校务问答机器人项目计划

G07小组

组长：郭伟进

组员：范品樟

组员：阮精特

组员：麻克强

组员：林锴

目录

[校务问答机器人项目计划 1](#_Toc643)

[一、引言 3](#_Toc16819)

[1.1 目的 3](#_Toc12059)

[1.2 范围 3](#_Toc3238)

[1.3 定义、简写和缩略语 3](#_Toc20142)

[1.4 综述 3](#_Toc21707)

[二、总体描述 4](#_Toc2377)

[2.1 产品描述 4](#_Toc21763)

[2.2 产品功能 4](#_Toc15655)

[2.3 用户特点 5](#_Toc19959)

[2.4 约束 5](#_Toc32319)

[2.5 假设和依赖关系 6](#_Toc18924)

[2.6 需求分配 7](#_Toc841)

[三、具体需求 7](#_Toc11316)

[3.1外部接口需求 7](#_Toc29661)

[3.2功能需求 9](#_Toc20817)

[3.3性能需求 15](#_Toc26778)

[3.4设计约束 15](#_Toc27904)

[3.5软件系统属性 16](#_Toc13688)

[3.6其他需求 17](#_Toc32197)

[四、项目分工 17](#_Toc6062)

[4.1角色与职责定义 17](#_Toc10674)

[4.2任务分配方案 19](#_Toc27611)

[4.3协作机制 22](#_Toc13944)

[4.3活动安排 22](#_Toc30281)

一、引言

1.1 目的

本项目旨在开发一个校务问答机器人，向全体师生提供校园生活、学习、教学、科研等综合性校务问答服务，便捷师生的校园生活。同时支持点赞、吐槽、热搜排序、 评价反馈、交流讨论、聊天记录等功能，丰富产品的使用体验、以及常用的应用管理员功能，便于管理。

本《项目计划文档》的预期读者是：

学校信息中心相关人员

学校教务处相关人员

学工部相关人员

项目组所有人员

1.2 范围

本软件为校务问答机器人，用于提供校园生活、学习、教学、科研等综合性校务问答服务。通过LLM（大语言模型），系统旨在为校内师生提供生活上的便利，以提升师生生活满意度。

1.3 定义、简写和缩略语

LLM（大语言模型）：基于海量数据训练的 AI 系统，能理解生成人类语言，如 GPT-4。

API（应用程序接口）：软件间通信桥梁，通过标准化协议实现功能调用与数据交互。

ISO 8601：国际日期格式标准，统一表示年 - 月 - 日（如 2025-03-09）及时间，消除跨区域歧义。

1.4 综述

本SRS按照以下逻辑组织内容：

（1）基础信息（第1章）：

第1章（引言）：阐述项目背景、目标和范围；

（2）总体描述（第2章）：

第2章（总体描述）：对产品基础信息和约束的介绍；

（3）具体需求（第3章）：

第3章（具体需求）：描述产品外部接口、功能、性能等需求；

二、总体描述

2.1 产品描述

校务问答机器人作为校园信息化服务的一部分，与校园现有的信息系统（如教务管理系统、学生信息管理系统等）存在数据交互接口。通过这些接口，机器人能够获取最新的校园资讯、课程安排、成绩信息等，为师生提供准确的问答服务。在用户界面方面，将采用简洁明了的设计风格，适配多种终端设备（如电脑、手机、平板），方便师生随时随地使用。支持文字、语音两种交互方式，语音交互需具备语音识别和语音合成功能，识别准确率和合成语音的自然度需达到较高水平。硬件方面，需确保服务器具备足够的计算资源和存储容量，以支持 LLM 的运行和大量用户的并发访问。

软件方面，依赖于所选的 LLM 服务提供商的软件平台，以及开发过程中使用的各类开发工具和框架。通信方面，需保障网络通信的稳定性，确保用户请求和机器人响应能够及时传输。内存方面，合理优化程序代码，减少内存占用，避免因内存不足导致系统崩溃。运行方面，要求机器人能够 7×24 小时不间断运行，具备自动监控和故障恢复机制。现场适应性方面，可根据不同学院的特殊需求（如课程设置差异、作息时间不同等）进行定制化配置.

