清华大学

软件需求规格说明

Version 1.0

修改历史

日期	版本	说明	作者
2023-10-17	1.0	初稿	卢宇涛，耿睿

目　录

1. 简介

[*说明：本节提供对整个 SRS 的综述。*]

1.1 目的

[*说明：明确该 SRS 文档的目的与读者对象。*]

本文档面向“校内公众号信息整合助手”小程序的开发者。通过详细介绍小程序的基本信息、用户画像和需求标准，帮助开发者理解小程序的功能和用途，设计出有针对性和实用性的软件，准确高效地完成开发过程、解决用户需求。

1.2 范围

[*说明：提供所要开发产品的名称和总体功能描述，解释软件产品将完成什么工作，在必要时解释该产品无法完成什么工作，并描述具体的软件应用。*]

本文档所描述的开发产品名为“校内公众号信息整合助手”（简称“小助手”）。

“小助手”旨在为用户提供一个“获取多个公众号的最新文章信息、并生成精炼总结”的一站式平台，以帮助用户在短时间内获取多个公众号的消息，并可以通过用户自定义订阅信息。

对于用户而言，“小助手”提供了“一站式链接”：通过爬虫获取校内大部分公众号的文章链接，并按照时间顺序进行排序，用户可以直接点击链接跳转到公众号文章页面阅读完整文章，这样“按时间排序”而非“按公众号排序”的方式能够便利用户获取最新消息，避免逐个翻阅公众号，造成时间浪费。

对于那些想快速知晓消息、而不希望阅读文章全文的用户而言，“小助手”在爬取文章时会利用大模型对文章进行精炼总结，并提取出关键信息（如活动主体、活动时间、活动地点、注意事项等），使得用户快速知晓内容。

对于那些并没有特别目的、只想看看校内新闻的用户而言，“小助手”会提供推荐算法，根据文章内容的重要程度进行评级，将重要的内容优先呈现给用户。

“小助手”也将支持个性化内容定制。默认情况下，“小助手”只会提供校内主要公众号的文章信息。但用户可以根据自己的喜好进行其它公众号的订阅，提高用户的体验。

“小助手”还拥有着标签-过滤器内容。后台在提取文章时也会由大模型对文章生成标签，用户可以选择特定的标签以过滤掉不感兴趣的内容。

1.3 定义、缩写词以及简写

[*说明：提供正确理解 SRS 所必须的所有术语、缩写词和简写的定义，这些信息也可以在附录的参考文献或其他文档中提供。*]

“小助手”：即本文档所描述的开发产品，全名为“校内公众号信息整合助手”。

SRS：软件需求规格说明书 (Software Requirements Specification)。

API：应用程序编程接口 (Application Programming Interface)。

大模型/LLM：大语言模型 (Large Language Model)，用于文章总结和信息提取的人工智能服务。

爬虫：指自动抓取和解析目标公众号文章内容的程序或脚本。

信息流：指小程序首页按时间倒序展示用户订阅内容的文章列表。

Docker：一种容器化技术，用于将应用程序与其依赖环境打包，实现快速、一致的部署。

JWT：JSON Web Token，一种用于安全传输信息的令牌，可用于身份验证。

1.4 参考文献

[*说明：列举编写 SRS 时所参考的资料或其它资源，可能包括用户界面风格指导、合同、标准、系统需求规格说明、使用实例文档，或相关产品的 SRS。在这里应该给出详细的信息，包括标题名称、作者、版本号、日期、出版单位或资料来源，以方便读者查阅这些文献。*]

1.wewe-rss，https://github.com/cooderl/wewe-rss

2.搜狗微信，https://weixin.sogou.com/

3.微信订阅号。

4.rsshub，https://rsshub-docs.mlj-dragon.cn/

1.5 内容组织

[*说明：综合描述 SRS 的其他部分内容以及它是如何组织的。*]

本文档将按照总分的模式组织内容。第二部分将从总体视角描述产品概况，包括产品的使用前景、产品功能、会产生的效果与平台的限制及依赖关系。第三部分将着重讲述小程序的详细需求信息。附录中将展示用户地图和用户故事。

2. 综合描述

[*说明：本节将描述影响产品及其需求的常规因素，下面的每一部分将使需求更易于理解，但是并不强调具体的需求。*]

2.1 产品前景

[*说明：介绍该产品与其他产品或项目的联系，诸如该产品是否是产品系列中的下一成员，是否是成熟产品所改进的下一代产品、是否是现有应用程序的替代品，或者是否是一个新型的、自含型产品。如果软件需求规格说明定义了大系统的一个组成部分，那么就要说明这部分软件是怎样与整个系统相关联的，并且要定义出两者之间的接口。*]

