目 录

[1引言 2](#_Toc418153312)

[1.1编写目的 2](#_Toc418153313)

[1.2背景 2](#_Toc418153314)

[1.3定义 2](#_Toc418153315)

[1.4参考资料 2](#_Toc418153316)

[2总体设计 2](#_Toc418153317)

[2.1需求规定 2](#_Toc418153318)

[2.2运行环境 7](#_Toc418153319)

[2.3基本设计概念和处理流程 7](#_Toc418153320)

[2.4结构 9](#_Toc418153321)

[2.5功能需求与程序的关系 12](#_Toc418153322)

[2.6人工处理过程 12](#_Toc418153323)

[2.7尚未解决的问题 12](#_Toc418153324)

[3接口设计 12](#_Toc418153325)

[3.1用户接口 12](#_Toc418153326)

[3.2外部接口 13](#_Toc418153327)

[3.3内部接口 13](#_Toc418153328)

[4运行设计 13](#_Toc418153329)

[4.1运行模块组合 13](#_Toc418153330)

[4.2运行控制 14](#_Toc418153331)

[4.3运行时间 14](#_Toc418153332)

[5系统数据结构设计 14](#_Toc418153333)

[5.1逻辑结构设计要点 14](#_Toc418153334)

[5.2物理结构设计要点 16](#_Toc418153335)

[5.3数据结构与程序的关系 17](#_Toc418153336)

[6系统出错处理设计 18](#_Toc418153337)

[6.1出错信息 18](#_Toc418153338)

[6.2补救措施 19](#_Toc418153339)

[6.3系统维护设计 19](#_Toc418153340)

**概要设计说明书**

# 1引言

## 1.1编写目的

本文档面向系统设计分析人员以及程序员。

文档的编写目的在于从整体上描述系统的设计框架，包括模块划分、处理流程和接口设计等部分，帮助系统分析人员和程序员更深入的理解系统的整体结构，有助于具体的编码实现以及之后优化改进。

