

目 录

1 引言	2
1.1 编写目的.....	2
1.2 背景.....	2
1.3 定义.....	2
1.4 参考资料.....	3
2 任务概述	3
2.1 目标.....	3
2.2 运行环境.....	3
2.3 假定和约束.....	4
3 需求规定	4
3.1 对功能的规定.....	4
3.2 对性能的规定.....	7
3.2.1 精度.....	7
3.2.2 时间特性要求.....	8
3.2.3 灵活性.....	8
3.3 输入输出要求.....	8
3.4 数据管理能力要求.....	9
3.5 故障处理要求.....	9
3.6 其他专门要求.....	9
4 运行环境规定	9
4.1 设备.....	9
4.2 支持软件.....	10
4.3 接口.....	10
4.4 控制.....	10

软件需求说明书

1 引言

1.1 编写目的

该文档首先给出了我们的应用软件系统的一个整体的任务概述，之后从功能、性能等方面对需求的规定做了详细的说明，最后指出了对运行环境的规定。

其中对于功能的需求采用了 IPO 表的格式，以表格的方式详细地说明了每个功能模块的输入、处理和输出的过程。这些文字和图表都是为了准确地描述用户的需求，同时也为用户更容易理解这些需求创造了条件。

该文档的规格说明是进行设计的基础，也是进行软件测试的依据，同时也是用户确定软件需求的主要依据。

预期的读者是小组的各成员和用户，具体如下：

- 1) 项目经理：根据该文档了解预期产品的功能，并依此进行系统设计和项目管理。
- 2) 设计人员：对需求分析，并进行设计。
- 3) 开发人员：了解系统的需求等方面的规格说明，便于开发。
- 4) 测试人员：根据文档描述，编写测试用例，进行功能性和非功能性测试。
- 5) 用户：了解预期产品的功能和性能，提出改进意见。

1.2 背景

- a. 软件系统名称：选呗
- b. 任务提出者是小组成员刘雨轩，并由小组开会投票决定；开发者为小组各成员；用户是任何有选择困难症的人群和乐于帮助别人做出选择的人群。

1.3 定义

IPO 表：结构化设计中变换型结构的输入（Input）、加工（Processing）、输出（Output）对应的表。

Use Case 图：表示用况、参与者以及它们之间联系的图，用于需求捕获。

1.4 参考资料

项目开发计划书，Use Case 图及文档，结构化分析文档。

2 任务概述

2.1 目标

本软件名称为“选呗”，是一项意图通过其他用户的意见和评价，解决某用户选择困难状况的产品。本产品的应用目标，首先是在日常选择上遇到困难的人。人在生活中会遇到很多困难和选择，然而并不是所有选择都是深刻的、本质的、需要独自去面对的，这些小选择小纠结，在群体智慧的帮助下，可以得到轻松而完满的解决。在两件衣服面前举棋不定、在两本自学教材之间犹豫不决、在两个学妹……这时大众的偏好起码可以让你知道大家会喜欢什么，至少在迷茫的时候也不会做出太差的选择；如果几个选择支持者差不多，说明可能这些选项本来就难分高下，选什么都好不必担心；甚至理想情况下，一些有经验有知识的其他热心用户，可以通过评价的方式，给出一针见血的意见指出利害关系，从而帮助你一举解决困惑。第二个应用目标，是一些乐于在网络平台上发出自己的声音来帮助他人的人。百度知道和知乎可以如此火热，关键在于那些愿意表达自我，以自身的意志对他人散发影响，以自己的知识经历来帮助他人的人，而不是其他物质的激励机制。“选呗”也是一样，它的一部分目标用户，是那些愿意在别人困惑的时候提出建议的人，或者是好奇别人在纠结什么困惑什么的人。对他们来说“选呗”是日常消遣的一部分。值得注意的是，这两部分目标用户是有重叠的——或许自己在面对选择的时候有多么无力，在帮助别人进行选择的时候就有多么热心。

本产品是一项独立的软件且全部内容自含，自身即是一个完备的整体，并不依赖于其他软件、平台或入口。或许会和一些社交平台有联系，比如说分享、转发，但是并不代表本软件是其他软件的组成部分。

2.2 运行环境

硬件环境：需联网环境。

软件环境：IE 或其他浏览器（网页版“选呗”）

Andriod 系统（手机版“选呗”）

用户特点：用户为具有选择困难症、乐于帮助别人做出选择和想要了解别人面临选择来帮助自己做出选择的人群。

2.3 假定和约束

开发时间限制：2015.4 至 2015.6
本软件的开发需要有后台服务器支持，用于数据的管理和用户请求的响应。

3 需求规定

3.1 对功能的规定

我们首先用 Use Case 图进行了需求捕获，然后进行了结构化分析。根据结构化分析得到的数据流图，数据字典和加工小说明，将我们“选呗”软件应用系统的功能划分为以下几个模块：注册模块，登录模块，展示投票模块，发起投票模块，执行投票模块，删除执行的投票评论模块，投票评论管理模块，关闭投票模块，关注投票模块，查询投票模块，查询历史模块，用户管理模块。

参考 Use Case 和结构化分析的结果，画出主要功能模块的 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	<ol style="list-style-type: none"> 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 <ol style="list-style-type: none"> 3.1 若密码不匹配,提示密码错误 3.2 若匹配,成功登录 	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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:
<ol style="list-style-type: none"> 1. 用户账号 2. 待投票图片 3. 投票描述 	<ol style="list-style-type: none"> 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 	<ol style="list-style-type: none"> 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:
<ol style="list-style-type: none"> 1. 投票序号 2. 用户账号 3. 投票选择 4. 评论描述 	<ol style="list-style-type: none"> 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 	<ol style="list-style-type: none"> 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

	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	
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

3.2

对性能的规定

3.2.1 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。

3.2.2 时间特性要求

该软件系统的重点在与用户的交互，所以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。等的时间超过 10s 时，提示出错。

3.2.3 灵活性

1. 对于需求的改变：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。
2. 对于操作形式：支持网页端和安卓手机端的操作。
3. 对于接口：通过 URL 访问该软件应用系统。

3.3 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。

3.4 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。

图片存储怎么办啊怎么办啊怎么办啊。存储会随着投票事件数量的增加而线性增加啊怎么办啊怎么办。

3.5 故障处理要求

服务器宕机？要求能容易恢复。

3.6 其他专门要求

因为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等。

4 运行环境规定

4.1 设备

预期分别开发出 WebApp 和 AndroidApp，对硬件没有特殊的要求。

4.2 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。

4.3 接口

该软件系统相对独立，与其他系统基本没有联系。

4.4 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。