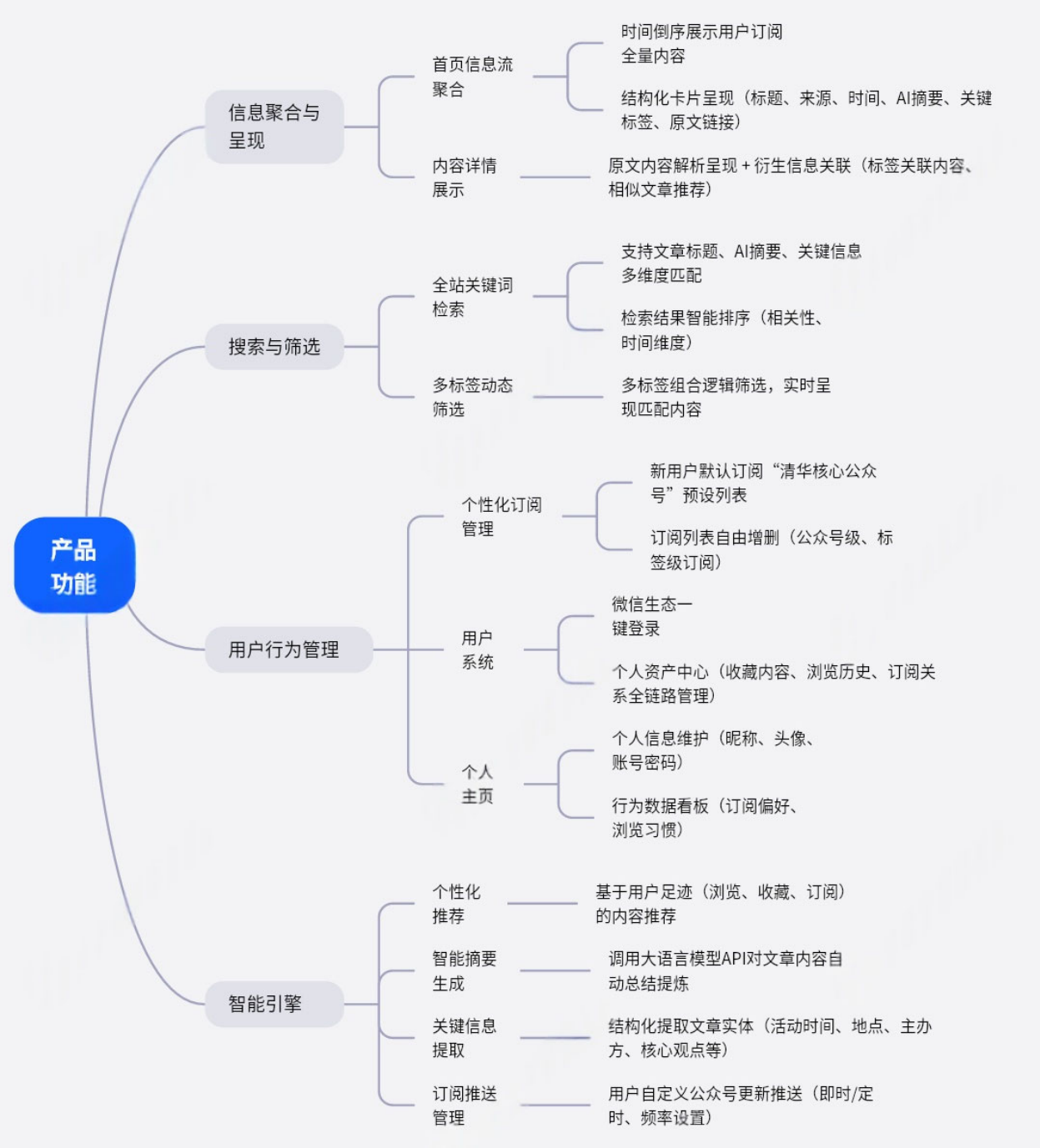
对于当下的信息时代而言，信息的产出量已经远远大于信息的接受量，如何从海量的信息中提取出关键内容成为重要的时代问题；而微信公众号作为日常碎片化阅读的一个主体，正需要

目前已经存在着一些公众号信息整合项目，比如微信官方提供的微信订阅号，第三方的 wewe-rss、搜狗微信、rsshub 等。wewe-rss 作为一个第三方项目，实际运行代码并不开源，可能存在安全性问题；搜狗微信只是作为一个“公众号文章”搜索引擎，而且早已停止内容更新；rsshub 的使用难度大、限制多，还需要科学上网，对于普通用户而言有较高的门槛；微信订阅号则是按“公众号”分类，缺少聚合性。此外，这些项目都存在着一个共同问题：没法提供精炼的内容总结，想要获取文章内容必须阅读所有文章全文，时间开销很大。

“校内公众号信息整合助手”则将提供一体化解决方案，突出“个性+聚合+精炼”的性质。“个性”体现在拥有着类似微信订阅号的订阅功能；“聚合”体现在将校内的主要公众号文章进行提取、发布，用户无需特别订阅也能获取最新消息；“精炼”则是利用大模型辅助信息提取和内容总结、标签分类等，降低了人为管理成本，提高用户获取信息效率。

2.2 产品功能

[*说明：概述该产品所具有的主要功能，这些功能应该按照一种有效的方式进行组织，使功能列表能够被客户或第一次阅读该文档的所有人都易于理解。用图形化的模型表示主要的需求分组以及它们之间的联系是十分有用的，但这并不是产品设计本身的要求，而只是一个有效的解释工具。*]



2.3 用户特征

[*说明：描述可能使用该产品的不同用户类及其相关特征，诸如用户、操作人员以及维护人员等，他们的某些特征（如教育程度、经验以及技术专长等）将对系统的操作环境产生重要的约束。*]

由于“小助手”主要目标是为清华在校生服务，但也可以服务其它需要“公众号一站式管理”的用户，因此将用户分为这四类：

1.清华本科生

这类用户的学习、生活过程中，常常需要从校内公众号获取最新的消息。低年级同学信息需求广泛，对社团、讲座、课程、校园活动等充满好奇，但信息来源碎片化，不知从何入手。高年级同学则面临升学或就业压力，课程、实习等事务繁忙，时间被分割成碎片，没有精力逐一翻阅公众号。他们共同的核心需求是一个能整合所有信息来源的信息门户，希望通过统一浏览和智能筛选，高效地获取有价值的内容。

