

广工枢纽：需求规格说明书

1. 引言

1.1 目的

本系统“广工枢纽”旨在为广东工业大学师生提供一个一站式校园信息服务平台。系统通过自动化爬取校园通知、水电服务、后勤公告、教务信息等数据，构建结构化知识库，并结合大语言模型提供智能问答与信息检索功能，实现以下目标：

- 集中信息**：将分散在多个网站的校园公告和服务信息统一呈现；
- 智能查询**：用户可通过自然语言提问，由大语言模型结合知识库生成准确、易理解的回答；
- 高效更新**：爬虫定期采集信息，保证数据及时、可靠；
- 增强用户体验**：通过清晰界面与交互设计，提高信息获取的便利性和视觉体验。

1.2 项目背景

目前，广东工业大学师生在获取校园信息时存在以下痛点：

- 信息分散**：通知、公告、服务信息分布在不同网站和平台；
- 访问繁琐**：需要频繁切换页面，增加查找成本；
- 更新不及时**：部分信息更新延迟或难以追踪。

广工枢纽通过技术手段集中信息，结合知识库与智能问答，为师生提供便捷、高效的校园信息服务平台，实现“信息集中化、可查询、易用化”。

1.3 系统功能概述

1.3.1 登录界面

- 用户输入学号/工号及密码进行身份验证。
- 登录后可进入首页。

1.3.2 首页

- 信息栏目**：跳转到相关关键词条目，如校园通知、水电服务、教务公告、后勤通知等。
- 自然语言问答**：用户输入问题，系统结合知识库与大模型生成回答。

1.3.3 通知与公告界面

- 按类型分类**：校园通知、水电服务、教务公告、后勤通知等。
- 详情查看**：点击通知标题查看完整内容。
- 关键词搜索**：可按关键词、时间区间筛选通知。
- 通知更新**：结合爬虫实时更新水电服务公告。
- 考试安排**：提供各类考试、比赛时间、地点及报名信息。
- 公告与提醒**：教务通知、选课提醒等集中显示。

1.4 系统特点与核心竞争力

- 1. **信息集中化**：整合校园各类信息源，减少用户访问成本。
- 2. **智能化查询**：结合大语言模型与知识库，实现自然语言问答。
- 3. **实时性与可靠性**：爬虫自动采集并清洗数据，保证信息及时更新。
- 4. **用户体验优先**：界面简洁、交互直观。
- 5. **可扩展性强**：模块化架构，易于增加新的信息源或功能。

2. 系统价值分析

2.1 真实性

- **数据来源可靠**：所有信息均来自校内公开网站，保证内容的准确性与合法性。
 - 网站参考

官网	https://www.gdut.edu.cn/
通知公告网	https://oas.gdut.edu.cn/seeyon/main.do?method=main&loginPortal=1&portalId=-7779029842361826066
图书馆	https://library.gdut.edu.cn/

- **场景贴近实际需求**：系统功能完全基于师生日常信息查询场景设计，真实解决校园信息分散的问题。

2.2 可用性

- **易用界面**：采用 Vue.js 构建简洁、响应式前端页面，支持**询问**、**关键词查看**等交互方式，用户无需培训即可使用。
- **高效信息获取**：通过关键词搜索和智能问答功能，用户可在几秒内获取所需信息，减少浏览多个网站的时间。
- **轻量化部署**：系统采用 JSON 知识库 + 本地前端，无需复杂数据库和服务器即可运行，便于快速部署和维护。

2.3 价值所在

- **节省时间成本**：帮助师生快速获取信息，提高校园生活与教学效率。
- **信息集中与智能化**：将分散信息整合为可检索知识库，并通过大模型 API 提供智能问答，提升信息利用效率。
- **可扩展性强**：可扩展至更多校园信息源或其他高校，具备推广价值。
- **技术实践价值**：涵盖爬虫、数据清洗、前端开发、AI 接口调用等多项技术。
- **情怀**：从学生视角出发解决校园生活痛点，让校园信息服务更智能、更贴心。

