

目 录

1 引言	3
1.1 编写目的	3
1.2 背景	3
1.3 定义	3
1.4 参考资料	3
2 测试概要	4
单元测试	4
集成测试	8
确认测试	11
3 对软件功能的结论	12
3.1 发布投票	12
3.1.1 能力	12
3.1.2 限制	12
3.2 参与投票	12
3.2.1 能力	12
3.2.2 限制	12
3.3 查看投票	13
3.3.1 能力	13
3.3.2 限制	13
3.4 关闭投票	13
3.4.1 能力	13
3.4.2 限制	13
3.5 关注投票	13
3.5.1 能力	13
3.5.2 限制	13
3.6 个人资料修改	13
3.6.1 能力	13
3.6.2 限制	14

3.7 管理员相关操作	14
3.7.1 能力	14
3.7.2 限制	14
4 分析摘要	14
4.1 能力	14
4.2 缺陷和限制	14
4.3 建议	14
4.3.1	14
4.3.2	15
4.3.3	15
4.3.4	15
4.3.5	15
4.3.6	15
4.4 评价	15

测试分析报告

1 引言

1.1 编写目的

本文档是在项目具体代码完成过程中，“选呗”项目团队主要负责测试的人员记录在测试过程中发现的问题以及对于问题改正的建议。

本文档将供本系统的开发人员、维护人员查阅和使用。

1.2 背景

说明：

- a. 被测试软件系统的名称：“选呗”软件应用系统。
- b. 本软件开发者的：软件工程课第四组（王钰翔，周昊宇，欧阳逸群，田堃，陈可心，元桢，彭广举，刘雨轩）。
- c. 用户：网络端用户，适用于社会各行各业人群，无固定目标用户。
- d. 测试环境与实际运行环境之间可能存在的差异以及这些差异对测试结果的影响。
实际运行环境：Apache 服务器，PC 网页浏览器，手机浏览器。
测试环境：Apache 服务器，MySQL 5.6，chrome 浏览器。
测试直接在服务器上进行，网页端实现手机适配，但由于各个网页及手机浏览器显示设置不同，可能上线之后用户体验稍有差别。

1.3 定义

列出本文件中用到的专问术语的定义和外文首字母组词的原词组。

1.4 参考资料

列出要用到的参考资料，如：

- a. 《“选呗”软件需求说明书》；
- b. 《“选呗”概要设计说明书》；
- c. 王立福，孙艳春、刘学洋. 软件工程(第 3 版). 北京大学出版社, 2009 年

2 测试概要

测试采用的标准及技术

表 1 开始/中断/完成测试

开始/中断/完成测试	标准说明
开始测试标准	服务器环境可用且软件开发阶段性完成
中断测试标准	系统服务异常或数据库修改
完成测试标准	完成测试计划中的测试规划并达到程序和测试质量的目标, 由组长王钰翔确认

表 2 测试技术

测试技术	说明
编写测试用例	在编码阶段编写测试用例
单元测试	对具体每个函数接口进行测试
集成测试	检测模块集成, 参数传递和页面跳转
系统测试	找周围人试用系统, 检验对业务流程及数据流处理是否符合标准

表 3 测试类型

测试类型	说明
功能测试	根据《“选呗” 软件需求说明书》, 检验产品是否正确实现功能
边界值测试	选择边界数据进行测试, 输入不同数目文字、上传不同大小图片等, 确保系统功能正常, 程序无异常
界面测试	检查界面是否美观合理
文档测试	检查文档是否足够、描述是否合理

单元测试

1. 测试 `vote_model.php` 文件中的 `get_all_votes()`
`Vote_model.php` 在 `appliction` 目录下 `model` 文件夹下
函数声明为:

```
function get_all_votes()
```


函数预期实现的为返回所有的投票 ID
2. 测试 `vote_model.php` 文件中的 `get_options()`
`Vote_model.php` 在 `appliction` 目录下 `model` 文件夹下
函数声明为:

```
function get_options($vote_id)
```


函数预期实现的为返回指定投票 ID 的所有投票选项
3. 测试 `vote_model.php` 文件中的 `get()`
`Vote_model.php` 在 `appliction` 目录下 `model` 文件夹下

函数声明为:

```
function get_option_by_id($option_id)
```

函数预期实现的为返回指定投票选项的选项详情, 如果查询值不为 1 则返回 NULL

4. 测试 vote_model.php 文件中的 get_followed()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_followed($user_id)
```

函数预期实现的为返回指定用户 ID 关注的投票信息, 如果查询成功, 以数组形式返回查询结果, 否则返回一个空的数组

5. 测试 vote_model.php 文件中的 get_voted_record()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_voted_record($user_id, $vote_id)
```