## 1.2背景

说明

项目名称：选呗

任务提出者：小组成员刘雨轩提出创意，小组会投票决定

开发者：全体小组成员

项目描述：实现web和安卓应用，让有选择困难的用户通过其他人的投票和评论解决选择问题。

## 1.3定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

## 1.4参考资料

《“选呗”结构化分析报告》

《“选呗”软件需求说明书》

# 2总体设计

## 2.1需求规定

1. 系统功能

“选呗”软件应用系统需要具有以下几个系统功能：

注册，登录，展示投票，发起投票，执行投票（投票和评论），删除投票及评论，关闭投票，关注投票，查询投票，查询历史信息，管理员对投票评论管理，管理员对用户的管理。

主要功能模块的IPO表如下：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 选呗 | | | | | | | | | | | |
| 模块名：注册模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | 输出O： | | | | | | |
| 用户账号和密码 | | 1. 在用户文件中查找匹配的用户账号 2. 若存在，提示用户名已存在 3. 否则，将用户名和密码存入用户文件，返回登录页面 | | | 1. 用户名已存在 2. 登录页面 | | | | | | |
| 模块名：登录模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | | 输出O： | | | | | |
| 用户账号和密码 | | 1. 在用户文件中查找匹配的账号 2. 若不存在，提示找不到用户信息 3. 否则，匹配密码    1. 若密码不匹配，提示密码错误    2. 若匹配，成功登录 | | | | 1. 不成功登录时，输出出错信息 2. 成功登录时，输出用户信息 | | | | | |
| 模块名：展示投票模块 | | | | | | | | | | | |
| 输入I： | | | 处理P： | | | | 输出O： | | | | |
| 投票文件 | | | 1. 维护一个按时间排序的数组，存储投票对象 2. 按最近时间顺序输出 | | | | 若干投票信息 | | | | |
| 模块名：发起投票模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | | | | 输出O： | | | |
| 1. 用户账号 2. 待投票图片 3. 投票描述 | | 1. 分配新的投票序号 2. 将投票信息输出到投票文件 3. 根据用户账号在用户文件中找到用户信息，将投票序号添加到用户信息 | | | | | | 1. 将投票信息输出到投票文件 2. 将投票序号输出到用户文件 | | | |
| 模块名：执行投票模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | | | | 输出O： | | | |
| 1. 投票序号 2. 用户账号 3. 投票选择 4. 评论描述 | | 1. 根据投票序号找到投票文件，将对应的选择计数+1，并将用户账号写入投票文件 2. 如果有评论，根据投票序号，找到投票文件，分配评论序号，形成评论，写入投票信息（会有是否投过票的判断，若没有投过不允许评论） 3. 根据用户账号找到用户文件，将用户与投票的参与关系写入用户文件 | | | | | | 1. 投票选择和用户账号到投票文件 2. 投票序号到用户文佳 | | | |
| 模块名：删除投票和评论 | | | | | | | | | | | |
| 输入I： | | 输出P： | | | | | | | 输出O： | | |
| 1. 删除评论 | 删除投票的选择 2. 用户账号 3. 投票序号 4. 评论序号 | | if 删除评论 then 根据投票序号，找到投票文件，根据用户账号和评论序号将评论删除，写回文件  else 删除投票选择，根据投票序号，找到对应投票，根据用户名删除对应选择和评论，在用户文件中取消用户与投票的参与联系。 | | | | | | | 将结果写回投票文件和用户文件 | | |
| 模块名：关闭投票模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | | | | | | 输出O： | |
| 1. 用户账号 2. 投票序号 | | | 根据用户账号和投票序号，修改投票文件中对应投票的状态为结束状态，并输出最终结果。 | | | | | | | 投票最终结果 | |
| 模块名：关注投票模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | | | | | | 输出O： | |
| 1. 用户账号 2. 投票序号 | | 根据用户账号，将投票序号写入用户文件对应的关注投票中 | | | | | | | | 投票序号输出到用户文件 | |
| 模块名：查询投票模块 | | | | | | | | | | | |
| 输入I： | 处理P： | | | | | | | | | 输出O： | |
| 投票序号 | 根据投票序号从投票文件中获取信息并输出 | | | | | | | | | 投票信息 | |
| 模块名：查询历史模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | | | | | | 输出O： | |
| 用户账号 | | 根据输入的用户账号在用户文件中获取用户发起关注和参与的投票序号，从投票文件中取出输出 | | | | | | | | 若干投票信息 | |
| 模块名：投票和评论管理模块 | | | | | | | | | | | |
| 输入I： | | 处理P： | | | | | | | | 输出O： | |
| 1 管理员账户  2 投票序号+待删除评论序号  3 待删除投票序号 | | 验证管理员账号  删除评论  删除投票 | | | | | | | | 投票信息 | |
| 模块名：用户管理模块 | | | | | | | | | | | |
| 输入I： | | | | 处理P： | | | | | | | 输出O： |
| 1. 管理员账户 2. 待删除用户账号 | | | | 验证管理员账号，根据待删除用户账号，将该用户从用户文件中删除，并且把其参与的对应投票的内容删除 | | | | | | | 删除提示 |

1. 系统性能

1）精度

对于系统工作而言：输入数据形式主要有：用于投票的图片，对投票或者评论的描述文本，用户名和密码的字符串等；输出用于显示的数据形式主要有：投票图片，投票评论文本；传输过程中的数据形式主要是：用户名密码的字符串，描述的文本，以及投票评论等的序号。