2.清华研究生

这类用户的信息需求更加专业化，时间也更为宝贵。他们的关注点高度集中在与自身研究领域相关的学术讲座、国际会议、科研动态以及高质量的招聘信息（如博士后、企业研发岗）上。由于科研压力大，他们对于信息获取的效率和精准度要求极高，没有耐心阅读无关或冗余的内容。他们迫切需要一个能能够快速从海量文章中精准抓取所需信息的工具，并通过精炼的总结帮助他们迅速判断价值，从而将宝贵的时间投入到最重要的科研工作中。

3.教职工/行政人员

这类用户在日常工作中，既是信息的接收者，也常常是信息的发布者。他们需要确保由自己负责的重要通知能够被同学们及时、准确地接收到，同时也希望了解其他部门和院系的动态，以便从全局视角掌握校园态势，促进跨部门的工作协同。因此，他们希望既能方便地监督信息的传播效果，又能轻松地一览全局。

4.校友及其他校外用户

这类用户虽然已离开校园，但对母校保持着持续的关注与深厚的情感联结。他们希望了解学校的重大发展、科研成果和校友活动，但受限于工作和生活，无法像在校时那样高频次、全方位地获取信息。他们的关注更具选择性，通常只聚焦于自己感兴趣的特定领域。因此，他们希望能够以最低的时间成本，定期接收到由平台筛选和推送的母校精选要闻，以一种轻松、便捷的方式维系与母校的纽带。

2.4 一般性限制

[*说明：描述将限制开发人员进行设计选择的一些项目，可能的限制包括如下内容：*

- *必须使用或者避免的特定技术、工具、编程语言和数据库*
- *所要求的开发规范或标准*
- *企业策略、政府法规或工业标准*
- *硬件限制，例如定时需求或存储器限制*
- *数据转换格式标准*
- *软件运行环境等*]

1 技术工具

团队使用 Git 管理代码开发，将代码托管在 Github 仓库中实现动态管理。

使用 Docker 将代码从开发环境移植到部署环境。

前端采用 Vue.js 框架，使用 HbuilderX 和微信开发者工具。

后端使用 Python 作为开发语言。

使用 MySQL 作为主数据库。

2 开发规范

代码遵循团队约定的编码规范，包括命名、注释、目录结构。

前后端分离开发，接口文档使用 Swagger 进行维护。

单元测试覆盖率不低于 70%。

接口设计遵循 RESTful 风格。

3 企业策略

所有数据采集行为需遵守《微信公众平台运营规范》及相关法律法规。

用户数据存储与处理需符合《个人信息保护法》要求。

不得采集、存储或传播敏感信息或违法违规内容。

4 软件运行环境

开发者应确保服务器的安全性设置，包括防火墙、访问控制列表（ACL）和其他相关的网络和系统安全配置，以保护网页和服务器免受潜在的安全威胁。为了监控和管理网页的性能和运行情况，应当设置适当的监控工具和日志记录系统，以便及时检测和处理发生的问题。

后端服务部署于 Linux 服务器，支持容器化部署。支持在主流浏览器（Chrome、Safari、Edge）及微信小程序环境中运行。支持并发用户数不少于 100，文章爬取与处理任务在设定时间窗口内完成。

2.5 假设和依赖

［*说明：列举出在对 SRS 中影响需求陈述的假设因素，以及项目对外部因素存在的依赖。*］

假设：

我们假设用户主要为清华在校师生及校友群体，具备基本的移动互联网使用能力，能够熟练操作微信小程序。且用户对公众号内容聚合、信息筛选和 AI 摘要功能有实际需求，并认可通过技术手段提升信息获取效率的方式。

在公众号层面，我们假设目标公众号的内容结构在可预见的周期内保持相对稳定，使得爬虫系统能够持续有效地提取文章信息。

依赖：

本平台的公众号文章内容完全依赖于公开的微信公众号源，其可访问性与内容合法性是系统运行的基础。内容爬取与整合的效果直接依赖于这些公众号的更新频率、内容质量及排版规范性。核心的 AI 摘要与标签生成功能依赖于第三方大语言模型（LLM）API 服务的可用性、性能及调用成本。该功能的响应速度与总结质量不受本平台直接控制，而是由模型服务提供商决定。

用户系统的建立与运行依赖于微信开放平台提供的登录授权接口。推送服务等功能则依赖于微信小程序平台提供的消息能力与政策限制。

此外，平台的响应速度与稳定性部分依赖于用户自身的网络质量、服务器资源的充足性，以及所集成的各类第三方服务的工作状态。平台的内容推荐效果依赖于用户使用过程中产生的行为数据积累。

3. 详细需求

[*说明：本节是 SRS 的最重要部分，它包含开发人员用来创建一个设计方案所需的全部细节信息。*]

3.1 功能需求

[*说明：列出该产品的详细功能需求，指出每一个功能的输入、处理操作和输出。这些是必须提交给用户的软件功能，使用户可以使用所提供的特性执行服务或者使用所指定的用例执行任务。*]