2.2 产品功能

（1）问答功能：

接收师生输入的关于校园生活、学习、教学、科研等方面的问题，利用 LLM 技术进行理解和分析，快速给出准确、详细的回答。回答内容应涵盖常见问题的标准答案、相关政策解读、办事流程指引等。

（2）互动功能：

支持点赞、吐槽功能，师生可对机器人的回答进行评价，表达满意或不满意。提供热搜排序功能，根据问题的搜索热度进行排序，方便师生快速了解大家关注的热点问题。具备评价反馈功能，师生可以提交对机器人回答的具体意见和建议，促进机器人不断优化。实现交流讨论功能，师生之间可以针对某个问题展开讨论，分享经验和见解。同时，保存聊天记录，方便师生随时查看历史交流内容

（3）管理功能：

管理员可对机器人的回答内容进行审核和编辑，确保回答的准确性和规范性。能够管理用户信息，包括添加、删除、修改用户权限等操作。还可以对系统的运行状态进行监控，查看用户访问量、问题类型分布等数据，以便进行系统优化和资源调配

2.3 用户特点

（1）全体师生：

教育程度涵盖从本科到研究生等不同层次，具备一定的计算机操作能力和网络使用经验，但对技术细节的了解程度参差不齐。对校园信息的需求多样且迫切，希望能够快速、准确地获取所需信息，提高学习和工作效率。

（2）校园信息化相关人员：

具有专业的信息技术知识和丰富的校园信息化建设经验，熟悉校园各业务系统的运作流程。对机器人的功能和性能有较高的期望，关注机器人与现有系统的集成和数据交互，以及如何通过机器人提升校园信息化服务水平。

2.4 约束

（1）法规政策：

严格遵守国家关于教育信息安全、个人隐私保护等方面的法规政策，确保师生的个人信息和校园数据不被泄露。

（2）硬件局限：

服务器的计算能力和存储容量可能受到预算限制，在一定程度上影响机器人对大规模并发请求的处理能力和数据存储能力。同时，网络带宽的限制可能导致数据传输延迟，影响用户体验。

（3）与其他应用的接口：

与校园现有信息系统的接口可能存在兼容性问题，需要投入额外的开发资源进行适配和调试。不同系统的数据格式和接口规范不一致，增加了数据交互的难度。

（4）并行操作：

在高并发访问情况下，需要确保机器人系统的稳定性和数据一致性，避免出现数据错误或系统崩溃。这对系统的并发处理能力和资源调度算法提出了较高要求。

（5）审核功能：

管理员对回答内容的审核需要及时、准确，以保证回答的质量和合规性。但审核流程可能会影响信息发布的及时性，需要在审核效率和质量之间找到平衡。

（6）控制功能：

系统应具备完善的权限控制机制，确保只有授权用户能够进行相应的操作，防止非法访问和数据篡改。但过于严格的权限控制可能会影响用户的操作便利性，需要合理设计权限体系。

（7）高级语言需求：

开发过程中可能需要使用特定的编程语言和开发框架，以满足项目的技术需求。但这可能对开发团队的技术能力提出较高要求，增加开发成本和风险。

（8）可靠性需求：

机器人需要具备高可靠性，能够稳定运行，确保在大量用户并发访问和长时间运行的情况下不出现故障。这对系统的架构设计、硬件选型和软件测试提出了严格要求。

（9）应用的关键性：

作为校园信息化服务的重要组成部分，机器人的正常运行对师生的学习和工作具有重要影响。一旦出现故障，可能会导致师生获取信息困难，影响校园的正常运转。

（10）安全和保密安全考虑：

加强系统的安全防护，防止黑客攻击、数据泄露等安全事件的发生。对涉及师生个人隐私和敏感信息的数据进行加密存储和传输，确保数据安全。

2.5 假设和依赖关系

（1）假设：

假设所选的 LLM 服务提供商能够持续提供稳定、高效的服务，其服务质量和性能不会出现大幅波动。假设校园网络环境能够保持相对稳定，网络带宽能够满足机器人的数据传输需求。假设校园现有信息系统的数据准确性和完整性能够得到保障，为机器人提供可靠的数据来源。

（2）依赖关系：

本项目依赖于所选的 LLM 服务提供商的 API 接口，需要与该接口进行稳定、高效的对接。同时，依赖于校园现有信息系统的开放接口，以便获取所需的校园数据。此外，开发过程中依赖于相关的开发工具和框架，以及服务器硬件资源的支持。