函数预期实现的为返回指定用户 ID 和投票 ID 的投票参与记录, 如果有且仅有一条记录则返回, 否则返回 NULL

6. 测试 vote_model.php 文件中的 get_voted()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_voted($user_id)
```

函数预期实现的为返回指定用户 ID 投票的历史记录, 如果有则循环以数组形式输出, 否则输出空的数组

7. 测试 vote_model.php 文件中的 get_published()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_published($user_id)
```

函数预期实现的为返回指定用户 ID 发布的投票, 直接返回查询结果

8. 测试 vote_model.php 文件中的 get_comments ()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_comments($vote_id)
```

函数预期实现的为返回指定用户 ID 发布的评论, 直接返回查询结果

9. 测试 vote_model.php 文件中的 get_participate_num ()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_participate_num($vote_id)
```

函数预期实现的为返回指定投票 ID 被参与的次数, 直接返回查询结果的记录个数。

10. 测试 vote_model.php 文件中的 get_comment_num()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_comment_num($vote_id)
```

函数预期实现的为返回指定投票 ID 被评论的次数, 直接返回查询结果的记录个数。

11. 测试 vote_model.php 文件中的 get_follow_num ()

Vote_model.php 在 appliction 目录下 model 文件夹下

函数声明为:

```
function get_follow_num($vote_id)
```

函数预期实现的为返回指定投票 ID 被关注的次数，直接返回查询结果的记录个数。

12. 测试 vote_model.php 文件中的 close_test ()

Vote_model.php 在 application 目录下 model 文件夹下

函数声明为：

```
function close_test($user_id, $vote_id)
```

函数预期实现的为关闭指定用户 ID, 指定投票 ID 的对应的投票。如果关闭成功返回 0，否则返回-1；

13. 测试 vote_model.php 文件中的 get()

Vote_model.php 在 application 目录下 model 文件夹下

函数声明为：

```
function get($userid)
```

函数预期实现的功能为根据用户 ID 查询用户信息并返回查询结果，如果查询结果不唯一则返回 NULL；

14. 测试 vote_model.php 文件中的 update_profile()

Vote_model.php 在 application 目录下 model 文件夹下

函数声明为：

```
function update_profile(user_id)
```

函数预期实现的为更新指定用户的个人信息，如果更新成功不返回，否则返回-1 并打印失败信息

15. 测试 vote_model.php 文件中的 is_participated()

Vote_model.php 在 application 目录下 model 文件夹下

函数声明为：

```
is_participated($user_id, $vote_id)
```

函数预期实现的为更新

16. 测试 vote_model.php 文件中的 is_followed()

Vote_model.php 在 application 目录下 model 文件夹下

函数声明为：

```
function is_followed($user_id, $vote_id)
```

函数预期实现的为返回判断指定用户是否关注了指定投票，如果是则返回 1, 否则返回 0

17. 测试 vote_model.php 文件中的 vote()

Vote_model.php 在 application 目录下 model 文件夹下

函数声明为：

