**采用MVC模式进行设计**

**一、控制层：**

主要设计的类有：

LoginController类：登录控制类

VoteController类： 投票控制类

AdminController类: 管理员控制类

WaitController类：等待页面控制类

1. **登录控制类：LoginController**

/\*

函数功能：登录验证

参数说明：第1个参数为传入参数：用户名，

第2 个参数为传入参数：密码

返回值： 投票界面或等待界面

函数流程：

1. 用户名、密码验证。成功到第（2）步，不成功跳转到（6）并提示。
2. 判断isLogin对象中对应的key是否存在，如果不存在则生成投票HashMap对象isLogin对应的key（value赋值为0）。0表示已经登录。如果存在，表示用户已经登录过则不修改isLogin中的值。跳转到（3）
3. 获取投票状态，投票状态voteStatus，值为0跳转到（4），值为1跳转到（5）

voteStatus，0表示投票进行中；1表示投票已经结束

1. 取数据模块：取投票界面数据，取阶段、轮次，进入投票界面

（5） 进入等待页面

（6） 返回登录界面

\*/

loginCheck(String username, String password)

1. **投票控制类：VoteController**

/\*

函数功能：用户投票

参数说明：第1个参数为传入参数：用户编号

第2个参数为传入参数：插入SQL语句

返回值： 投票界面或等待界面

函数流程：

1. 验证服务器状态voteStatus, 值为0跳转到（2），值为1跳转到（5）
2. 检查是否已经投票，如果已经投票跳转到（6），未投票跳转到（3）
3. 判断阶段、轮次是否正确。正确跳转到（4），不正确跳转到（7）
4. 插入数据，插入成功跳转到（5），插入失败，不做任何操作，返回提示信息给界面。
5. 修改isLogin对象对应的key对应的value值为1（表示已经投票）。跳转到（6）
6. 进入等待页面
7. 取数据模块：取投票界面数据，取阶段、轮次，进入投票界面

\*/

userVote(String userid, String strSql)

1. **AdminController类: 管理员控制类**

/\*

函数功能：初始化化系统参数

函数参数： 无

函数流程：

1. 清空isLogin对象内容
2. 数据库操作：

①清空投票信息表

②清空投票结果表

③生成第1阶段1轮次候选人（读候选人信息表）

1. 初始化application对象:

stage阶段初始化为1；

round轮次初始化为0；

voteStatus投票状态初始化为0；投票进行中

\*/

systemInit()

/\*

函数功能：第一阶段投票开始

函数参数：

函数返回值：

函数流程：

1. voteStatus投票状态0；投票进行中
2. round轮次+1
3. 生成投票候选人数据（写json文件）

\*/

firstStageStart()

/\*

函数功能：本轮次结束

函数参数：无

函数返回值：

函数流程：

1. voteStatus投票状态设为1（结束投票）
2. 统计候选人票数
3. 哪些人进入下一轮（被动）（生成jason文件）

（4）保存到数据库-投票结果表（JD LC 候选人编号 状态）

状态： 0 当选； 1 待选； 2 淘汰

\*/

roundEnd()

/\*

函数功能：下一轮开始

函数参数：无

函数返回值：

函数流程：

1. voteStatus投票状态0；投票进行中
2. round轮次+1
3. 生成投票候选人数据（写json文件）

\*/

nextRoundStart()

/\*

函数功能：下一阶段开始

函数参数：无

函数返回值：

函数流程：

1. stage值+1
2. 投票状态设置为0
3. 生成投票候选人数据（生成jason文件）
4. round设为1

\*/

nextStageStart()

1. **WaitController类：等待页面控制类**

/\*

函数功能：进入下一轮投票

函数参数：第1个参数为传入参数：当前阶段

第2个参数为传入参数：当前轮次

函数返回值：等待页面或投票页面

函数流程：

1. 获取系统阶段、轮次；用户阶段、轮次（界面层传过来）。

比较阶段、轮次；不相同跳转到（2），相同跳转到（3）

1. 取数据模块：取投票界面数据，取阶段、轮次（交给投票模块）。
2. 等待页面。

\*/

nextRound()

**二、业务层**

**LoginService类**

**VoteService类**

**ApplicationService类**

**WaitService类**

1. **登录业务类：LoginService**

/\*

函数功能：登录验证

参数说明：第1个参数为传入参数：用户名，

第2 个参数为传入参数：密码

返回值： 真或假

函数流程：直接调用DAO层的函数

\*/

loginCheck(String username, String password)

1. **投票业务类：VoteService**

/\*

函数功能：获取候选人名单

\*/

getCandidates()

/\*

函数功能：插入投票结果

参数说明：第1个参数为传入参数：用户编号，

第2 个参数为传入参数：插入SQL语句

返回值： true或false

\*/

insertVoteResult(String userid, String strSql)

1. **Application对象业务类：ApplicationService类**

成员变量：

voteStatus 投票状态：0表示投票进行中；1表示投票已经结束

HashMap对象isLogin：（用户编号，value） 1表明该用户已经投票，0表明该用户已经登录。

stage: 阶段

round: 轮次

成员方法：

所有成员变量对应的getter和setter方法

/\*

函数功能：判断用户是否登录过

函数参数：用户编号

函数流程：

判断isLogin对象中对应的userid是否存在，如果不存在则没有登录。如果存在表示用户已经登录过。

\*/

checkLogin(String userid)

/\*

函数功能：检测当前阶段当前轮次是否正确

参数说明：第1个参数为传入参数：界面层保存的阶段，

第2 个参数为传入参数：界面层保存的轮次

返回值：true或false，true表示阶段、轮次正确，false表示阶段、轮次不正确

函数流程：获取系统阶段、轮次；和界面层传过来阶段、轮次比较；

\*/

checkStageRound(String stage, String round)

/\*

函数功能：检测用户是否已经投票

函数参数：用户编号

函数返回值：true或false

函数流程：

判断HashMap对象isLogin中对应的用户编号值是否为1，1表明该用户已经投票

\*/

isVoteCheck(String userid)

/\*

函数功能：检测投票是否已经开始

函数返回值：true或false

函数流程：

获取voteStatus，判断其值是否为0，0表示投票进行中；1表示投票已经结束

\*/

voteStatusCheck()

**三、数据库访问层：**

**1、登录数据访问类：LoginDao**

/\*

函数功能：登录验证

参数说明：第1个参数为传入参数：用户名，

第2 个参数为传入参数：密码

返回值： 真或假

函数流程：调用数据库存储过程

\*/

loginCheck(String username, String password)

**2、投票数据库访问类：VoteDao**

/\*

函数功能：插入投票结果

参数说明：第1个参数为传入参数：用户编号，

第2 个参数为传入参数：插入SQL语句

返回值： true或false

\*/

insertVoteResult(String userid, String strSql)

/\*

函数功能：获取候选人名单

\*/

getCandidates()

**四、数据库DAO层**

根据数据库中的表创建对应的DAO访问对象，采用注释的方式添加DAO对象和表的映射关系。

**五、系统控制**

**过滤器**

/\*

函数说明：检测是否成功登录了。没有登录，不允许访问除登录页面的其他页面。如果登录了没有退出就提示用户要正常退出。

\*/

checkIsLogin()