3. 面向用户分析

2.1 用户群体

在校学生、教师，查询课程通知、活动安排、宿舍水电信息、后勤服务、最新教学安排、校内公告及通知。

2.2 用户需求

- **快速获取信息**：无需访问多个网站，一站式查询。

- 智能问答：自然语言输入问题，系统返回简洁、准确的回答。
- 信息可视化：界面清晰、交互友好，支持搜索、列表、时间线查看等功能。
- 实时更新：保证数据及时抓取并同步到知识库。

2.3 预期用户数量

- 首期测试用户：约 25 人
 - 包括身边的熟悉的在校学生和教师，用于内部试用、反馈收集和系统优化。
- 正式部署后潜在用户：约 20,000 人
 - 覆盖广东工业大学全校在校学生、教职工及相关校内部门人员。

4. 需求分析

(1) 功能性需求

功能模块	功能描述	输入	输出
数据采集模块	自动爬取校园通知、水电信息、公告等	网站URL、爬取规则	JSON结构化数据
数据清洗与知识库构建	对爬取信息进行去重、分类、格式化	原始爬取数据	可检索的知识库
智能问答模块	根据用户问题从知识库中检索信息并生成回答	用户自然语言问题	精炼回答文本
前端展示模块	提供网页界面展示信息和问答交互	用户操作	页面显示信息列表、搜索结果、问答结果
搜索与关键词匹配	支持关键词搜索和内容过滤	用户关键词	匹配条目列表
日志与监控	记录数据采集、处理、问答操作日志	系统事件	日志文件

(2) 技术需求

5.1 前端

- 技术栈：HTML5、CSS3、JavaScript、Vue.js
- 功能：响应式界面设计、用户交互、搜索与问答显示
- 输出：可访问的网页应用

5.2 后端 / 数据处理

- 技术栈：Python，及Requests、BeautifulSoup、Pandas、JSON等相关库
- 功能：数据爬取、清洗、存储、知识库维护
- 输出：结构化知识库文件