2.6 需求分配

（1）近期需求：

优先实现基本的问答功能，确保能够准确回答常见的校园问题。完成点赞、吐槽、评价反馈等基础互动功能的开发，以及管理员对回答内容的审核功能。

（2）中期需求：

优化问答算法，提高回答的准确性和全面性。实现热搜排序、交流讨论功能，增强用户之间的互动体验。完善管理员对用户信息的管理功能和系统运行状态的监控功能。

（3）远期需求：

根据用户反馈和业务发展，不断拓展机器人的功能，如增加智能推荐、个性化定制等功能。持续优化系统性能，提高并发处理能力和响应速度，以适应校园规模的扩大和用户需求的增长。

三、具体需求

3.1外部接口需求

（1）用户界面  
交互设计：

采用简洁直观的界面布局，确保全体师生（包括不同年龄段、技术熟悉程度各异的人群）都能轻松上手。例如，问答输入框突出显示在页面顶部，便于用户快速定位和输入问题；答案展示区域清晰划分，对不同类型的回答（如文本、链接、图片等）进行合理排版。​

响应式设计：

支持多终端访问，包括电脑端（各类操作系统的浏览器）、移动端（手机和平板，适配主流移动操作系统如 iOS 和 Android）。界面能根据不同设备屏幕尺寸自动调整布局，保证信息展示完整且美观。​

操作反馈：

在用户输入问题后，立即给予输入确认提示（如光标闪烁或显示 “正在输入” 状态）；在问题提交后，展示加载动画，告知用户系统正在处理请求，避免用户重复操作。对于点赞、吐槽、评价反馈等操作，及时显示操作成功或失败的提示信息。

（2）硬件接口

服务器端:

服务器需具备强大的计算能力和内存，以应对大量师生同时访问及大语言模型运算的需求。建议配置高性能多核 CPU、大容量内存（如 32GB 及以上），以及高速存储设备（如固态硬盘 SSD），确保数据读写速度快，减少系统响应延迟。

终端设备:

支持与常见终端硬件设备兼容，如电脑的键盘、鼠标，手机的触摸屏、麦克风等。确保用户能通过各种便捷的方式与问答机器人进行交互。

（3）软件接口

大语言模型接口：

与选定的 LLM 技术提供商建立稳定可靠的接口连接，确保能够高效调用模型进行问题解答。接口需具备良好的扩展性，以便在未来根据需求切换或升级大语言模型。同时，要严格遵循模型提供商的使用规范和安全要求，保障数据传输安全和模型使用合法性。

校内信息系统接口：

与校园信息化相关的各类系统（如教务处的教务管理系统、学工部的学生管理系统、信息中心的校园资源管理系统等）进行对接。通过标准化的数据接口，实现数据的实时共享和交互，使问答机器人能够获取最新的校园生活、学习、教学、科研等信息，为师生提供准确的答案。例如，与教务系统接口可获取课程安排、考试时间等信息；与学生管理系统接口可查询学生个人成绩、奖惩记录等。

第三方服务接口：

集成一些常用的第三方服务接口，如地图服务接口（用于查询校园内建筑物位置等信息）、天气服务接口（提供校园所在地区天气情况）等，丰富问答机器人的服务内容。确保第三方接口的稳定性和数据准确性，并遵循相关服务协议。

（4）通信接口

第三方服务接口：

集成一些常用的第三方服务接口，如地图服务接口（用于查询校园内建筑物位置等信息）、天气服务接口（提供校园所在地区天气情况）等，丰富问答机器人的服务内容。确保第三方接口的稳定性和数据准确性，并遵循相关服务协议。

实时通信功能：

对于交流讨论功能，支持 WebSocket 等实时通信协议，实现用户之间消息的即时推送，营造流畅的交流环境。确保在多人同时参与讨论时，消息收发及时准确，不出现消息丢失或延迟过高的情况。

3.2功能需求

1. 功能描述

|  |  |  |
| --- | --- | --- |
| 功能 | 具体描述 | 目标 |
| 用户登录 | 提供用户的注册、登录和个人信息管理 | 确保用户安全、便捷地使用平台，并便于后续任务操作 |
| 问答模块 | 用户通过文本或语音输入问题，系统利用自然语言处理技术分析问题，从知识库表和 LLM 模型获取信息生成回答。回答生成后，将问题 ID、提问时间、用户 ID、问题内容、回答内容记录到用户提问记录表中。 | 为用户提供准确、快速的校园生活、学习、教学、科研等方面问题的解答，满足用户对校园信息的需求。 |
| 互动模块 | 具备对回答进行点赞、评价反馈、吐槽功能。 | 收集用户意见，促进系统不断优化，提升服务质量，满足用户需求。 |
| 讨论与聊天模块 | 用户还有交流讨论、聊天等互动功能 | 营造良好的交流氛围，促进用户之间的信息共享和经验交流，丰富校园信息获取途径。 |
| 管理模块 | 管理员可以进行回答审核、用户管理、系统运行监控、知识库管理等 | 确保平台运行的可靠性和规范性，为用户提供可靠的信息服务。 |