精度要求：

用户名字符串长度在5-12位之间，密码长度大于6；

描述文本长度不超过140字；

投票和评论的序号为整数；

图片过大时要进行适当地压缩，便于存储。

2）时间特性要求

该软件系统的重点在与用户的交互，对时间的要求比较高。

要求对于用户的点击等操作需要作出及时响应，如投票，关注或者评论操作。在网络较好的情况下，响应时间应该在1s以内。

用户发起投票时，由于要进行图片的处理和存储以及文件的操作，所以要求处理时间在3s以内。

若等待时间超过10s，提示出错。

3）灵活性

需求改变：比如需要用户增加绑定社交账号，添加好友功能等，只需要在用户文件中加入特定的数据项，对用户的对象增加响应的处理即可，便于扩展。

操作形式：支持网页端和安卓手机端的操作。

接口：通过URL访问该软件应用系统。

4）可靠性

该软件涉及用户信息管理、投票信息管理等功能，需要具有一定可靠性，在开发过程中需要开发者尽量把错误消除在开发前期。首先对需求分析定义明确，然后在设计数据结构和编码时尽量避免错误，考虑好特殊情况和错误处理，再次需要设计完整正确的测试用例。另外，因此软件用户输入的部分主要通过点击，键入文字比较少，所以对非法输入的容错能力即软件健壮性比较容易保证。需要用户键入信息时，对精度做出明确规定，提前考虑错误输入的情况。

1. 输入输出要求

本产品为网站及网页式应用，输入数据全部为用户的请求（或者管理员的请求）。具体而言是输入为事件或者http请求。

用户可以将指向某一投票的“点击”作为输入，请求查看某一个具体的投票。经过浏览器和服务器的处理之后，返回一个页面作为输出。输出为对应的投票内容，主要包含各个选项，投票结果，用户评价，和各种用户操作接口。

在这个页面上，登录用户可以选择操作“投票”——即做出单选投票表示自己支持哪一方另外还可以做出文字评论。此时用户输入是投票选项和评论文字（评论可以为空），通过浏览器和服务器的处理，修改数据节点的数据（添加该用户的投票结果），并将更新过的该投票信息作为输出返回。

用户也可以发起投票。输入信息为两张以上的图片，对应的描述信息，无数据数值要求。成功输入并发起投票之后，后台修改了数据节点的信息（存储了这个新投票）。无输出。

用户信息管理方面，用户的输入是各种文字信息，无输出。有些会有格式要求，如密码，手机号，邮箱等。

1. 数据管理能力要求

数据规模与用户个数和投票个数有关。用户的信息有个人信息和历史记录，主要是文字和链接。单个用户的资料1kb级别的存储完全可以满足。

投票信息包括投票本身的文字和图片描述，以及所有的用户投票和评论。数据规模为图片数据大小\*2+投票数\*（显示用户信息+投票结果+用户评论）。图片按照1M每张计算，投票按照0.1k计算，每个投票（100人投票）大约为2M多，人数增加后存储增长也没有显著增加。