5.3 AI 接口

- 技术栈：OpenAI API 或通义千问 API
- 功能：接收用户问题，将知识库内容作为 prompt 输入，返回生成文本
- 输出：智能问答结果

5.4 系统环境

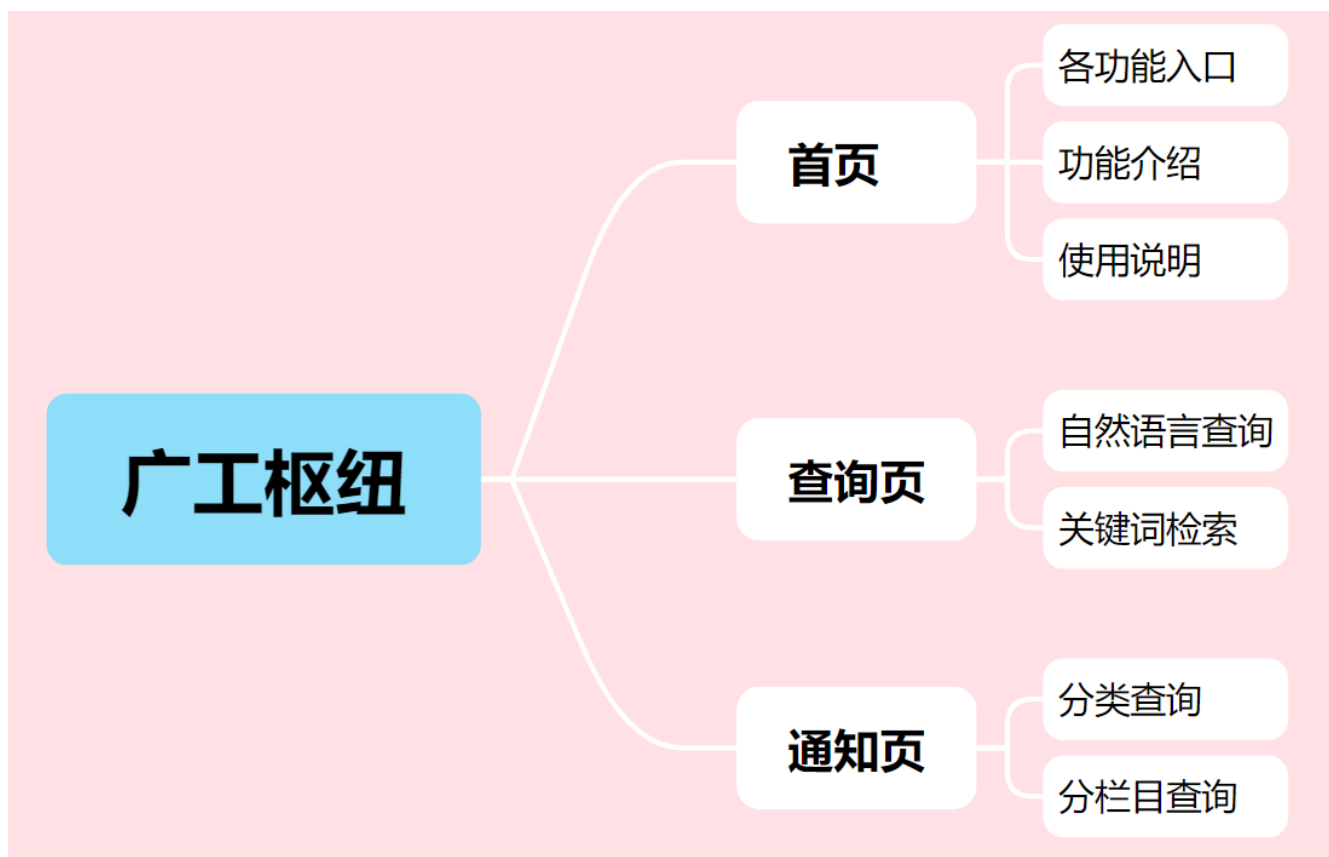
- 浏览器兼容 Chrome、Edge、Firefox

5.5 其他

- **性能**：前端页面响应时间 < 2 秒，问答流式输出
- **可维护性**：模块化设计，方便后续扩展或功能调整
- **可靠性**：爬虫与数据处理具备错误处理机制，防止数据缺失或重复
- **可扩展性**：可增加更多信息来源或接入其他高校系统

5. 系统框架图及初步界面

(1) 框架图



(2) 系统原型图

- 登录页 (Login Page)



• 首页

GDUT Hub

问答模式

表单模式

通知公告

水电服务

后勤报修

教务信息

关于我们

个人设置

退出登录

吴小吴

晚上好！吴小吴^_^

请输入内容... (eg. 2025寒假放假安排)

GDUT Hub 广工枢纽

一个面向校园生活的智能信息聚合平台，旨在帮助师生快速获取校内的各类通知公告、水电服务、后勤报修、教务信息等内容。我们希望让繁杂分散的校园信息实现自动化聚合与智能化服务，使师生无需频繁浏览多个网站即可快速获取所需内容。

通知公告

查询考试、放假安排通知等

水电服务

查询水电方面通知

后勤报修

查询报修通知

教务信息

查询考试安排、教学活动等

• 查询页

GDUT Hub

问答模式

表单模式

通知公告

水电服务

后勤报修

教务信息

关于我们

个人设置

退出登录

● 吴小吴

新对话1

返回

2025/10/4 00:16:33

明月几时有，把酒问青天？

2025/10/4 00:16:33

不好意思，没有检索到相关关键词。请重试。

2025/10/4 00:20:43

2024 级新生报到注册的通知

2025/10/4 00:20:55

各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张，逾期未报到将按规定处理。咨询电话：XXX-XXXXXXX。

• 通知页

GDUT Hub

问答模式

表单模式

通知公告

水电服务

后勤报修

教务信息

关于我们

个人设置

退出登录

● 吴小吴

自 通知公告

校园通知 学术活动 行政通知 社区公告

<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院
<div><div>2024 级新生报到注册通知</div><div>各位 2024 级新生及家长：新生报到时间为 9 月 1 日 - 2 日（8:00-18:00），报到地点为学校体育馆。需携带身份证、录取通知书、档案材料及近期 1 寸免冠照片 3 张...</div></div>	2025-10-04 计算机学院