1. 用户管理

1.1. 微信一键登录：

用户通过微信授权直接登录系统，系统自动获取用户的微信 OpenID、昵称和头像，并在后端创建或关联账户，实现无缝注册与登录流程。

1.2. 用户信息维护：

用户可在个人中心编辑个人信息，包括修改昵称、上传个性化头像、设置内容偏好。

1.3. 浏览历史记录：

系统自动记录用户浏览过的文章，并按照浏览时间倒序排列，用户可随时查看和清空浏览历史。

1.4.文章收藏与取消收藏：

用户可将感兴趣的文章加入收藏夹，并可在“我的收藏”页面中管理（查看、取消收藏）所有已收藏的文章。

2. 内容聚合与处理

2.1. 公众号爬虫：

系统定时轮询预设的公众号列表及用户自定义订阅的公众号，通过合法技术手段爬取其最新发布的文章元数据（标题、链接、发布时间）及正文内容。

2.2. 数据清洗与去重：

对爬取到的原始数据，系统通过计算文章标题和正文内容的特征码（哈希值）进行智能比对，有效识别并过滤转载、重复发布的内容，确保信息流的唯一性。

2.3. AI 摘要生成：

系统将清洗后的文章正文发送至大语言模型（LLM）API，生成一段凝练、准确、客观的文本摘要，概括文章核心内容。

2.4. 关键信息结构化提取：

系统利用大语言模型（LLM）的命名实体识别（NER）能力，从文章中精准提取关键信息（如活动时间、地点、主办方、报名方式、截止日期、联系人等），并转化为结构化的数据，便于存储和前端调用。

2.5. 内容标签自动分类：

基于文章摘要和全文内容，系统自动为每篇文章打上多个分类标签（如“学术讲座”、“社团招新”、“志愿活动”、“后勤通知”、“就业招聘”等），支持多级标签体系。

2.6. 内容质量与优先级评分：

系统根据公众号权威度、文章互动量、内容关键词（如“紧急”、“重要”）等因素，对文章进行综合评分，为个性化推荐和排序提供依据。

3. 内容展示与交互

3.1. 统一信息流展示：

首页信息流默认聚合用户所有订阅公众号的文章，并按发布时间选择其中前二十篇降序排列。信息流支持无限滚动模式，提供流畅的浏览体验。

3.2. 内容卡片渲染：

每条内容以卡片形式呈现，清晰展示：文章标题、来源公众号名称、发布时间、AI 生成的摘要、关键信息标签（可点击）、以及“阅读原文”跳转按钮。

3.3. 全局搜索：

提供顶部全局搜索框，支持对公众号、文章标题及标签进行检索。

3.4. 文章详情页：

点击文章卡片后可进入详情页，页面内更完整地展示摘要、所有关键信息标签，并提供明显的原文跳转按钮。

3.5. 内容分享：

支持将文章卡片或详情页分享给微信好友或微信群，分享内容包含小程序路径，可直接点击进入。

4. 订阅与推送管理

4.1. 默认订阅：

新用户注册后，系统自动为其订阅一个由运营人员预设的、覆盖校园学习生活主要方面的“核心公众号列表”。

4.2. 订阅管理后台：

用户可在“我的订阅”页面中，浏览“推荐订阅”列表，查看并管理“已订阅”的公众号，支持取消订阅或者通过搜索公众号名称或 ID，发现并添加新的订阅。

4.3. 智能推荐订阅：

系统根据用户的浏览历史、收藏行为及标签偏好，在“推荐订阅”列表中个性化地推荐其可能感兴趣的公众号文章。

4.4. 个性化推送设置：

用户可在设置中管理推送通知：开关全局推送、选择接收推送的公众号 。

4.5. 推送任务执行：

系统根据用户的推送设置，可以通过微信小程序服务通知模板，向用户发送推送消息。

5. 系统管理

5.1. 管理员后台：

为运营人员提供完整的后台管理系统，支持以下功能：

公众号源管理：对公众号进行增、删、改、查，并监控其爬取状态。

内容管理：查看所有文章列表，对违规、失效或低质内容进行手动隐藏、删除或批量操作。

用户管理：查看用户列表，管理用户状态。

反馈管理：查看和处理用户提交的内容反馈。

5.2. 数据统计：

为管理员提供数据可视化面板，展示关键指标，如：每日活跃用户（DAU）、文章爬取总量、热门标签等。

5.3. 系统日志与监控：

记录系统的关键操作日志（如爬虫运行状态、API 调用情况），并建立监控告警机制，在服务异常时及时通知运维人员。

3.2 外部接口需求

[*说明：描述可以保证该产品与外部组件正确连接的需求，包括用户界面、硬件接口、软件接口和通信接口等。*]