然而图片存储方面存在一定的挑战，存储会随着投票事件数量的增加而线性增加。

1. 故障处理要求

服务器宕机时要求能及时恢复。

1. 其他专门要求

该软件是一个面向用户的交互型应用，所以用户友好性十分重要。需要比较美观的界面。此外功能需要简洁易用，页面的链接指向和查找要合理。

1. 数据管理能力要求
2. 故障处理要求
3. 其他专门要求

## 2.2运行环境

1. 设备

预期分别开发出WebApp和AndroidApp，对硬件没有特殊的要求。

1. 支持软件

WebApp支持IE/Chrome/Firefox等浏览器的兼容，AndriodApp支持安卓版本2.3以上。

1. 接口

该软件系统相对独立，与其他系统基本没有联系。

1. 控制

用户以使用安卓App或者访问网页的形式，通过后台数据的处理来使用该软件系统。

## 2.3基本设计概念和处理流程

根据上一阶段结构化分析的结果，我们对此软件进行了结构化设计。以下为从DFD转化而来的初始MSD。



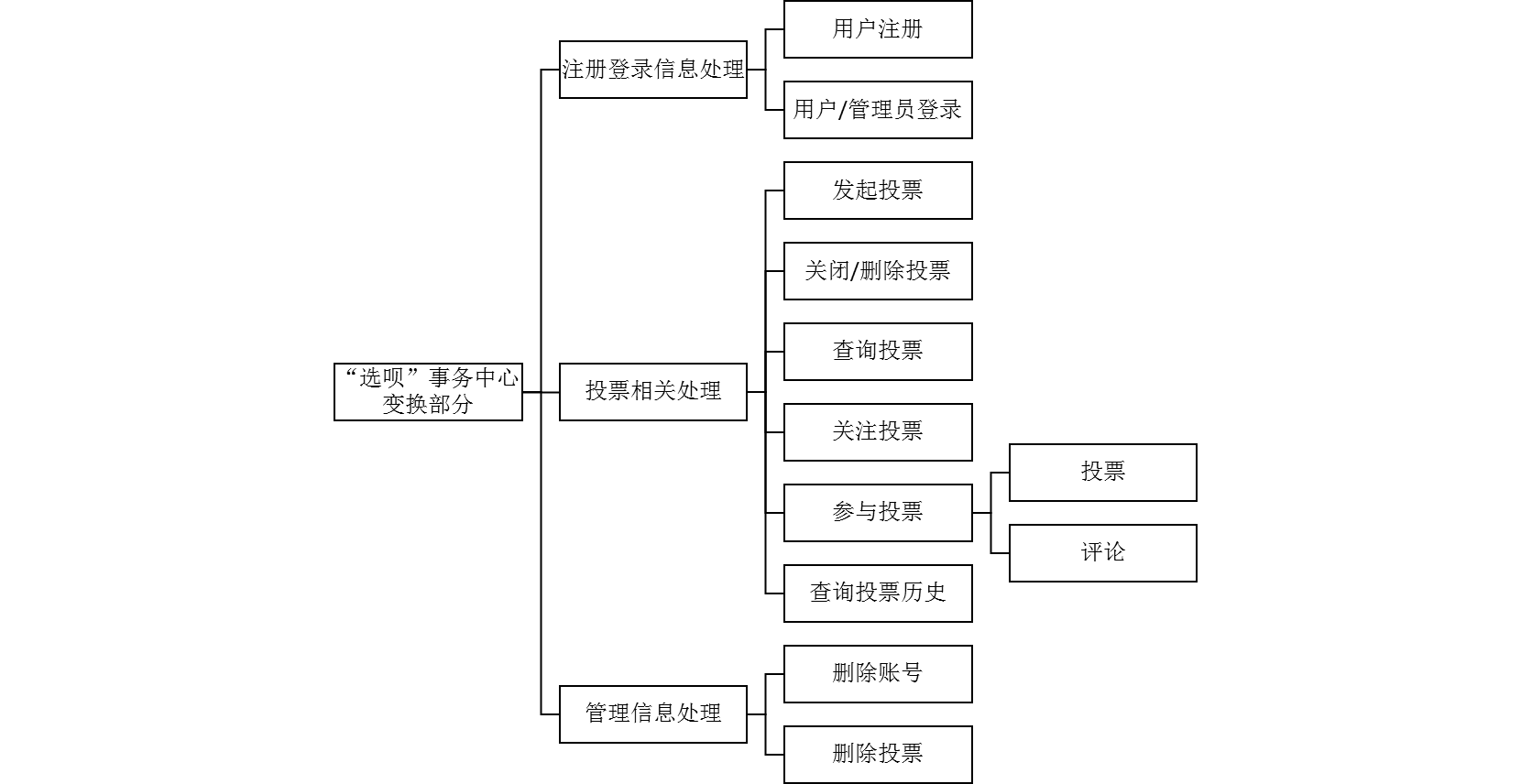
“选呗”应用系统事务中心可分为输入模块、变换模块和输出模块。输入模块读入管理员和用户的登录或注册信息，管理员管理的请求和信息，及投票相关信息（发起投票、参与投票、查询历史等）；变换模块将这些信息进行处理，反馈给事务中心登录和注册的结果、管理员管理结果及投票相关结果；输出模块将这些结果显示出来。