（2）过程描述

1. 用户登陆

用户操作：用户打开校务问答机器人应用或网页，在登录界面输入注册时使用的手机号或邮箱，以及对应的验证码（通过短信或邮件获取），或者输入密码进行登录。若用户尚未注册，则点击注册按钮，进入注册流程，填写手机号、邮箱、密码等注册信息。

系统响应：

登录验证：系统接收到用户的登录请求后，将用户输入的手机号 / 邮箱与数据库中用户表的对应字段进行比对，找到匹配的用户记录。同时，根据用户 ID 从用户表中获取用户的相关信息（如用户身份、所在学院 / 部门等），用于后续个性化服务。

注册处理：当用户进行注册操作时，系统先对用户输入的手机号、邮箱进行格式校验，确保其符合规范。

1. 用户问答

用户操作：用户通过问答界面的文本输入框直接输入问题，或者点击语音输入按钮，说出问题，由系统将语音转换为文本。问题内容围绕校园生活、学习、教学、科研等方面，如 “明天图书馆是否开放？”“本学期的课程表在哪里查看？” 等。​

系统响应：​

问题接收与预处理：系统接收用户输入的问题文本，同时获取用户 ID。调用自然语言处理模块，对问题进行分词、词性标注、命名实体识别等操作，提取关键信息

答案检索与生成：根据提取的关键信息，在知识库表中进行匹配查询，查找是否有与之对应的标准化问题描述。然后整合知识库答案和 LLM 模型建议，生成最终回答内容

1. 点赞、评价反馈、吐槽

用户操作：​

点赞：用户可以给机器人的回复点赞。​

评价反馈：用户可以给机器人的回复评价

吐槽：若用户对回答不满意，点击吐槽按钮。​

系统响应：​

点赞处理：系统接收到用户的点赞请求，将点赞记录插入点赞表中，同时返回点赞成功提示信息。​

评价反馈处理：系统接收用户提交的评价反馈内容将评价反馈记录插入评价反馈表中，并返回评价反馈提交成功提示信息。

吐槽处理：系统接收到用户的吐槽请求，将吐槽记录插入吐槽表中，并返回吐槽成功提示信息。

1. 讨论、回复

用户操作：​

发起讨论：用户可以在界面发起新的讨论主题。

回复他人：在已有的讨论区中，用户可以对其他用户的内容进行回复。

系统响应：​

讨论发言处理：系统接收用户提交的讨论发言请求，获取用户 ID、问题 ID 和讨论发言内容。将讨论发言记录插入交流讨论表中，并返回发言提交成功提示信息。​

回复处理：系统接收用户的回复请求，获取用户 ID、问题 ID、回复内容，生成讨论发言记录，将讨论发言记录插入交流讨论表中。

1. 与他人聊天

用户操作：用户在聊天界面选择与之聊天的对象（若系统支持多用户聊天，可选择多个对象。​

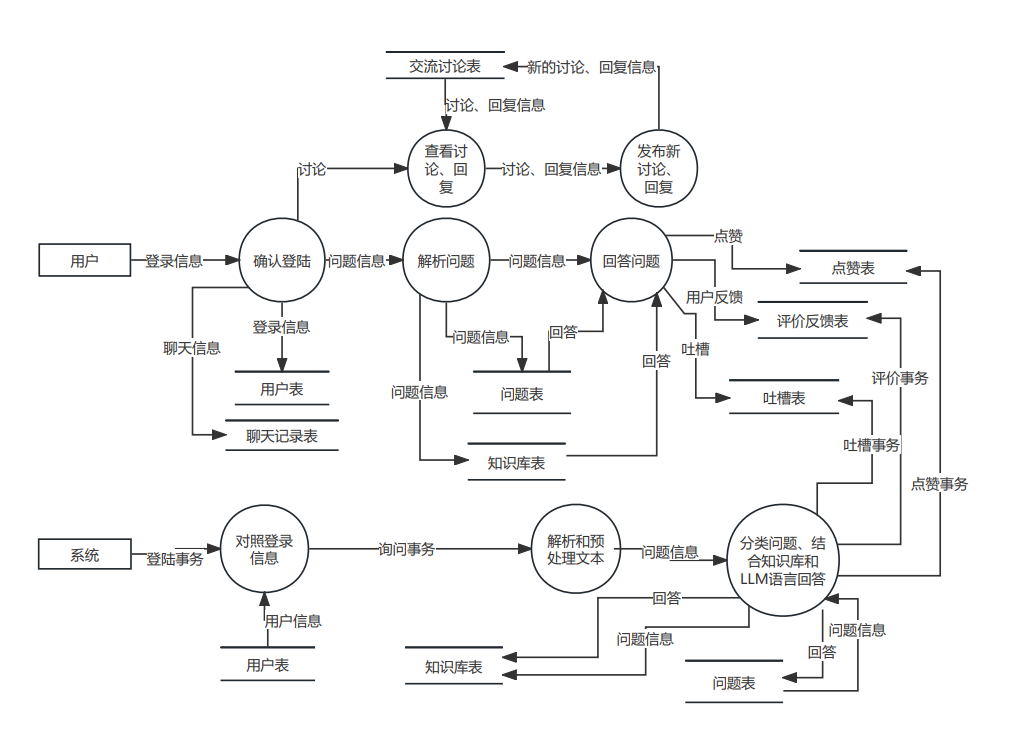
系统响应：​

聊天记录保存：系统每次接收到用户输入的聊天内容，获取用户 ID、问题 ID（若有）和聊天内容。

（3）数据流分析

用户登录成功后，可通过文本输入框或语音输入提出校园生活、学习、教学、科研等方面的问题，系统接收问题文本后调用自然语言处理模块提取关键信息，先在知识库表匹配查询，若未找到匹配项则调用 LLM 服务提供商接口获取回答建议，整合后生成回答内容并记录到用户提问记录表，最后返回给用户。用户对回答可进行点赞、评价反馈、吐槽操作，系统会根据用户操作生成相应记录并存储到点赞表、评价反馈表、吐槽表。