上页123下页

团队计划

1. 修正后计划

周次	时间 段	主要 目标	具体任务内容	相关成员	预计 工时	预期成果
第11周 (深化设计 与计划 阶段)	第 1- 2 天	用户 反馈 与原型 改进	1. 向潜在用户（学生/老师）展示原型页面并收集反馈（功能可用性、界面交互、信息布局） 2. 根据反馈调整UI层次、交互逻辑与信息结构 3. 记录典型需求点与问题清单	前端负责人+产品设计负责人	每人 3h×2 天	改进版原型图、用户反馈报告
	第 3- 5 天	系统 架构 与数据流 设计	1. 明确系统分层结构（数据采集层→清洗层→知识库层→前端展示层） 2. 评审架构的可扩展性与模块边界	技术负责人+后端负责人	每人 3h×2 天	系统架构图、数据流程图初稿
	第 6- 7 天	测试 计划 与阶段 总结	1. 编写《测试计划文档》：包括测试目标、测试类型（功能/接口/UI）、初步用例样例。 2. 讨论风险点（如API限流、数据更新频率）并制定应对方案。 3. 汇总优化最终开发计划	全体成员	每人 3h×2 天	测试计划文档、风险应对表、阶段总结报告

周次	时间段	主要目标	具体任务内容	相关成员	预计工时	预期成果
第12周 (Alpha阶段启动)	第1天	模块启动与分工	1. 建立爬虫与前端项目框架 2. 明确接口数据格式(JSON)	全体成员	每人3h×3天	接口文档、项目框架初始化
	第2-4天	数据采集与前端框架初版	1.完成校园通知、水电服务页面数据爬取 2. 清洗并结构化数据保存为JSON格式 3. 前端完成Vue路由、页面布局与组件初版	全体成员	每人3h×3天	爬虫与JSON知识库初稿、前端主页面可展示
	第5-6天	接入智能问答接口	1. 调通LLM API接口，实现知识库+prompt输入输出 2. 前端联调问答接口，支持输入查询	全体成员	每人3h×3天	Alpha版本初步闭环
	第7天	Alpha功能联调	1. 完成本地测试与初步性能验证 2. 修复爬虫与问答逻辑的兼容问题	全体成员	每人3h×3天	可运行Alpha版本初版、阶段小结报告

周次	时间段	主要目标	具体任务内容	责任人	预计工时	预期成果
第13周 (Alpha整合与测试)	第1-3天	系统整合与调试	1. 整合爬虫数据、知识库API与前端UI 2. 调试接口返回数据格式与响应时间	全体成员	每人3h×3天	Alpha可运行闭环Demo
	第4-6天	内部测试与验证	1. 编写测试用例（功能+接口），进行单元测试与接口测试 2. 收集错误日志，修复主要Bug	全体成员	每人3h×3天	Alpha测试报告、问题清单
	第7天	阶段总结与展示	1. 完成《Alpha阶段总结报告》 2. 内部展示Alpha版本，收集团队反馈	全体成员	每人3h×3天	阶段总结文档、展示Demo

2. 原有安排

周次	主要目标	具体任务内容	责任人	时间估计	输出成果
第10周	项目立项与准备阶段	1. 明确项目目标与功能范围，完成《需求规格说明书》初稿。 2. 绘制系统功能框架图与初步界面原型（包括主页、通知页、查询页）。 3. 确定前后端技术栈（Vue.js + Python爬虫 + JSON知识库 + LLM API）。 4. 建立项目仓库（GitHub/Coding），制定统一代码规范与接口标准。 5. 完成开发环境配置与数据采集测试脚本验证。	全体成员	20h	1. 需求规格说明书初稿 2. 原型设计初版 3. 仓库与协作平台搭建完成
第11周	深化设计与计划阶段	1. 收集潜在用户（学生/老师）对原型的反馈并改进交互逻辑。 2. 绘制系统架构图、数据流图及WBS工作分解结构。 3. 明确模块分工（爬虫模块、清洗模块、知识库模块、问答模块、前端展示模块）。 4. 制定《测试计划文档》，撰写测试目标与初步用例。	全体成员	25h	1. 完整原型图 2. 系统架构/WBS图 3. 模块分工与工期表 4. 测试计划文档
第12周	Alpha阶段启动（敏捷冲刺）	1. 按WBS分配任务，制定7天Alpha冲刺目标。 2. 完成数据采集与清洗模块初版（Python爬虫 + 正则提取 + JSON结构化存储）。 3. 前端完成页面框架搭建（Vue + 路由 + 页面切换）。 4. 接入LLM API，实现基础Prompt问答功能。 5. 每日Scrum Meeting，撰写冲刺日志。	全体成员	每人 3h×7天	1. Alpha版本核心模块初版 2. 7篇冲刺日志 3. 可运行代码版本
第13周	Alpha阶段收尾与整合	1. 集成爬虫、知识库与前端展示，实现基础闭环。 2. 完成接口调试与本地部署测试。 3. 撰写《Alpha测试报告》与阶段总结。 4. 展示Alpha Demo，收集组内评审意见。	全体成员	15h	1. 可运行Alpha Demo 2. Alpha测试报告 3. 阶段总结文档