```
function vote($user_id, $vote_id, $option_id, $comment)
```

函数预期实现的为插入指定用户 ID 对指定投票的指定投票选项的投票信息并且插入一个可以为空值的评论信息。

详细测试结果参照测试页

http://123.57.204.246:8004/index.php/vote_model_test/test

http://123.57.204.246:8004/index.php/user_model_test/test

Vote_model 单元测试

表 4

序号	测试函数名	函数功能	是否成功	失败原因	是否更正
1	get_all_votes	按照插入时间返回所有投票信息，直接返回查询结果	是		/
2	get_options	输入投票ID，返回该投票的所有投票选项ID	是		/
3	get	输入投票ID, 如果只有1个查询结果。返回该投票的所有投票信息。如果结果不为1则返回NULL。	是		/
4	get_followed	根据用户ID查询关注过的投票。如果查询结果数量大于零则根据投票ID找到投票信息并循环以数组格式输出，否则输出空数组	是		/
5	get_voted_record	输入用户ID，查询用户参与过的投票，直接返回查询结果	是		/
6	get_voted	输入用户ID, 查询用户投票过的投票，直接返回查询结果	是		/
7	get_published	输入用户ID, 查询用户发起过的投票，直接返回查询结果	是		/
8	get_comments	输入投票ID, 查询对该投票的所有评论，直接返回查询结果	是		/
9	get_participate_num	输入投票ID, 查询该投票总的参与数量，直接返回查询结果	是		/
10	get_comment_num	输入投票ID, 查询该投票总的评论数量，直接返回查询结果	是		/
11	get_follow_num	输入投票ID, 查询该投票总的关注数量，直接返回查询结果	是		/
12	close_test	输入用户ID和投票ID, 关闭指定投票，如果投票已经被关闭则返回-1否则返回0并更新数据库信息	否	屡次测试会对数据库数据造成删除操作，第二	是

User_model 单元测试

表 5

序号	测试函数名	函数功能	是否成功	失败原因	是否更正
1	get	输入用户ID, 返回该用户的信息。如果查询结果不为1条则返回NULL, 否则返回查询结果	是		/
2	update profile	输入用户ID, 根据表单更新对应用户信息, 如果查询用户ID结果少于0则返回-1, 否则无返回值	是		/
3	is participated	输入用户ID和投票ID, 判断该用户是否参与过该投票。如果查询结果个数小于等于0则返回0, 否则返回1	是		/
4	is followed	输入用户ID和投票ID, 判断该用户是否关注过该投票, 如果查询结果个数小于等于0则返回0, 否则返回1	是		/
5	vote	输入用户ID, 投票ID, 投票选项ID和评论, 执行投票错左, 如果投票成功则返回0, 否则返回-1	是		/

集成测试

经过单元测试的各个模块, 组装在一起之后也可能会出现各种各样的问题, 本测试属于集成测试, 主要测试链接跳转和各个页面中参数传递是否正确。首先是链接跳转测试

表 6

编号	跳转链接名称	所在页面	测试结果	测试人	测试时间
1	网站主页	所有页面	通过	田堃	2015/6/16
2	搜索	所有页面	通过	田堃	2015/6/16
3	登录	主页 (未登录)	通过	田堃	2015/6/16
4	登录信息提交	主页 (未登录)	通过	田堃	2015/6/16
5	注册	主页 (未登录)	通过	田堃	2015/6/16

6	注册信息提交	signup.html	通过	田堃	2015/6/16
7	退出登录	所有页面（已登录）	通过	田堃	2015/6/16
8	我的首页	所有页面（已登录）	通过	田堃	2015/6/16
9	发布投票	所有页面（已登录）	通过	田堃	2015/6/16
10	提交新发布的投票	publish_vote.html	通过	田堃	2015/6/16
11	关闭投票	publishhistory.html	通过	田堃	2015/6/16
12	查看投票	主页 followhistory.html votehistory.html publishhistory.html	通过	田堃	2015/6/16
13	关注投票	主页 各投票详情页	未通过 （在主页点击之后应跳转到当前页面位置）	田堃	2015/6/16
14	取消关注	主页 各投票详情页 followhistory.html	未通过 （在主页点击之后应跳转到当前页面位置）	田堃	2015/6/16
15	参与投票	主页 各投票详情页	未通过 （在主页点击之后应跳转到当前页面位置）	田堃	2015/6/16
16	个人信息	usercenter.html	通过	田堃	2015/6/16
17	修改信息	userinfo.html	通过	田堃	2015/6/16
18	保存信息	Editprofile?.html	通过	田堃	2015/6/16
19	关注的投票	usercenter.html	通过	田堃	2015/6/16
20	参与的投票	usercenter.html	通过	田堃	2015/6/16
21	发布的投票	usercenter.html	通过	田堃	2015/6/16

接着是参数传递测试。具体步骤是将各个传参链接的预设行为和实际表现进行比较，得到测试结果。本测试使用深度优先自顶向下方法。测试结果如下：

表 7

编号	参数传递链接	所在页面	预期结果	发现问题	测试结果
1	注册	signup	空白的邮箱、用户名，密码输入以及确认框		通过

	注册成功	index 主页	数据库中加入新用户信息，可用于登录		通过
2	用户登录	导航栏弹窗	空白的用户名、密码登录窗		通过