用户能在问答界面或讨论区针对问题发起新讨论或回复他人，系统接收请求后生成讨论发言记录插入交流讨论表，回复时还会向被回复用户推送通知。用户还可在聊天界面选择对象输入聊天内容，系统将聊天记录保存到聊天记录表，若与机器人聊天，系统按类似问答模块方式处理并回复，用户也可随时查询聊天记录 。



（4）数据词典

a)用户表（User）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| user\_id | 用户唯一标识，与校园统一身份认证系统对接 | 字符串或数字 | 是 |
| username | 用户真实姓名 | 文本 | 否 |
| user\_type | 用户身份，取值为 “教师”“学生”“管理员” | 枚举 | 否 |
| registration\_date | 用户在本系统的注册日期 | 日期 | 否 |
| college\_major (学生) | 学生所在学院及专业 | 文本 | 否 |
| department (教师) | 教师所在工作部门 | 文本 | 否 |

b)问题表（question）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| question\_id | 问题唯一标识符 | UUID 字符串 | 是 |
| question\_timestamp | 用户提问时间 | 遵循 ISO 8601 格式的日期时间 | 否 |
| user\_id | 提问用户的唯一 ID | 字符串或数字 | 否 |
| question\_content | 用户输入的原始问题文本 | 文本 | 否 |
| answer\_content | 系统生成的针对该问题的回答文本 | 文本 | 否 |

c)知识库表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| entry\_id | 知识库条目唯一标识 | UUID 字符串 | 是 |
| question\_description | 标准化的问题表述 | 文本 | 否 |
| answer\_details | 针对问题的详细解答内容 | 文本 | 否 |
| tags | 用于分类标记知识库条目的标签数组 | 数组（每个元素为字符串） | 否 |
| creation\_date | 知识库条目创建时间 | 遵循 ISO 8601 格式的日期 | 否 |
| modification\_date | 知识库条目最后一次修改时间 | 遵循 ISO 8601 格式的日期 | 否 |

d)点赞表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| like\_id | 点赞记录的唯一标识 | UUID 字符串 | 是 |
| user\_id | 执行点赞操作的用户 ID | 字符串或数字 | 否 |
| question\_id | 被点赞回答所对应的问题 ID | UUID 字符串 | 否 |
| like\_time | 点赞操作发生的时间 | 遵循 ISO 8601 格式的日期时间 | 否 |

