# JAD会议记录

## **一、会议基本信息**

1. ****会议主题****：深入研讨校务问答机器人各功能模块需求、可行性及风险应对策略
2. ****会议时间****：4/6
3. ****会议地点****：理四1楼自习室
4. ****参会人员****：小组成员、孙曙迎老师（线上参会）、此前访谈的一部分同学

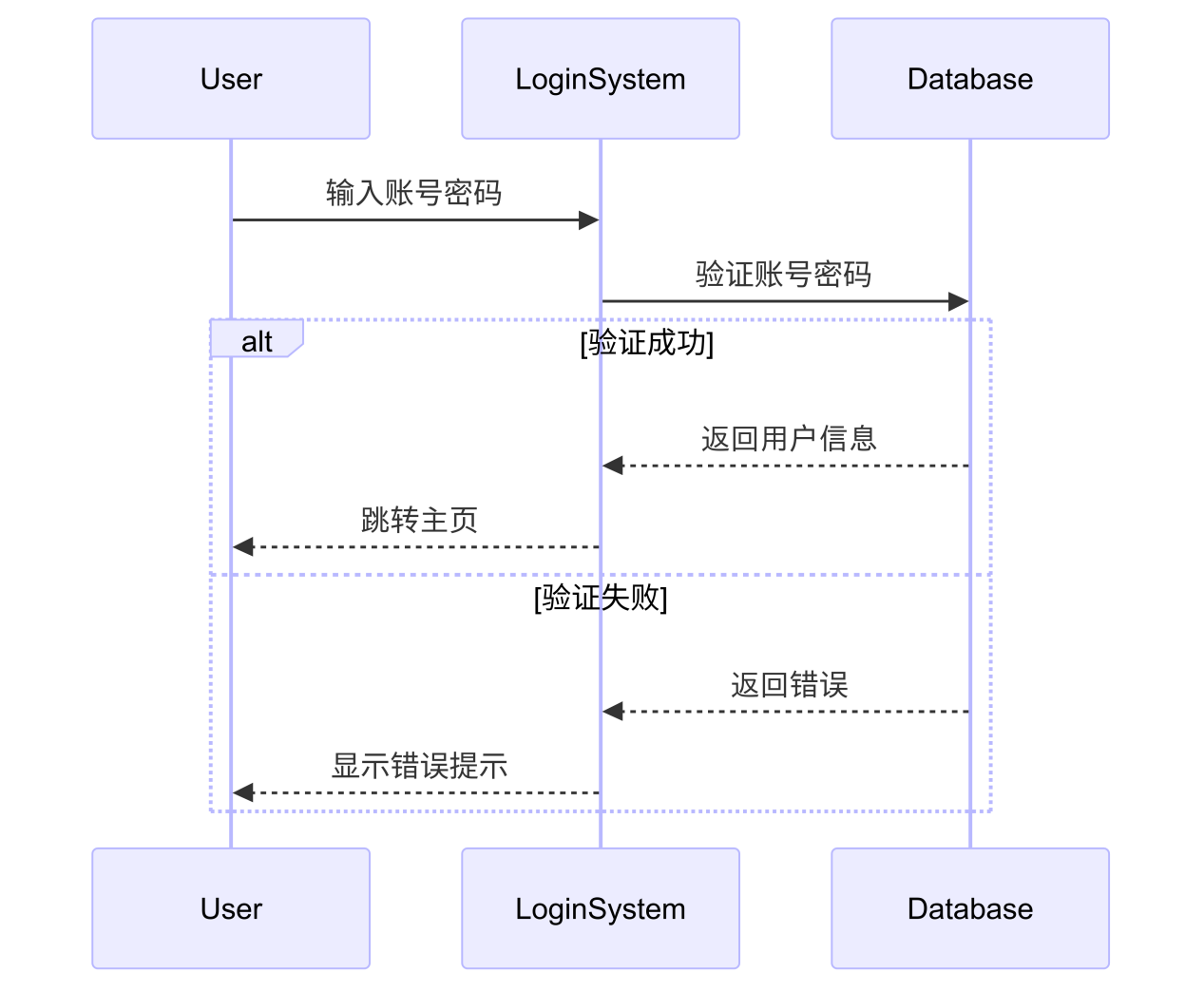
## **二、会议议程及内容**

### **（一）开场与会议目的介绍（5min）**

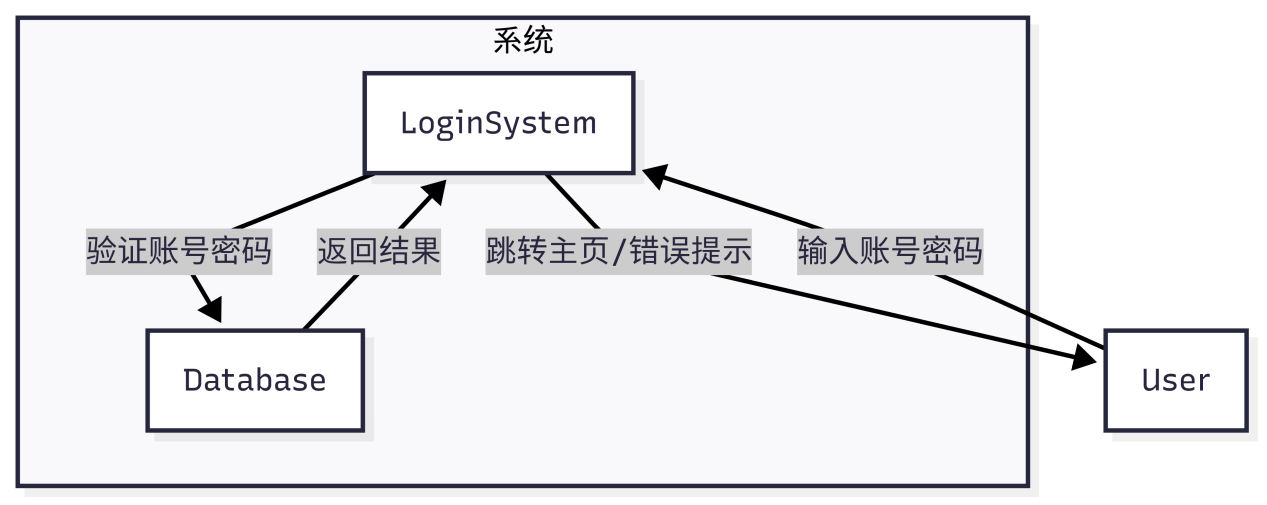
1. ****主持人开场****：由产品经理担任主持人，介绍参会人员，强调本次会议聚焦校务问答机器人功能模块，旨在明确需求、评估可行性并制定风险应对策略，为项目推进奠定基础。
2. ****会议目标阐述****：详细说明会议期望达成的成果，包括对各功能模块需求的深度理解与确认、技术实现路径的初步规划、风险识别与应对方案的制定，确保项目开发方向的准确性和高效性。

### **（二）用户登录模块讨论（20min）**

1. ****结构化需求详细解读（10min）****