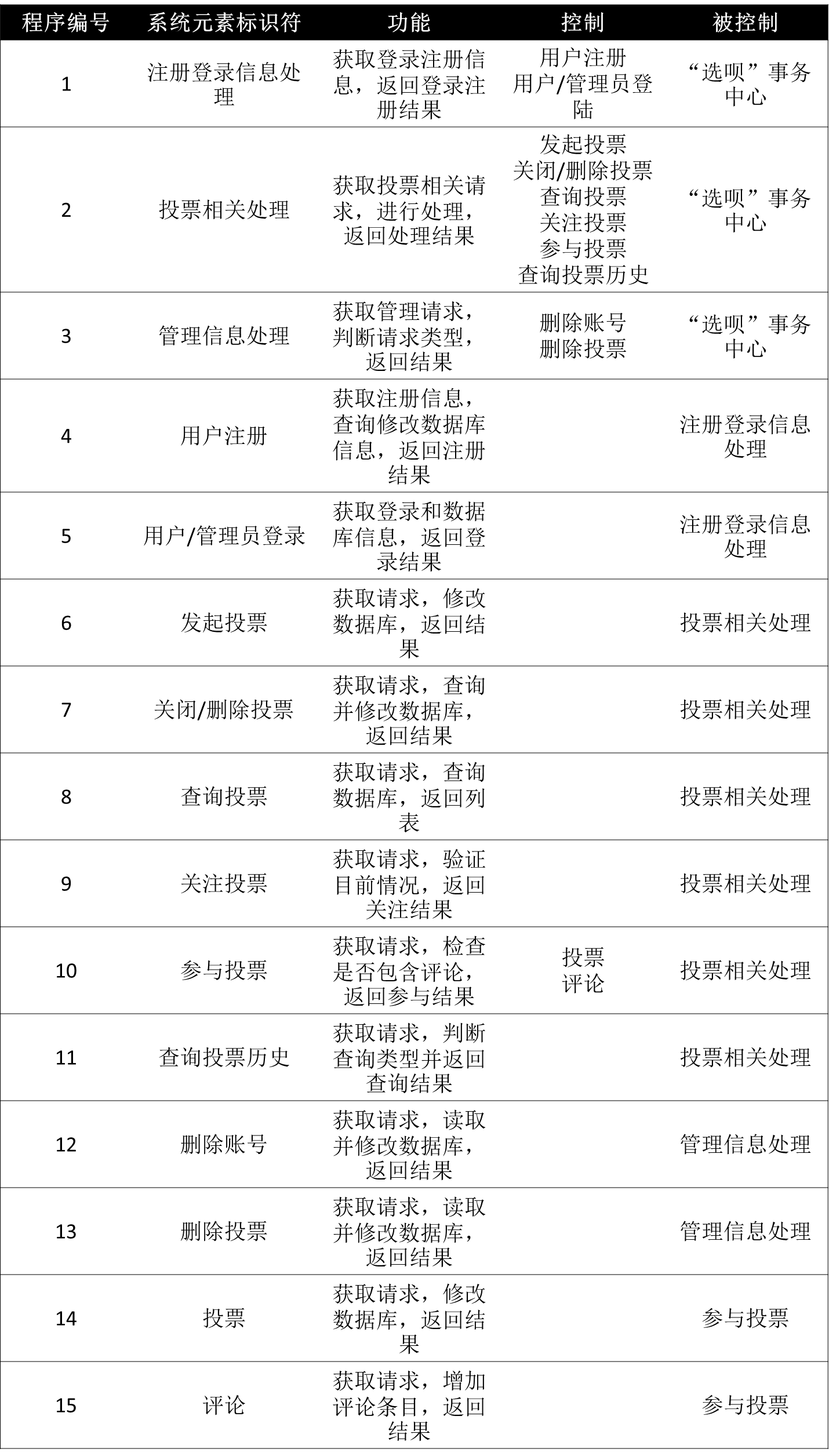
 以下为输入模块和变换模块的具体处理流程。

## 2.4结构

根据对初始MSD的精化，形成了可供详细设计的MSD，说明了本系统的各层模块的划分，并给出了各元素之间的控制与被控制关系，其中变换部分精化后的MSD如下。. 

