此票用户的id

private Integer optionId; // 投票选项id

private Integer pollNum; // 投票数量

private Date pollDate; // 投票时间

private Integer pollIp; // 投票的IP地址

private String pollMac; // 投票的MAC地址

private Integer pollState; // 投票的状态 0正常 1取消

public interface UserDao {

public Integer login(User user); // 用户登录

public List<User> select(User user); // 查询用户(模糊)

public User selectById(Integer id); // 查询用户通过id

public Integer insert(User user); // 新增用户

public Integer update(User user); // 修改用户

public Integer delete(Integer id); // 删除用户

public interface PostDao {

public List<Post> select(Post post); // 查询发布(模糊)

public List<Post> selectByUserId(Integer id); // 查询发布通过用户id

public Post selectById(Integer id); // 查询发布通过id

public Integer insert(Post post); // 新增发布

public Integer update(Post post); // 修改发布

public Integer delete(Integer id); // 删除发布

public interface OptionDao {

public List<Option> select(Option option); // 查询选项(模糊)

public List<Option> selectByPostId(Integer id); // 查询选项通过发布id

public Option selectById(Integer id); // 查询选项通过id

public Integer insert(Option option); // 新增选项

public Integer update(Option option); // 更新选项

public Integer delete(Integer id); // 删除选项

public Integer addNum(Option option); // 增加投票数量

public interface RecordDao {

public List<Record> select(Record record); // 查询投票记录(模糊)

public List<Record> selectByUserId(Integer id); // 查询投票记录通过用户id

public List<Record> selectByOptionId(Integer id); // 查询投票记录通过选项id

public Record selectById(Integer id); // 查询投票记数通过id

public Integer insert(Record record); // 新增投票记录

public Integer update(Record record); // 更新投票记录

public Integer delete(Integer id); // 删除投票记录

public interface UserService {

public Integer add(User user); // 注册

public Integer login(User user); // 登录

public Integer changePwd(String newPwd, User user); // 修改密码

public Integer change(User user); // 修改

public User checkById(Integer id); // 查询通过id

public PageInfo<User> check(User user, Integer page, Integer rows); // 查询(模糊)

public List<User> download(); // 下载

public interface PostService {

public Integer add(Post post); // 新增

public Integer delete(Integer id); // 删除通过id

public Integer change(Post post); // 修改

public Post checkById(Integer id); // 查询发布通过id

public PageInfo<Post> checkByUserId(Integer id, Integer page, Integer rows); // 查询通过用户id

public PageInfo<Post> check(Post post, Integer page, Integer rows); // 查询(模糊)

public interface OptionService {

public Integer add(Option option); // 新增

public Integer delete(Integer id); // 删除通过id

public Integer change(Option option); // 修改

public List<Option> checkByPostId(Integer id); // 查询通过发布id

public Option checkById(Integer id);

public Integer addNum(Option option); // 增加投票数量

public interface RecordService {

public Integer add(Record record); // 新增

public Integer delete(Integer id); // 删除通过id

public Integer change(Record record); // 修改

public PageInfo<Record> checkByUserId(Integer id, Integer page, Integer rows); // 查询通过用户id

public PageInfo<Record> check(Record record, Integer page, Integer rows); // 查询(模糊)