   1. ****业务规则剖析****：产品经理详细讲解用户登录模块的业务规则，明确用户必须输入有效的学号 / 工号和密码，系统通过与数据库中加密密码比对进行验证。针对 “记住登录状态” 功能，详细讨论了 Cookie 和 Token 两种实现方式的优缺点，倾向于在安全性要求较高的场景下优先使用 Token，以降低因 Cookie 泄露导致的安全风险。
   2. ****输入输出要求明确****：进一步明确输入信息的格式要求，如账号为字符串类型，密码为加密字符串，强调加密算法需采用行业标准的如 SHA - 256 等，以保障密码存储安全。对于输出，详细规定了登录失败时错误提示的具体内容和格式，如 “账号或密码错误，请重新输入”，并明确错误提示应在登录页面显眼位置展示，提高用户体验。
2. ****Mermaid 顺序图与协作图分析（10min）****



* 1. ****顺序图讲解****：开发团队成员结合 Mermaid 顺序图，逐步展示用户登录过程中用户、登录系统和数据库之间的交互流程。详细解释在验证账号密码环节，若验证成功，系统如何获取用户信息并跳转主页；若验证失败，如何返回错误信息并在前端显示。同时，讨论了在高并发场景下，顺序图中各环节可能出现的性能瓶颈，如数据库查询压力过大等问题。



* 1. ****协作图分析****：通过协作图，清晰呈现系统各部分在用户登录过程中的协作关系。与会人员共同探讨如何优化协作流程，提高系统响应速度，如在数据库查询环节增加缓存机制，减少重复查询数据库的操作。