	登录成功	index 主页	以已登录身份进入原浏览的页面	在查看投票时也跳转到主页	未通过
3	进入首页	所有页面	看到首页投票的展示，根据是否登录决定能否投票关注		通过
4	进入个人主页	/page/usercenter	该用户头像、用户名、签名，个人信息页展示		通过
5	查看个人信息	/page/userinfo	个人信息：账号、邮箱、签名；其他信息：生日、性别、所在地。已有则如实展示，否则为空。		通过
6	修改个人信息	/page/editprofile	与编号 4 中相同的个人信息。		通过
	修改成功	/page/userinfo	将修改如实更新并在“查看个人信息”中显示		通过
7	关注的投票	/page/followhistory	关注的每一个投票的详细信息（带链接及取消关注）		通过
	取消关注	/page/followhistory	在数据库中删除这一关注信息，并在“关注的投票”中体现出来		通过
8	参与的投票	/page/votehistory	参与的每一个投票的详细信息（带链接及用户投票选项及留言）		通过
9	发布的投票	/page/publishhistory	发布的每一个投票的详细信息（带链接及关闭投票功能）		通过
	关闭投票	/page/publishhistory	将相应投票关闭，并在所有用户浏览投票时不再拥有投票的权利。		通过
10	搜索投票	/page/search/?search=萌（例：以“萌”为关键字）	与“萌”相关的所有开放的和关闭的投票的预览		通过

11	查看投票详情	/vote/view/22 (以 22 号投票为例)	该投票的名称、发起者、各个选项详情、用户投票与评论。体现了发布投票、投票、评论、关注的结果		通过
12	关注投票及取消关注	有投票预览的页面(index 与搜索结果页面)、投票详情页面、个人信息“关注的投票”(仅供取消关注)	后台数据库中对应关注\取消,可在浏览投票,个人信息等页面中的操作体现出是否关注		通过
13	投票给某选项并评论	有投票预览的页面(index 与搜索结果页面)、投票详情页面	后台数据库中添加对应记录,可在浏览投票,个人信息等页面中的体现出来		通过
14	发布投票	/vote/publish_vote	空的投票标题、描述,空的 2 到 4 个选项的名称、描述及图片		通过
	发布成功	/vote/submit_vote	可被查看的带有上述文字及图片的详情投票	偶尔会发布失败	未通过
15	退出登录	所有页面	返回主页,浏览时不再带有个人登录状态,无法投票关注查看个人信息		通过

确认测试

确认测试人员是全组组员和一些内部测试用户,测试过程中用户体验反馈不错,期间发现了手机端适配顶栏布局错乱,手机上传头像失败,手机注册布局错乱等问题,都已经一一解决。

表 8

阶段	名称	测试结果	备注
1	用户登录注册	测试成功	无
1	发布投票	测试成功	无
1	查看热门投票列表	测试成功	无
1	查看投票	测试成功	无
2	查询投票	测试成功	无
2	参与投票	测试成功	无

2	修改个人资料	测试成功		无
2	查看发布投票历史	测试成功		无
2	查看参与投票历史	测试成功		无
2	关注投票	测试成功		无
2	查看关注投票列表	测试成功		无
2	关闭投票	测试成功		无
2	管理员操作	测试成功		包括删除投票、投票结

3 对软件功能的结论

3.1 发布投票

3.1.1 能力

能够支持两张到四张照片的上传及预览，
简述该项功能，说明为满足此项功能而设计的软件能力以及经过一项或多项测试已证实的能力。

3.1.2 限制

投票支持的最大投票选项数量为 4。

3.2 参与投票

3.2.1 能力

能够对投票进行选项投票操作，在投票后可以选择附加相应评论。

3.2.2 限制

在主页参与投票后不能返回原位置，只能跳转到页首

3.3 查看投票

3.3.1 能力

支持查看所有已经发布投票，另一方面能通过搜索显示所有相关投票。

3.3.2 限制

不能对结果分页，所以如果有大量结果，滑动查看结果的效果和体验可能会比较差

3.4 关闭投票

3.4.1 能力

支持在个人发布的历史投票中关闭指定投票

3.4.2 限制

关闭投票后不能恢复。

3.5 关注投票

3.5.1 能力

支持对所有投票(包括已经关闭的投票)的关注，关注结果显示在个人关注页

3.5.2 限制

关注结束后会直接返回页首，用户的一般习惯是停留在被关注的投票位置
在主页关注投票后不能返回原位置，只能跳转到页首

3.6 个人资料修改

3.6.1 能力

支持用户修改头像，用户名，以及签名信息

3.6.2 限制

上传头像时不支持用户对头像的剪切修改

3.7 管理员相关操作

3.7.1 能力

支持对指定用户、投票和评论的删除操作

3.7.2 限制

管理员的登录和设置都没有做，也没有跳转到管理员的链接，管理员界面也比较简单

4 分析摘要

4.1 能力

本软件实现的用户基本操作包括：登录注册、发布投票、参与投票(投票以及评论)、关注投票(取消投票)、关闭投票、修改个人资料、历史查看以及搜索投票的功能。实现的管理员操作包括删除用户及所有个人记录，删除投票信息，删除评论信息。

4.2 缺陷和限制

本软件的缺陷及限制如下：登入注册界面不能管理员登入、发布投票限制投票选项在2~4之间、用户上传图片大小未加限制、修改个人资料上传头像不支持图片剪切、所有投票的显示没有分页。实现的管理员操作界面缺陷主要是界面没有做优化，也没有相应的管理员管理数据库。

4.3 建议

4.3.1

增加用户管理员数据库储存用户管理员信息

4.3.2

增加投票选项上限，支持多图上传，在显示投票时较多图片以滚动方式显示

4.3.3

增加上传图票大小限制，超出限制返回提醒用户图片过大

4.3.4

在个人资料修改页面增加上传头像剪切功能

4.3.5

显示投票时以分页形式显示，在页尾显示页码信息。

4.3.6

优化管理员界面，在管理员界面增加搜索功能。

4.4 评价

该软件已经完成基本功能，没有严重错误，可以交付使用