1 用户界面接口：

“小助手”将提供基于微信小程序的用户界面，用户可在微信内直接访问和使用。

用户界面应遵循微信小程序设计指南，确保在主流微信版本上能够正常显示和交互。

核心交互组件包括信息流卡片、搜索框及个人中心页面，所有操作应流畅且符合用户直觉。

2 硬件接口：

“小助手”作为微信小程序，其运行依赖于用户设备的硬件性能，无需与硬件组件直接交互。

3 软件接口：

前端使用微信小程序原生开发框架，技术栈包括 WXML、WXSS 和 JavaScript。

后端服务器使用 Python 编程语言，基于 Django 的 Web 框架开发。

数据存储采用 MySQL 数据库。

与微信开放平台接口集成，调用 API 实现用户一键登录与身份识别。

与大模型提供商的 API 接口进行通信，通过 HTTP 协议调用其文本总结与信息提取功能。

数据采集部分通过 Python 爬虫框架与微信公众号及网页进行交互，以获取文章列表及内容。

集成推荐算法模块，通过读取用户行为数据和文章特征数据，输出个性化推荐列表。

4 通信接口：

小程序前端与后端服务器之间使用 HTTPS 协议进行通信，以确保数据传输的安全性。

前后端数据交互格式统一采用 JSON，以保证数据的结构化与可解析性。

后端与微信开放平台接口、大模型 API、目标公众号服务器之间的通信，均需遵循各服务提供商定义的 API 规范、认证方法和调用频率限制。

3.3 性能需求

［*说明：阐述不同的应用领域对产品性能的需求，并解释其原理以帮助开发人员选择合理的设计，诸如确定所支持的客户端数、并发用户数、文件或记录规模、操作响应时间等。*］

1 支持的客户端数

需求：系统应能稳定支持大量微信小程序客户端的连接与访问，确保在校内信息发布高峰时段（如晚间通知、招新季、期末考试周）所有用户均可正常使用。

原理：通过采用无状态的后端服务设计和负载均衡技术，将用户请求分发到多个服务器实例进行处理。

2 并发用户数

需求:系统需支持至少 100 名用户同时进行核心操作(如浏览信息流、搜索文章、管理订阅)，以应对早晚高峰时段集中的用户访问压力。

原理:高并发支持通过异步编程来实现。核心读操作优先访问缓存，以减轻数据库压力，从而支撑更高的并发用户数。

3 文件或记录规模

需求:系统需能够高效处理并存储持续增长的公众号文章数据，预期在项目初期需支持至少 1000 篇文章记录、50 名用户数据及其行为记录的存储与快速检索。

原理:采用关系型数据库并进行合理的分表索引优化，以支持大规模文章数据的管理。

4 操作响应时间

需求：用户打开小程序首页，信息流加载时间应在 2 秒内完成。用户进行关键词搜索或标签筛选，结果返回时间应在 1.5 秒内完成。用户点击文章链接，跳转至原文页面的延迟应低于 1 秒。

原理:前端通过小程序分包加载和本地缓存优化首次加载速度。后端通过优化数据库查询语句、为常用查询字段建立索引来存储热点文章数据，从而显著降低操作响应时间。AI 文章总结为异步任务，其生成时间不计算在页面响应时间内。

5 数据采集与处理性能

需求:后台数据采集系统应能高效运行，每日定时抓取任务需在 1 小时内完成对预设公众号列表的全量扫描。

原理:爬虫系统采用异步 IO 和非阻塞式并发请求，以提高抓取效率。AI 处理模块通过任务队列将总结任务异步化，并可通过增加 Worker 数量来水平扩展处理能力，从而应对批量文章的处理需求。

3.4 质量属性

[*说明：详尽陈述与客户或开发人员至关重要的产品质量特性，这些特性必须是确定、定量的并在可能时是可验证的。有关质量属性的定义如下：*

- *可用性：系统可以使用并且完全操作的时间*
- *可扩展性：软件中增加新功能的所需时间*
- *安全性：控制软件被未经授权者访问的范围*
- *可靠性：程序的精度范围、系统无故障执行时间概率、故障恢复要求等*
- *互操作性：该系统与其他系统交换数据和服务的要求*
- *可维护性：在操作过程中查找和修复一个错误所需的工作量*
- *可移植性：从一个硬件或软件环境转移到另外一个硬件或软件环境中所需的工作量*
- *可重用性：程序能够在另外一个应用环境中重复使用的范围*
- *可测试性：测试组件或系统以查找缺陷的简单程度*
- *易用性：用户学习、操作、为程序准备输入以及解释程序的输出所需的工作量。*]

可用性：

“小助手”应具备高可用性，确保在校学生学习与生活的高频使用时段系统服务可以正常运行。

计划内的系统维护应尽量安排在凌晨低峰时段进行。

可扩展性：

系统应采用微服务架构，以保证良好的可扩展性。当需要增加新功能时，平均开发与集成周期不应超过 2 周。

安全性：

系统应采取严格的安全措施。所有用户需通过微信官方认证登录，确保身份真实。用户数据与订阅关系在传输和存储时均需加密。

爬虫组件需遵守 Robots 协议并设置合理的访问频率，避免对公众号服务器造成攻击性负载，同时防止 IP 被封禁。

可靠性：

系统应保证高可靠性。核心服务的无故障运行能力应该足够高。对于文章 AI 总结功能，应确保其内容准确率。

互操作性：

“小助手”应具备良好的互操作性，能够与微信生态系统、大模型提供商以及未来可能接入的校内其他信息平台进行顺畅的数据与服务交换。

可维护性：

系统的代码结构应清晰，并配有详细的注释和技术文档。确保在运行过程中，开发人员能够快速定位并修复错误。

可移植性：

通过采用 Docker 容器化技术，确保后端服务可以快速地在不同的服务器环境中进行部署和迁移性。

可重用性：

系统中的核心组件应设计为可重用的模块。数据爬取引擎、AI 信息摘要模块，应能在其他需要类似功能的内容聚合或信息处理项目中直接复用，以减少重复开发成本。

可测试性：

平台应易于测试，以确保在部署之前能够识别和纠正问题。测试组件或系统的简易性将有助于有效的质量保证。

易用性：

“小助手”的用户界面应直观易用，确保新用户无需阅读帮助文档即可快速完成核心功能的上手操作。所有功能点的操作步骤应尽可能简洁。

3.5 其他需求

[*说明：定义在软件需求规格说明的其它部分未出现的需求，例如国际化需求、数据库需求等，还可以增加有关操作、管理和维护部分来完善产品安装、配置、启动和关闭、修复和容错，以及登录和监控操作等方面的需求。*]