1. ****功能优先级划分讨论（13min）****
   1. ****优先级确定依据****：基于 MoSCoW 法则，全体参会人员一致认为用户登录功能作为系统的入口，是必须实现的核心功能。业务专家补充强调，若登录功能不稳定或不可用，将直接影响用户对整个系统的使用和信任度，因此其优先级不可动摇。
2. ****用户旅程图探讨（17min）****
   1. ****旅程步骤细化****：对用户旅程图中的各个步骤进行深入讨论，明确在用户访问登录页面时，应提供清晰的账号密码输入提示和视觉引导。对于登录失败的情况，除了显示错误提示，还应考虑提供密码找回或账号申诉的快捷链接，方便用户解决问题。
   2. ****关键触点优化****：针对关键触点，如登录页面的输入框和提交按钮，讨论了如何从交互设计上提高用户操作的便捷性和准确性。例如，通过设置输入框的自动聚焦功能，减少用户手动点击输入框的操作；提交按钮在用户点击后，应及时给予反馈，如变色或显示加载动画，让用户知晓操作已被接收。
3. ****技术可行性分析（19min）****
   1. ****技术难点深入讨论****：开发团队详细阐述密码加密存储和防止暴力破解的技术难点。在密码加密方面，对比了多种加密算法的安全性和性能，最终确定采用 SHA - 256 算法，并结合盐值（Salt）技术进一步增强密码安全性。对于防止暴力破解，除了限制尝试次数外，还探讨了采用图形验证码、短信验证码等方式进行二次验证，提高系统安全性。
   2. ****解决方案评估****：针对提出的技术方案，进行可行性评估和成本效益分析。评估不同方案对系统性能、用户体验和开发成本的影响，确保选择的方案在满足安全需求的前提下，具有较高的可行性和性价比。
4. ****数据字典（用户实体）确认（10min）****

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

* 1. ****字段含义与类型明确****：对数据字典中用户实体的各个字段进行详细确认，明确每个字段的含义、数据类型和是否为主键。特别讨论了 user\_id 字段与校园统一身份认证系统对接的具体方式和数据格式要求，确保数据的准确性和一致性。
  2. ****数据存储与关联规划****：结合系统架构，规划用户实体数据的存储方式和与其他模块数据的关联关系。考虑到数据的扩展性和查询效率，确定采用关系型数据库存储用户信息，并建立合适的索引，提高数据查询速度。