e)吐槽表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| dislike\_id | 吐槽记录的唯一标识 | UUID 字符串 | 是 |
| user\_id | 执行吐槽操作的用户 ID | 字符串或数字 | 否 |
| question\_id | 被吐槽回答所对应的问题 ID | UUID 字符串 | 否 |
| dislike\_time | 吐槽操作发生的时间 | 遵循 ISO 8601 格式的日期时间 | 否 |

f)热搜排行榜表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| hot\_search\_id | 热搜记录的唯一标识 | UUID 字符串 | 是 |
| question\_id | 热搜问题的 ID | UUID 字符串 | 否 |
| search\_count | 该问题的搜索次数，用于衡量热度 | 整数 | 否 |
| last\_update\_time | 热度数据的最后更新时间 | 遵循 ISO 8601 格式的日期时间 | 否 |

g)评价反馈表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| feedback\_id | 评价反馈记录的唯一标识 | UUID 字符串 | 是 |
| user\_id | 提交评价反馈的用户 ID | 字符串或数字 | 否 |
| question\_id | 针对的问题 ID | UUID 字符串 | 否 |
| feedback\_content | 用户提交的详细评价反馈文本 | 文本 | 否 |
| feedback\_time | 评价反馈提交的时间 | 遵循 ISO 8601 格式的日期时间 | 否 |

h)交流讨论表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| discussion\_id | 交流讨论记录的唯一标识 | UUID 字符串 | 是 |
| user\_id | 参与讨论的用户 ID | 字符串或数字 | 否 |
| question\_id | 讨论所围绕的问题 ID | UUID 字符串 | 否 |
| discussion\_content | 用户的讨论发言内容 | 文本 | 否 |
| discussion\_time | 发言时间 | 遵循 ISO 8601 格式的日期时间 | 否 |
| reply\_to | 回复的目标发言 ID（若为回复他人，则记录对方发言 ID；若为新主题发言，则为空） | UUID 字符串 | 否 |

1. 聊天记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段含义 | 字段类型 | 是否主键 |
| chat\_record\_id | 聊天记录的唯一标识 | UUID 字符串 | 是 |
| user\_id | 参与聊天的用户 ID | 字符串或数字 | 否 |
| question\_id | 相关问题 ID（若聊天围绕某问题展开） | UUID 字符串 | 否 |
| chat\_content | 聊天内容文本 | 文本 | 否 |
| chat\_time | 聊天发生的时间 | 遵循 ISO 8601 格式的日期时间 | 否 |

3.3性能需求

（1）响应时间

在正常负载情况下，用户提交问题后，系统应在 1 秒内给出初步响应，并在 3 秒内返回完整的答案。对于复杂问题或涉及大量数据查询的情况，最长响应时间不得超过 5 秒，以保证用户体验流畅，避免用户因等待时间过长而流失。​

（2）吞吐量

系统能够支持至少10000名师生同时在线访问和使用问答机器人服务，确保在高并发情况下系统性能不出现明显下降。随着校园师生数量的增长或使用场景的拓展，系统应具备良好的横向扩展能力，能够方便地增加服务器资源来提升吞吐量。​

（3）准确性

基于大语言模型给出的答案准确率应达到95% 以上。对于常见的校园相关问题，确保回答准确无误；对于复杂或模糊问题，能提供合理的解释和引导。定期对模型进行优化和训练，利用校内真实数据不断提高答案的准确性和针对性。​

（4）稳定性

系统应具备高度的稳定性，7×24 小时不间断运行，每月系统故障时间不超过 5分钟。通过采用冗余设计、负载均衡、数据备份与恢复等技术手段，确保在硬件故障、网络波动等异常情况下系统仍能正常工作，保障校园师生的正常使用。

3.4设计约束

（1）法律法规

严格遵守国家和地方相关法律法规，特别是关于数据隐私保护、网络安全等方面的规定。确保师生的个人信息（如姓名、学号、成绩等）在收集、存储、使用和传输过程中的安全性和合法性，未经用户明确授权，不得将用户信息泄露给第三方。​

（2）技术标准

遵循行业内通用的软件开发技术标准和规范，如代码编写规范、接口设计规范等，保证系统的可维护性、可扩展性和兼容性。在选择技术框架、开发工具和数据库管理系统时，优先考虑主流且成熟的产品，便于技术团队进行开发和后续维护。​

