

软件需求说明书

——《赛搜搜》

0. 目录

软件需求说明书	
——《赛搜搜》	
0. 目录	
1. 背景介绍	
1.1 愿景	
1.2 理念	
2. 功能介绍	
2.1 导航栏	
2.2 推荐界面	
2.3 搜索界面	
2.4 竞赛信息	
2.5 个人界面	
3. 技术介绍	
3.1 小程序开发	
3.1.1 关于小程序的技术优势	
3.1.2 界面设计	
3.1.3 推荐功能实现	
3.1.4 筛选功能实现	
3.1.5 用户浏览记录及关注内容实现	
3.1.6 云开发技术应用	
3.2 数据爬取	
3.2.1 云开发	
3.2.2 爬虫	

1. 背景介绍

在校大学生，无论是为了评奖评优考研出国或是充实学习生活亦或丰富简历，都需要参加比赛考取证书，但平常通过学校了解到的赛事只有少数，若是自己上网查找，靠单一的搜索，不够全面，不够便捷，有些比赛不知道赛名就可能难以找到，筛选有效、含金量高的赛事更是繁琐。除此之外，一些比较重要的比赛可能还会因为时间没有关注好而遗憾错过；自己搜索的比赛种类繁多，对于比赛的选择，一般人不能够有充分的判断，对于自己的需求不清晰，难以做出选择。为消除这些信息差，赛搜搜小程序应运而生。

1.1 愿景

帮助用户（主要大学生）来获得更加精准，全面的竞赛信息。及时了解热门的比赛讯息，得到相关需求的推荐，人性化帮扶，解决大学生竞赛没思路，消息滞后等情况。更能够主动了解到更多比赛，有充分的参考，做出最合理的选择。努力提升小程序的用户体验，使之得到更加广泛地应用。

1.2 理念

如今正是信息飞速流通的时代，信息爆炸性增长。什么样的信息才是我们最需要的？怎样才能把最适合的信息收集起来，整理好，精准提供给用户。放在大学生竞赛方面，通过小程序的使用给大学生们的竞赛信息浏览和收集提供便捷，带来更加良好的阅读体验或者效益是我们的宗旨。

2. 功能介绍

2.1 导航栏

导航栏分为三个栏目：推荐，搜索，我的。分别链接：推荐，搜索和自己三个界面，方便用户的查看操作。

2.2 推荐界面

主界面设置为推荐界面，包含热门和猜你喜欢两大模块。热门专栏专门向用户推荐浏览人数比较多的，受欢迎的比赛项目。猜你喜欢专栏根据用户自行设置的感兴趣的科目类别

2.3 搜索界面

通过搜索查找相关比赛。最上面的是搜索框，用以检索输入的比赛名字，匹配比赛内容。设置有科目，比赛时间，等级三个方面的分类。提供通过科目查找相关比赛，通过比赛月份查找比赛，通过等级查找比赛。

2.4 竞赛信息

点击进入竞赛内部，可以浏览到比赛的日期，科目，等级，图片，简介等等内容，提供主要信息给用户参考。如果最后感兴趣还可以收藏起来到个人界面查看，方便下次浏览。

2.5 个人界面

个人界面，可以查看浏览过的历史，收藏的竞赛，我的兴趣，还有关于赛搜搜的介绍。

3. 技术介绍

3.1 小程序开发

3.1.1 关于小程序的技术优势

- 1. 方便快捷，即用即走。**不需要再下载额外的APP，既费流量，又占空间内存，即用即走。
- 2. 速度快、不占内存。**因为小程序前端代码都是存在微信服务器上的，在腾讯云端存放，所以无需加载，直接就打开了，速度也比较快。并且还不占用手机内存。
- 3. 安全稳定、保密性强。**其实小程序需要审核才能发布，具有一定的保障。其次小程序通信采用的是Https访问，SSL加密通信，并且小程序样式代码都封装在微信小程序里面，所以安全性更高，更稳定。
- 4. 开发成本低、维护简便。**同样的功能，做一个APP成本估值更高，而开发一个小程序，更加方便，维护起来相对更加简单方便。

3.1.2 界面设计

小程序在界面设计中参考了许多相关app和小程序的设计，最终决定了现在以推荐+搜索为主要的的设计，并且在UI中参考了github上较为热门的ColorUI项目，通过该UI组件使得程序设计更美观。