国际化需求：

“小助手”初期主要面向清华大学在校师生及校友，用户界面与内容处理默认支持简体中文。未来若用户群体扩展至国际学生或对多语言有显著需求时，系统架构应支持界面语言的国际化与本地化，用户应能够便捷地切换界面语言。

数据库需求：

数据库应提供高性能的数据读写能力，针对文章列表查询、用户订阅关系等高频操作建立有效的索引策略。

必须建立定期的数据库备份机制，以防止单点故障导致的数据丢失。

数据库结构设计应清晰规范，便于进行数据的增删改查。

数据库访问必须实行严格的权限控制，应用程序使用专属账户连接，禁止默认或弱密码，并通过网络策略限制非授权 IP 的访问，确保数据安全。

配置需求：

系统应提供一个统一的后台配置文件，允许管理员灵活配置关键参数，包括但不限于：公众号爬取名单与频率、AI 模型提供商与 API 密钥、内容缓存过期时间、以及向用户推送消息的模板等。

日志和监控需求：

平台应生成详细的操作日志，以记录用户的活动和系统事件，以便于平台管理员维护平台。

监控系统应监测系统性能、资源使用和错误情况，并向管理员提供警报和通知。

容错和修复需求：

系统应具备良好的容错机制。当非核心服务出现故障时，不应导致整个系统崩溃。系统应能够自动从错误中恢复，并减少用户的中断时间。

4. 支持信息

[*说明：本节包含所有必要的术语表、引用文档列表、待确定问题的列表等支持信息。*]