（3）成本限制

在项目开发过程中，需充分考虑成本因素。在满足功能和性能需求的前提下，合理选择硬件设备、软件工具和技术方案，避免过度投入。对项目预算进行严格把控，确保各项费用支出清晰透明，如有必要可进行成本效益分析，评估不同方案对项目整体成本的影响。

3.5软件系统属性

（1）可维护性

系统架构设计应具备良好的模块化和分层结构，各功能模块职责明确，接口清晰。代码编写遵循规范，添加详细注释，便于开发团队成员理解和维护。定期进行代码审查和优化，及时修复潜在的软件缺陷，确保系统长期稳定运行且易于维护升级。​

（2）可扩展性

随着校园信息化建设的不断发展和师生需求的变化，问答机器人系统应具备良好的可扩展性。能够方便地添加新的功能模块（如增加新的校内服务接口、优化用户交互功能等），扩展数据存储容量，以及适应不同规模的用户访问量。在系统设计阶段，充分考虑未来的扩展需求，预留相应的接口和数据结构。​

（3）可移植性

系统应具备一定的可移植性，能够在不同的服务器操作系统（如 Linux、Windows Server 等）上部署运行，且无需进行大规模的代码修改。同时，确保移动端应用能够在主流移动操作系统平台上稳定运行，减少因操作系统差异带来的兼容性问题。

3.6其他需求

（1）数据更新机制

建立完善的数据更新机制，确保与校园生活、学习、教学、科研等相关的数据及时准确更新。对于校内信息系统接口获取的数据，根据数据的变化频率设定合理的更新周期（如课程安排每日更新，校园活动信息实时更新等）。定期对大语言模型进行微调训练，使其能更好地适应校园特定领域的知识和问题。​

（2）安全审计

设置安全审计功能，记录用户的操作行为（如提问内容、点赞吐槽记录、评价反馈等）以及系统的关键操作日志（如数据查询、接口调用、模型运算等）。通过安全审计，能够及时发现潜在的安全风险和异常行为，为系统安全维护提供依据。同时，审计数据应妥善保存，保存期限根据相关法律法规和学校规定执行。​

（3）多语言支持

考虑到校园内可能存在国际学生和教师，系统应支持多语言功能，至少包括中文和英文。用户可根据自身需求选择语言界面，问答机器人能够识别用户输入的语言，并以相应语言给出回答。在多语言支持方面，需确保翻译的准确性和流畅性，不影响用户正常使用。

四、项目分工

4.1角色与职责定义

（1）前端工程师

| ****职责项**** | ****描述**** | ****工具要求**** |
| --- | --- | --- |
| 界面开发 | 实现 UI/UX 设计稿 | React/Vue |
| 兼容性测试 | 多浏览器 / 设备适配 | BrowserStack |
| 交互逻辑实现 | 复杂前端交互功能（如表单验证） | Redux/Angular |

（2）后端工程师

| ****职责项**** | ****描述**** | ****工具要求**** |
| --- | --- | --- |
| API 开发 | 设计与实现后端接口 | Spring Boot/Django |
| 数据库管理 | 数据库设计、优化与维护 | MySQL/PostgreSQL |
| 服务端性能优化 | 高并发场景下的服务端响应优化 | Redis/Nginx |

（3）模型工程师

| ****职责项**** | ****描述**** | ****工具要求**** |
| --- | --- | --- |
| 模型调优 | 提升模型精度与泛化能力 | MLflow |
| 特征工程 | 数据特征提取与筛选 | Featuretools |
| 模型监控 | 跟踪模型线上性能与异常 | Prometheus |

（4）数据工程师

| ****职责项**** | ****描述**** | ****工具要求**** |
| --- | --- | --- |
| 数据管道建设 | 设计与维护数据采集、清洗、存储流程 | Apache Kafka |
| 数据质量监控 | 检测数据缺失、异常并修复 | Great |
| 数据可视化 | 制作数据看板与报表 | Tableau/Power BI |

（5）需求工程师

| ****职责项**** | ****描述**** | ****工具要求**** |
| --- | --- | --- |
| 需求分析 | 收集、整理用户需求并转化为文档 | Axure/Visio |
| 需求评审 | 组织需求澄清会议与优先级排序 | Jira/Confluence |
| 用户故事编写 | 拆分需求为可执行的用户故事 | PBI/INVEST 原则 |
| 需求跟踪 | 监控需求变更与实现进度 | VersionOne |
| 业务流程优化 | 分析现有流程并提出改进方案 | BPMN/ProcessOn |