周次	主要目标	具体任务内容	责任人	时间估计	输出成果
第14周	用户反馈与改进	1. 向部分学生开放Alpha Demo试用，收集使用反馈。 2. 优化LLM回答逻辑与知识检索精度。 3. 更新UI细节与关键词检索交互体验。 4. 撰写并发布《Alpha博客》。	全体成员	18h	1. 用户反馈汇总表 2. 改进版测试计划 3. Alpha博客发布版
第15周	项目反思与优化	1. 进行项目Postmortem分析，总结开发中出现的问题与改进建议。 2. 撰写《项目反思报告》与Alpha博客“事后分析篇”。 3. 确定Beta阶段优化方向（包括数据更新机制、前端搜索优化等）。	全体成员	10h	1. 事后分析报告 2. 团队反思总结 3. Beta阶段计划草案

3. 矫正计算方法

计算内容	原计划	修正后	差异说明
时间粒度	按“周”为单位（7天一阶段）	按“3天”为单位（细化为迭代周期）	修正后可更快反馈开发风险，符合敏捷冲刺节奏
任务拆解	模块级（爬虫、前端、API）	功能级（数据采集、清洗、接口调试、前端交互）	提高可追踪性与任务验收精度
成果检查	阶段性（Alpha完结后评估）	每3天产出一次中间成果（如JSON文件、Demo界面）	便于形成持续交付与滚动改进

计划修正公式：

修正后进度 = 原计划进度 × 任务完成率 × 调整系数

若第12周原计划完成度为 100%，实际任务完成率为 80%，调整系数 1.1，则修正后进度 = 100% × 0.8 × 1.1 = 88%，用于动态评估进度偏差并调整后续节奏

4. Issue截图

5. 团队分工

成员	角色定位	主要负责任务	项目分工内容	本周完成情况	本周完成感想
王韵清	算法开发	数据爬取与清洗	1. 校园通知、水电系统的爬取。 2. 设计数据清洗规则与格式模板。	调研了爬虫相关技术及校园信息网页结构	
曾钰仪	算法开发	知识库架构设计	1. 设计JSON知识库结构与存储格式。 2. 规划数据接口输入输出规范。	调研了知识库建立的相关技术及设计思路及接口设计规则	
徐伊彤	算法开发	数据清洗与接口标准设计	1. 完善爬取数据清洗规则（正则、文本清理）。 2. 编写数据处理脚本将原始数据转为结构化JSON。	调研了数据清洗相关规则及技术及数据处理技术	
吴佳童	前端开发	前端框架搭建与组件规划	1. 完成表单验证与数据绑定。 2. 测试前后端接口联调可用性。	调研了不同数据的传输方式、输出请求的相应处理	
张洁	前端开发	页面交互与路由配置	1. 搭建前端组件模板与公共样式，实现页面切换、菜单栏交互与响应式布局。 2. 与UI设计对接。	调研了前端界面的相关项目及设计思路	
李恺凝	UI/产品设计	原型设计与用户体验规划	1. 绘制初步原型，包括主页、通知页、查询页。 2. 设计交互逻辑图及风格指南。 3. 调研优化用户体验、色彩和排版。	完成并优化了原型的初步设计，产出网页整体的流程规划用于后续项目推进	
戴军霞	项目测试	文档与规范制定	1. 撰写《需求规格说明书》初稿，明确系统功能与性能需求。 2. 建立测试计划框架及初步测试用例模板。	调研了知识库、爬虫相关的优化、测试方法	