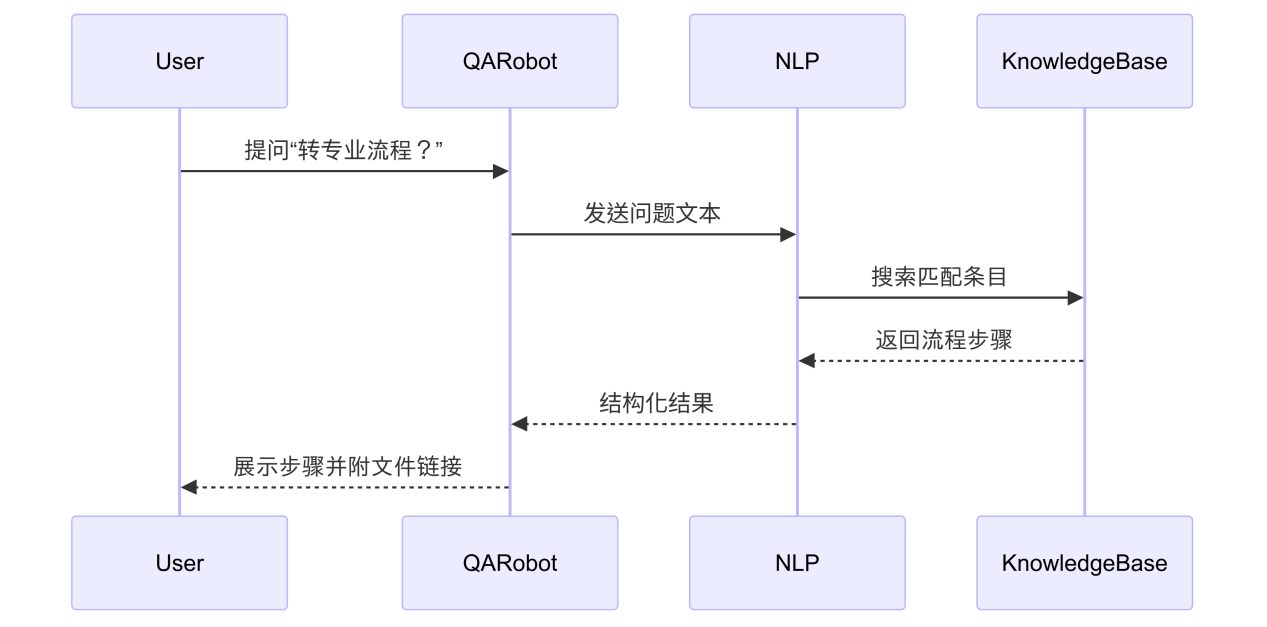
1. ****风险评估表（用户登录模块）分析（15min）****

| **风险类型** | **具体风险描述** | **严重程度** | **应对措施** |
| --- | --- | --- | --- |
| 安全风险 | 密码泄露导致系统被入侵 | 高 | 采用HTTPS加密传输，定期更换密码 |
| 性能风险 | 大量并发登录导致系统响应变慢 | 中 | 优化数据库查询，增加缓存层 |

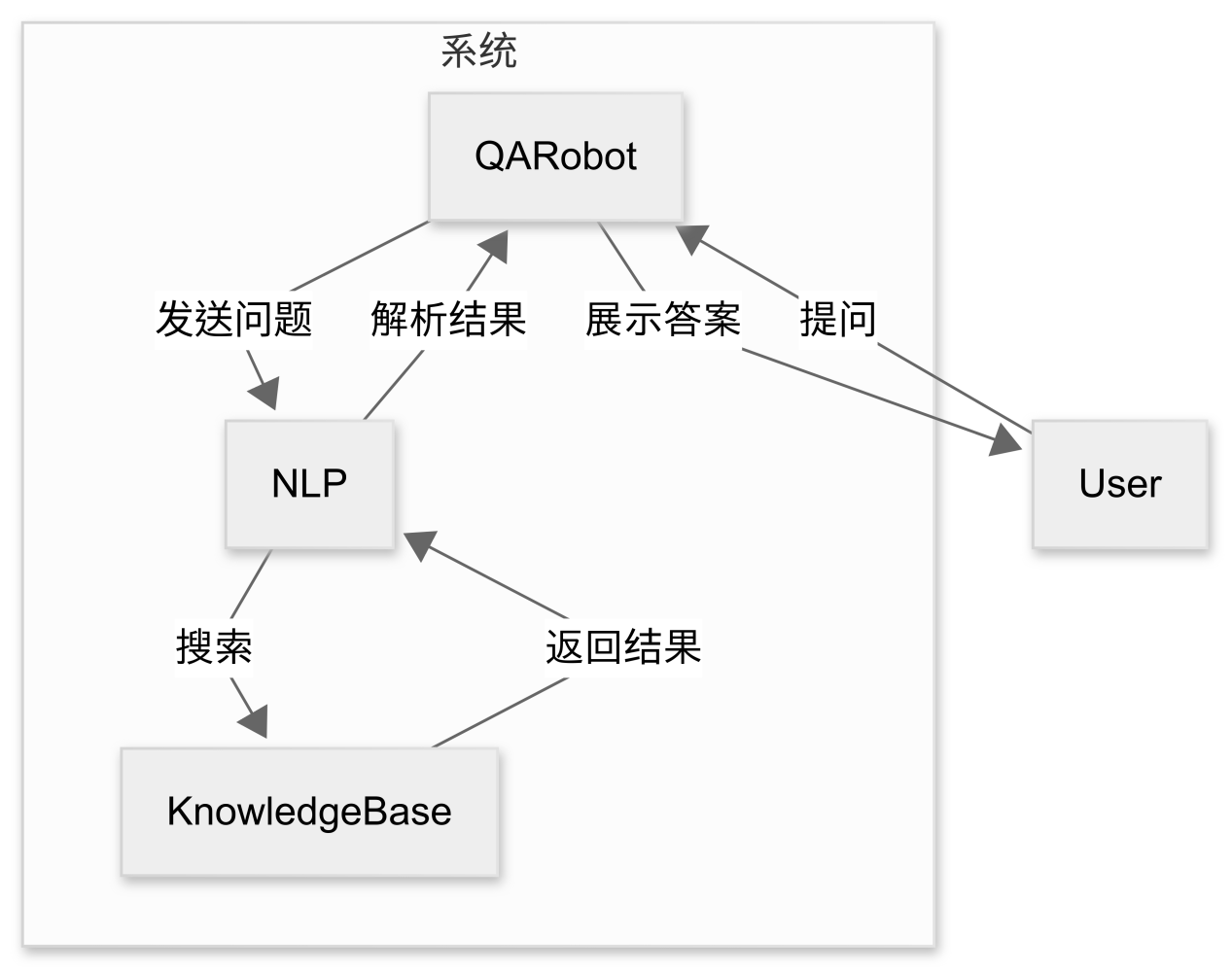
* 1. ****风险识别与评估****：全面分析安全风险和性能风险。对于密码泄露导致系统被入侵的安全风险，评估其可能造成的严重后果，如用户信息泄露、系统被恶意篡改等。对于大量并发登录导致系统响应变慢的性能风险，通过模拟高并发场景进行压力测试，预估系统可能出现的性能瓶颈和响应延迟情况。
  2. ****应对措施制定与讨论****：针对识别出的风险，制定详细的应对措施。在安全风险方面，除了采用 HTTPS 加密传输和定期更换密码外，还讨论了建立安全监控机制，实时监测密码登录异常行为，及时发现并处理潜在的安全威胁。在性能风险方面，确定优化数据库查询语句，采用缓存技术（如 Redis）存储常用用户信息，减少数据库查询压力，提高系统并发处理能力。