4.2任务分配方案

（1）需求阶段任务清单

对接需求提出方：

与信息中心、教务处、学工部等部门沟通，梳理校园信息化场景下的核心问答需求（如教学安排、学籍管理、校园生活服务等）。

师生用户调研：

设计问卷或访谈提纲，调研师生对校务问答的高频问题（如考试安排、奖学金政策、图书馆资源使用等）。分析师生对功能的期望（如交互方式、反馈机制、数据安全等）。

需求分类与筛选：

按功能模块分类，区分核心问答功能（教学、生活、科研）、辅助功能（点赞、吐槽、评价反馈）、管理功能（管理员后台）。筛选高频需求与优先级，剔除不合理或难实现的需求。

技术需求梳理：

明确基于 LLM 技术的实现要求，如模型调用接口、数据适配规则、响应速度等。梳理原型开发模式下多轮反馈的需求确认机制。

概念原型设计：

设计问答界面、功能入口（点赞 / 吐槽 / 搜索）、管理员后台框架。并描述交互逻辑（如用户提问→机器人应答→用户反馈）、原型迭代计划。

1. 开发阶段任务清单

技术架构与环境搭建：

| ****任务名称**** | ****任务描述**** | ****交付物**** |
| --- | --- | --- |
| 技术栈选型与环境配置 | 确定后端框架、前端技术、数据库，搭建开发环境。 | 技术方案文档、开发环境配置说明 |
| LLM 接口对接开发 | 集成大语言模型（如调用 API 或部署本地模型），实现问题输入与答案输出的基础逻辑。 | LLM 调用接口代码、测试 Demo |

核心功能开发：

| ****任务名称**** | ****任务描述**** | ****交付物**** |
| --- | --- | --- |
| 问答功能开发 | 实现用户提问解析、模型响应处理、答案结构化输出（如课程查询、学籍政策解答）。 | 问答功能模块代码、测试用例 |
| 交互功能开发 | 开发点赞、吐槽、评价反馈接口，实现热搜排序逻辑，记录用户聊天记录。 | 交互功能 API 文档、功能代码 |
| 管理员功能开发 | 设计管理员后台，实现内容审核、FAQ 管理、用户反馈处理等功能。 | 管理员后台代码、操作手册 |

前端开发：

| ****任务名称**** | ****任务描述**** | ****交付物**** |
| --- | --- | --- |
| 问答界面开发 | 设计用户端界面，包含提问输入框、答案展示区、点赞 / 吐槽按钮等。 | 前端页面代码、界面原型 |
| 管理后台开发 | 搭建管理员操作界面，支持内容管理、数据统计等功能。 | 管理后台页面代码、交互说明 |

测试与优化：

| ****任务名称**** | ****任务描述**** | ****交付物**** |
| --- | --- | --- |
| 功能测试 | 对问答准确性、交互功能、管理员操作进行测试，修复 Bug。 | 测试报告、问题修复记录 |
| 性能测试与优化 | 测试系统响应速度、并发能力，优化 LLM 调用效率与数据库查询性能。 | 性能优化方案、优化后代码 |
| 兼容性测试 | 验证机器人在不同浏览器、终端（PC / 手机）的适配性。 | 兼容性测试报告 |

部署与上线准备：

| ****任务名称**** | ****任务描述**** | ****交付物**** |
| --- | --- | --- |
| 服务器部署 | 部署后端服务、前端页面，配置域名与服务器环境。 | 部署文档、服务器配置记录 |
| 上线前检查 | 检查功能完整性、数据安全性，准备应急方案。 | 上线检查清单、应急预案 |

4.3协作机制

在 “校务问答机器人” 开发阶段，协作机制如下：通过每日线上会同步任务进度与卡点问题，利用GitHub实时更新代码、需求变动；前端、后端、测试组按分工推进，定期召开跨组会议对齐开发节奏；遇到技术难点或需求调整时，由项目经理牵头组织讨论，结合团队头脑风暴与外部资源（如 LLM 技术支持）快速制定解决方案，确保开发高效协同。

4.3活动安排

项目计划在5月31日前完成，在3月24日前完成技术架构的搭建工作，之后有条不紊的进行开发任务。