各个模块具体层次关系和功能如下：





## 2.5功能需求与程序的关系

如下矩阵图说明了各项功能需求的实现同各块程序的分配关系：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 程序编号  系统功能 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 注册 | √ |  |  | √ |  |  |  |  |  |  |  |  |  |  |  |
| 登录 | √ |  |  |  | √ |  |  |  |  |  |  |  |  |  |  |
| 展示投票 |  | √ |  |  |  |  |  | √ |  |  | √ |  |  |  |  |
| 发起投票 |  | √ |  |  |  | √ |  |  |  |  |  |  |  |  |  |
| 执行投票 |  | √ |  |  |  |  |  |  |  | √ |  |  |  | √ | √ |
| 删除投票和评论 |  | √ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 关闭投票 |  | √ |  |  |  |  | √ |  |  |  |  |  |  |  |  |
| 关注投票 |  | √ |  |  |  |  |  |  | √ |  |  |  |  |  |  |
| 查询投票 |  | √ |  |  |  |  |  | √ |  |  |  |  |  |  |  |
| 查询历史 |  | √ |  |  |  |  |  |  |  |  | √ |  |  |  |  |
| 投票管理 |  |  | √ |  |  |  |  |  |  |  |  |  | √ |  |  |
| 用户管理 |  |  | √ |  |  |  |  |  |  |  |  | √ |  |  |  |

## 2.6人工处理过程

数据库建表和初始化。

软件日常维护。

在软件出错时开发人员需要从后台进行恢复。

## 2.7尚未解决的问题

无

# 3接口设计

## 3.1用户接口

|  |  |  |  |
| --- | --- | --- | --- |
| 命令 | 语法 | 信息正确 | 信息错误 |
| 登陆 | 点击事件 | 登陆成功，进主界面 | 显示错误，重新登陆 |
| 注册 | 点击事件 | 注册成功，进主界面 | 注册失败，重新注册 |
| 请求发布投票 | 点击事件 | 显示发布界面 | 提交请求失败 |
| 上传图片 | 点击事件 | 显示图片 | 显示失败 |
| 确认发送 | 点击事件 | 发布成功，查看投票 | 发布失败，回到发布页面 |
| 投票 | 点击事件 | 投票成功，显示投票结果 | 投票失败，回之前页面 |
| 查看投票结果 | 点击事件 | 显示投票结果 | 显示查看失败 |
| 关闭投票 | 点击事件 | 删除投票，给出删后页面 | 删除失败，回之前页面 |
| 关注投票 | 点击事件 | 显示关注成功 | 显示关注失败 |
| 查看关注的投票 | 点击事件 | 给出查看关注页面 | 显示查看失败 |
| 查看投票历史 | 点击事件 | 给出投票历史页面 | 显示查看失败 |

## 3.2外部接口

无

## 3.3内部接口

内部接口主要是client与sever端的接口。

注：与数据库的接口由框架提供

|  |  |  |
| --- | --- | --- |
| 函数 | 输入 | 返回 |
| UsrLoginCheck | 用户名，密码 | 是否登陆成功 |
| ManLoginCheck | 管理员名，密码 | 是否登陆成功 |
| Register | 注册的用户名，密码 | 是否增加成功 |
| Vote | 用户名，投票编号，投票 | 数据是否更新成功 |
| CheckVote | 投票编号 | 投票结果 |
| DeleteVote | 用户名，投票编号 | 是否删除成功 |
| DeleteUsr | 用户名 | 是否删除成功 |
| MarkVote | 用户名，投票编号 | 是否关注成功 |
| CheckMark | 用户名 | 用户关注列表 |
| CheckHistory | 用户名 | 用户投票历史 |