必要术语表：

“小助手”：本文档所描述的开发产品，全名为“校内公众号信息整合助手”。

SRS：软件需求规格说明书 (Software Requirements Specification)。

API：应用程序编程接口 (Application Programming Interface)。

大模型/LLM：大语言模型 (Large Language Model)，用于文章总结和信息提取的人工智能服务。

爬虫：指自动抓取和解析目标公众号文章内容的程序或脚本。

信息流：指小程序首页按时间倒序展示用户订阅内容的文章列表。

Docker：一种容器化技术，用于将应用程序与其依赖环境打包，实现快速、一致的部署。

JWT：JSON Web Token，一种用于安全传输信息的令牌，可用于身份验证。

待确定的问题列表：

1.大模型供应商的最终选择与成本评估。

2.数据采集的合法性与合规性边界。

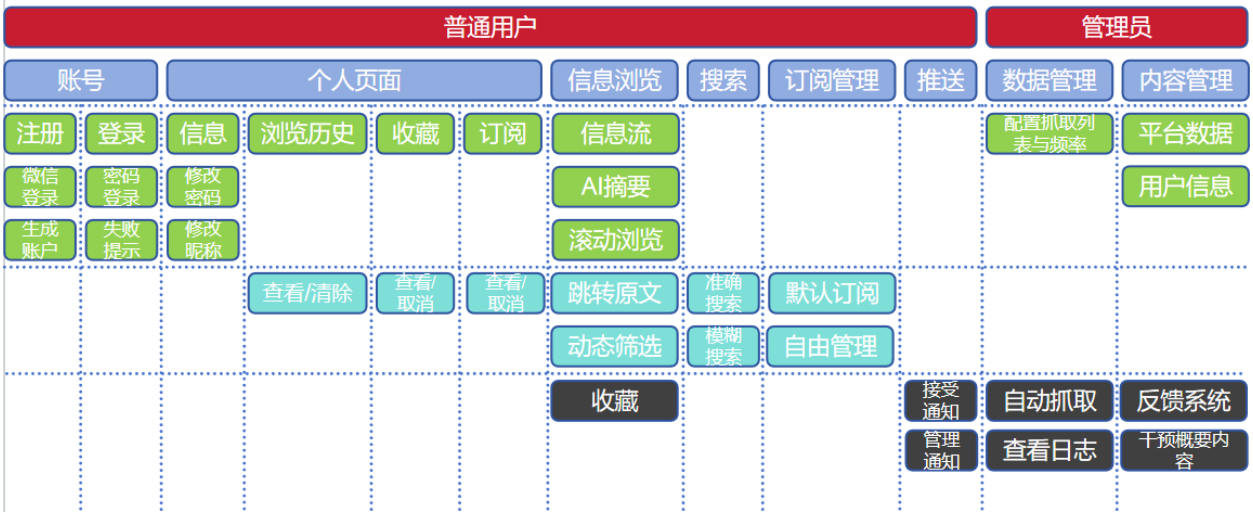
3.消息推送的频率与用户打扰平衡，需要确定向用户推送公众号更新消息的默认频率。

4.内容安全与审核机制，防止传播不当或错误信息。

5. 附录

5.1 用户故事

用户故事地图：



普通使用者：

登录部分：

1. 作为新用户，我希望能够使用微信一键授权登录，以便快速注册和进入小程序，无需记忆额外的账号密码。

2. 作为已登录用户，我希望登录状态能够在一定期限内有效，以免频繁重复登录。

3. 作为用户，如果登录失败，我希望看到清晰的错误提示，以便知道如何解决问题。

个人页面部分：

1. 作为普通用户，我可以在“我的”页面查看和编辑我的昵称和头像，以便展示我的个人形象。

2. 作为普通用户，我可以查看我的浏览历史列表（按时间倒序），并能够清空历史，以便快速找回我之前感兴趣的内容。

3. 作为普通用户，我可以收藏感兴趣的文章，并在“我的收藏”页面查看、取消收藏，以便标记和回顾重要内容。

4. 作为普通用户，我可以管理我订阅的公众号列表（添加或取消订阅），以便定制我专属的信息流内容。

信息浏览部分：

1. 作为普通用户，我可以在首页看到一个按发布时间倒序排列的聚合信息流，内容来自我订阅的所有公众号，以便一站式获取最新消息。

2. 作为普通用户，我可以在信息流中看到每篇文章的 AI 生成的精炼摘要和关键标签，以便快速判断文章内容是否与我相关。

3. 作为普通用户，我可以点击信息流中的文章卡片，直接跳转到微信公众号原文进行阅读，以便查看完整内容。

4. 作为普通用户，我可以对信息流中的文章进行收藏，以便标记喜好。

5. 作为普通用户，我可以通过点选一个或多个标签，动态筛选出同时匹配所有标签的文章，以便精准过滤出我感兴趣的内容类型。

6. 作为普通用户，我在浏览信息流时可以通过无限滚动加载更多内容，以便无缝浏览历史文章。

搜索部分：

1. 作为普通用户，我可以通过顶部搜索框搜索文章标题、公众号名称或标签，以便快速找到特定内容。

2. 作为普通用户，我在搜索公众号时能看到相关搜索建议，以便提高搜索效率。

订阅管理部分：

1.作为新用户，我在首次登录时系统能为我默认订阅一批“清华核心公众号”，以便我无需手动操作即可开始使用。

2.作为普通用户，我可以在“订阅管理”页面自由搜索、添加新的公众号或取消订阅现有公众号，以便根据我的兴趣变化个性化我的信息源。

推送部分：

1.作为普通用户，我可以在设置中管理推送通知：开关全局推送、选择接收推送的公众号，以便及时获取重要更新而不被过多打扰。

2.作为普通用户，我可以收到微信服务通知，当有新的文章发布时，以便及时阅读。

后台管理者：

数据采集与处理部分：

1. 作为后台管理者，我可以配置需要抓取的公众号列表及其抓取频率，以确保信息源的全面性和时效性。

2. 作为后台管理者，系统能够自动、定时地运行爬虫任务，抓取新文章并存入数据库，以减少人工干预。

3. 作为后台管理者，我可以查看爬虫任务的成功/失败日志和 AI 处理的结果，以便监控系统运行状态并及时处理异常。

内容与用户管理部分：

1. 作为后台管理者，我可以在后台管理界面中手动干预文章的标签或摘要，以修正 AI 可能产生的错误。

2. 作为后台管理者，我可以查看全平台用户的基本统计信息，以便了解产品运营状况。

3.作为后台管理者，我可以查看和处理用户提交的内容反馈，以便改进系统。

4.作为后台管理者，我可以查看平台数据面板，包括每日活跃用户、文章爬取总量、热门标签等关键指标，以便了解产品运营状况。

5.2 小组分工

原型设计：于川皓，耿睿

前端：于川皓，耿睿

后端：卢宇涛，王雪松

组织组内会议，统筹进度：卢宇涛

5.3 开发迭代计划

轮次	周次	任务
第一轮	**第 7-8 周**	实现首页信息流静态展示，后端 docker+django+MySQL 环境部署，实现前后端的通信。
第二轮	**第 9 周**	完善爬虫系统，实现定时自动抓取预设的核心公众号列表。集成大模型 API，实现文章自动摘要生成、关键信息提取和标签分类功能。前端动态渲染真实