3.1.3 推荐功能实现

推荐页面包括推荐功能以及猜你喜欢两个模块，推荐的赛事是通过对数据库中存有是赛事数据的关注人数进行排序，选取前五名的赛事来进行热门推荐。猜你喜欢则在关注人数的排序基础上，结合了用户的感兴趣的内容进行推荐，用户的兴趣可以自行设置

3.1.4 筛选功能实现

比赛列表筛选默认为全部，用户选择不同的比赛时间或者类别进行筛选查看的时候，前端会返回用户选择的内容给后端，后端检测到数据改变则会调用已经写好的云函数，从云数据库获取相关的比赛内容，前端再进行渲染。

3.1.5 用户浏览记录及关注内容实现

用户点击一个赛事后，首先会对用户个人数据库进行查找，用户是否曾经浏览过该赛事，如果没有则在用户数据库中添加该赛事。关注功能大致同理。

3.1.6 云开发技术应用

小程序的比赛数据全部存储与云数据库中，省去搭建数据服务器的工作，大大提高了开发进度，在获取比赛列表，获取用户信息也使用了云函数，能够进行更高效的数据获取并在前端渲染。

3.2 数据爬取

3.2.1 云开发

1. 无需搭建服务器

无需搭建服务器，只需使用平台提供的各项能力，即可快速开发业务。

2. 免登录、免鉴权调用微信开放服务

无需管理证书、签名、秘钥，直接调用微信 API

复用微信私有协议及链路，保证业务安全性。

3. 统一开发多端应用

支持环境共享，一个后端环境可开发多个小程序、公众号、网页等，便捷复用业务代码与数据。

4. 不限开发语言和框架

开发者可以使用任意语言和框架进行代码开发，构建为容器后，快速将其托管至云开发。

5. 按量计费，成本更低

支持按量计费模式，后端资源根据业务流量自动扩容，先使用后付费，无需支付闲置成本。

6. 技术应用

小程序采用前后端分离的方法进行开发，有较高的开发效率，耦合度低。

所有数据均存放在云开发环境中的数据库，并设置相应的读写权限（对于竞赛信息采用所有用户可读，而用户信息仅创建者可读写），保证了数据的安全性。利用云函数来对数据库进行增删改查的操作，而本地只需要调用云函数即可。

3.2.2 爬虫

1. Python

采用Python作为爬虫的脚本，Python具有许多优秀的数据获取与处理库，同时具有较好的可移植性。因此，后端的爬虫代码有python语言编写。

2. Requests

Requests是用python语言基于urllib编写的，采用的是Apache2 Licensed开源协议的HTTP库。与urllib相比，Requests更加方便，可以节约我们大量的工作。

数据的获取，相关图片的下载均采用requests库。通过request库的下载竞赛信息相关的图片，并加以分类，丰富小程序的内容和浏览体验。

3. Beautiful Soup

Beautiful Soup 是一个HTML/XML的解析器，主要的功能是解析和提取 HTML/XML 数据。

BeautifulSoup支持Python标准库中的HTML解析器,还支持一些第三方的解析器，如lxml，lxml 解析器更加强大，速度更快，在本项目中我们使用lxml。

数据的处理与分类均采用beautiful soup库。通过beautiful soup收集竞赛信息，竞赛的简介，类别，时间等等，而后保存。

4. 多线程

使用线程可以把占据长时间的程序中的任务放到后台去处理。

用户界面可以更加吸引人，比如用户点击了一个按钮去触发某些事件的处理，可以弹出一个进度条来显示处理的进度。程序的运行速度可能加快。在一些等待的任务实现上如用户输入、文件读写和网络收发数据等，线程就比较有用了。在这种情况下我们可以释放一些珍贵的资源如内存占用等等。在python使用threading标准库来实现多线程操作，搭配requests.session()来实现会话保持，以此提高爬虫运行速度，解决最初爬取效率慢的情况。

5. 技术应用

通过模拟浏览器标识，利用竞赛网站的url构建通道，获得比赛项目的主要信息，还有比赛的内部url，进一步跳转爬取，获得竞赛页面的更多的详细信息，最后是保存和整理数据。