# 4运行设计

## 4.1运行模块组合

从整体上看，所有的操作的运行流程如下：

输入模块🡪变换模块🡪输出模块。

即输入模块收集用户及管理员输入，并对输入进行相关处理，将处理结果通过转化模块得到响应处理的结果，并转给输出模块进行输出。

对于特定的控制操作，会使用不同的输入、变化子模块，这些在模块介绍中有详细的描述，在此不赘述。

## 4.2运行控制

参见2.3处理流程

## 4.3运行时间

这一点现在并不确定，而且没有具体的要求。最后视实现技术而定。

# 5系统数据结构设计

## 5.1逻辑结构设计要点

数据存储为了方便管理，采用sql数据库，分成用户存储和投票存储两个部分。

四个主要的数据结构如下：

1. 用户数据 UsrData

用户编号-用户名-用户密码-注册时间-用户发起投票记录列表-用户关注记录列表-用户投票历史记录列表

1. 投票记录 VoteRecord

投票编号

1. 投票数据 VoteData

投票编号-发起时间-发起用户Id-投票选项列表

1. 投票选项 Choice

选项图片-选项票数

两部分的ER图如下：

**M**

用户

用户编号

用户名

用户密码

注册时间

发起

关注

参与

投票编号

**1**

**1**

**1**

**M**

**M**

投票记录

投票

投票编号

发起时间

发起用户

标识

选项图片

选项票数

**1**

**N**

投票选项

## 5.2物理结构设计要点

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据结构名 | 访问方法 | 存储区域 | 成员名 | 成员对应存储单位 |
| 用户数据  UsrData | sql语句 | 数据库 | 用户编号  UsrId | int |
|  |  |  | 用户名  Name | Char\* |
|  |  |  | 用户密码PassWord | int |
|  |  |  | 注册时间  ResgisterTime | int[3] |
|  |  |  | 发起投票记录  RaiseVote | VoteRecord\* |
|  |  |  | 关注投票记录  MarkedVote | VoteRecord\* |
|  |  |  | 投票历史记录  VoteHistory | VoteRecord\* |
| 投票记录  VoteRecord | Sql语句 | 数据库 | 投票编号  VoteId | int |
| 投票数据  VoteData | Sql语句 | 数据库 | 投票编号  VoteId | int |
|  |  |  | 发起时间  RaiseTime | int[3] |
|  |  |  | 发起用户Id  CalledUsrId | int |
|  |  |  | 投票选项列表  VoteChoice | Choice \* |
| 投票选项  Choice | Sql语句 | 数据库 | 选项图片  Picture | img |
|  |  |  | 选项票数  ChoiceNum | int |

## 5.3数据结构与程序的关系

|  |  |  |  |
| --- | --- | --- | --- |
| 函数 | 访问的数据结构 | 输入 | 返回 |
| UsrLoginCheck | UsrData | 用户名，密码 | 是否登陆成功 |
| Register | UsrData | 注册的用户名，密码 | 是否增加成功 |
| Vote | Usrdata,VoteData | 用户名，投票编号，投票 | 数据是否更新成功 |
| CheckVote | VoteData | 投票编号 | 投票结果 |
| DeleteVote | UsrData,VoteData | 用户名，投票编号 | 是否删除成功 |
| DeleteUsr | UsrData | 用户名 | 是否删除成功 |
| MarkVote | UsrData | 用户名，投票编号 | 是否关注成功 |
| CheckMark | UsrData | 用户名 | 用户关注列表 |
| CheckHistory | UsrData | 用户名 | 用户投票历史 |

# 6系统出错处理设计

## 6.1出错信息

|  |  |  |
| --- | --- | --- |
| **错误分类** | **子项** | **错误原因** |
| 数据库错误 | 连接 | 连接超时 |
| 连接断开 |
| 数据库本身错误 | 数据库代码错误 |
| 数据库溢出 |
| TCP连接错误 | 连接 | 连接超时 |
| 连接断开 |
| 其他TCP错误 | Socket自身错误 |
| 系统部分自定义错误 | 权限错误 | 管理员权限设置故障 |
| 输入错误 | 用户ID错误/为空 |
| 用户密码错误/为空 |
| 投票/评论ID错误 |
| 查找错误 | 未找到符合要求的记录 |
| 链接错误 | 内部链接错误 | 网页链接错误 |
| 文件/图片链接错误 |
| 外部链接错误 | 网页休整或者其他错误无法打开 |
| 友情链接网址维护或者其他错误无法打开 |

## 6.2补救措施

1. 后备技术

使用mysqldump以及Linux的定时功能（Cron），每隔固定时间将数据库导出备份。

1. 降效技术

无

1. 恢复及再启动技术

恢复重启时首先要确定错误的类型以及当前使用的数据库的状态，若当前数据库正常则将服务器正常重启，否则从备份数据库中恢复数据库，然后重新启动。

## 6.3系统维护设计

如果项目运行成功那么每天都会产生新的数据，也有可能发现新的bug。我们会增加用户反馈部分，并且每天监控程序的运行，保证网站和APP能够正常使用。