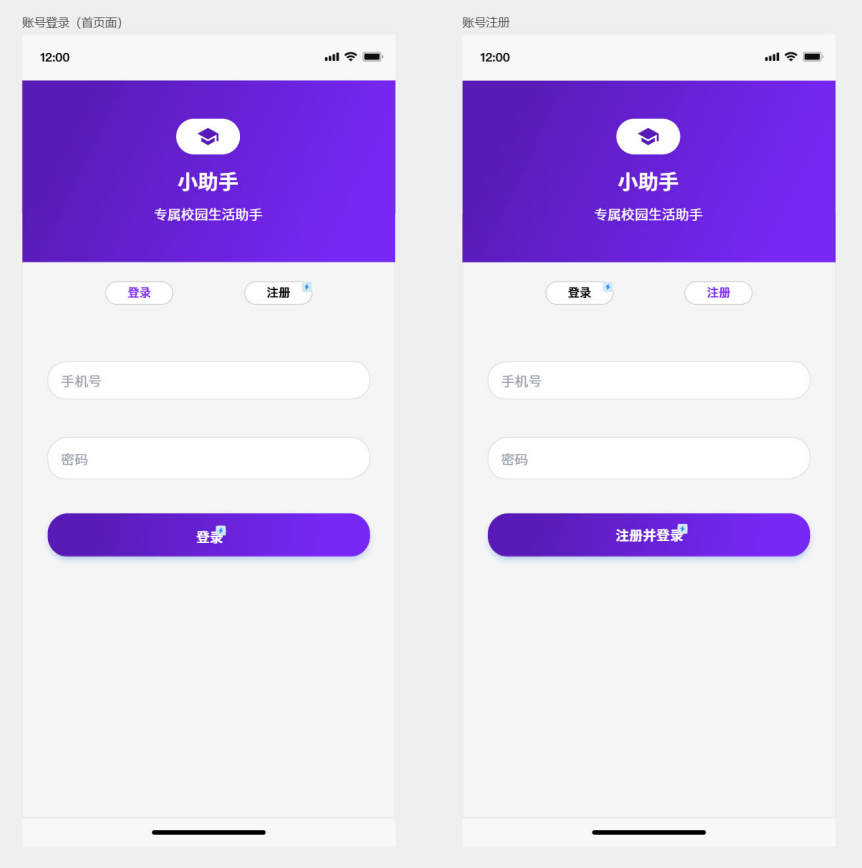
		文章信息流。发布版本 1。
第三轮	**第 10-11 周**	实现全局搜索功能（按标题、标签、公众号搜索）。开发“我的订阅”页面，允许用户自由添加和取消订阅公众号。实现用户的文章收藏与浏览历史功能。
第四轮	**第 12 周**	为新用户设置默认订阅列表。实现简单的推荐算法，在首页或订阅页面为用户推荐可能感兴趣的文章。发布版本 2。
第五轮	**第 13 周**	实现用户个性化推送设置（开关、按公众号筛选）。集成微信服务通知，实现文章更新推送功能。
第六轮	**第 14 周**	优化后台管理系统，增加内容管理和用户管理面板。实现用户反馈功能。进行系统性能优化，提升前端加载速度。发布版本 3。
第七轮	**第 15 周**	进行平台的测试，编写使用教程，发布最终版本。

5.4 产品原型

使用墨刀实现产品原型设计，点击以下链接预览：

https://modao.cc/proto/hLOJ4jgLt4fjpeJMI1yrpt/sharing?view_mode=read_only&screen=rbpV0AX4BEygiGXbt #小助手原型设计-分享

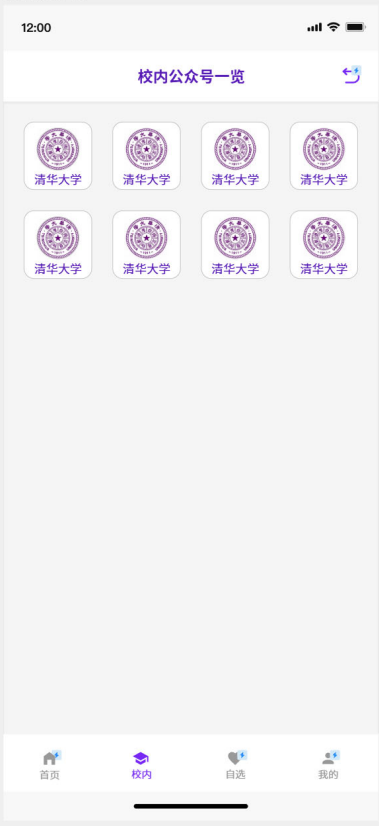
主要页面截图：


账号登录（首页面）
12:00
小助手
专属校园生活助手
登录
注册
手机号
密码
登录
账号注册
12:00
小助手
专属校园生活助手
登录
注册
手机号
密码
注册并登录

首页

校内

校内公众号一览



自选

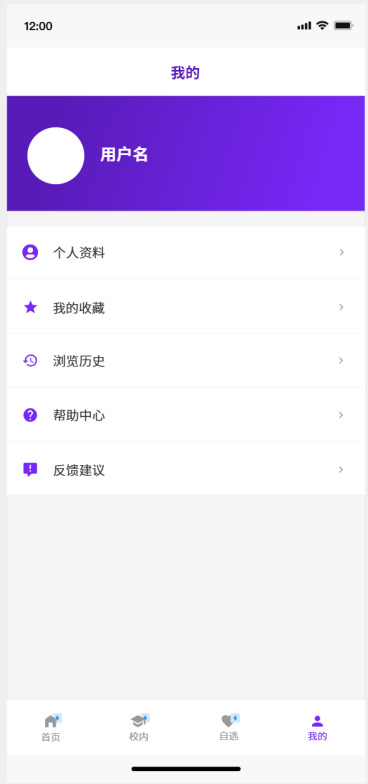
添加自选


12:00
添加自选
添加公众号
软小宣
添加

软小宣


12:00
软小宣
奖学金申请开放
12月25日截止，速速申请
#tag1 #tag2
教务处 12-16 09:30
xx食堂上新啦!
好吃好吃非常好吃
#tag1 #tag2
教务处 12-16 09:30
奖学金申请开放
12月25日截止，速速申请
#tag1 #tag2
教务处 12-16 09:30
奖学金申请开放
12月25日截止，速速申请
#tag1 #tag2
教务处 12-16 09:30
奖学金申请开放
12月25日截止，速速申请
#tag1 #tag2
教务处 12-16 09:30
奖学金申请开放
12月25日截止，速速申请
#tag1 #tag2
教务处 12-16 09:30
奖学金申请开放
12月25日截止，速速申请
#tag1 #tag2
教务处 12-16 09:30

我的


12:00
我的
用户名
个人资料
我的收藏
浏览历史
帮助中心
反馈建议
首页
校内
自选
我的