### **（三）问答模块讨论（20min）**

1. ****结构化需求详细解读（10min）****
   1. ****功能点深入剖析****：产品经理详细介绍问答模块的各个功能点，包括给出具体解决流程、文件规定依据、明确找哪个老师解决和多模态能力。对于给出具体解决流程功能，以转专业流程为例，详细说明系统如何通过 NLP 解析问题，在知识库中匹配流程条目，并以结构化方式返回给用户。在文件规定依据功能中，强调系统对文件名称、发布日期、链接等信息的准确匹配和展示。
   2. ****业务规则细化****：进一步细化业务规则，如在 NLP 解析问题时，讨论如何提高解析的准确性和灵活性。对于匹配失败的情况，除了提示用户重新提问或联系人工客服外，还考虑提供一些相似问题的推荐，引导用户更准确地表达问题。在多模态能力方面，明确语音输入和图片识别的具体技术要求和实现方式，如语音识别采用讯飞或 Google Speech 等成熟的 ASR 技术，图片 OCR 调用 Tesseract 或百度 AI 等工具。
2. ****Mermaid 顺序图与协作图分析（10min）****



* 1. ****顺序图讲解****：以 “给出具体解决流程” 为例，开发团队详细讲解顺序图中用户、问答机器人、NLP 和知识库之间的交互流程。分析在搜索匹配条目环节，如何优化算法提高匹配效率和准确性。例如，采用语义匹配技术，不仅匹配关键词，还考虑问题的语义相似度，提高匹配的精准度。



* 1. ****协作图分析****：通过协作图，展示系统各部分在问答过程中的协作关系。讨论如何加强各模块之间的协作，如在 NLP 解析问题后，如何将解析结果高效地传递给知识库进行匹配，以及匹配结果如何准确地返回给用户界面展示。

1. ****功能优先级划分讨论（10min）****

| **功能点** | **优先级** | **核心功能标记** | **理由** |
| --- | --- | --- | --- |
| 给出具体解决流程 | Must Have | ✅ | 核心问答功能 |
| 文件规定依据 | Must Have | ✅ | 必要的政策查询功能 |
| 明确找哪个老师解决 | Should Have |  | 高需求但可后续优化 |
| 多模态能力 | Could Have |  | 需要额外技术投入 |

* 1. ****优先级确定依据****：根据 MoSCoW 法则，结合业务需求和用户使用频率，确定 “给出具体解决流程” 和 “文件规定依据” 为 “Must Have” 功能。业务专家解释，这两个功能是用户使用问答机器人的核心需求，直接关系到用户获取信息的效率和准确性。“明确找哪个老师解决” 列为 “Should Have” 功能，虽然是高需求功能，但考虑到实现难度和开发资源，可以在后续阶段进行优化。“多模态能力” 列为 “Could Have” 功能，因为该功能需要额外的技术投入和资源整合，可根据项目进度和资源情况决定是否实现。

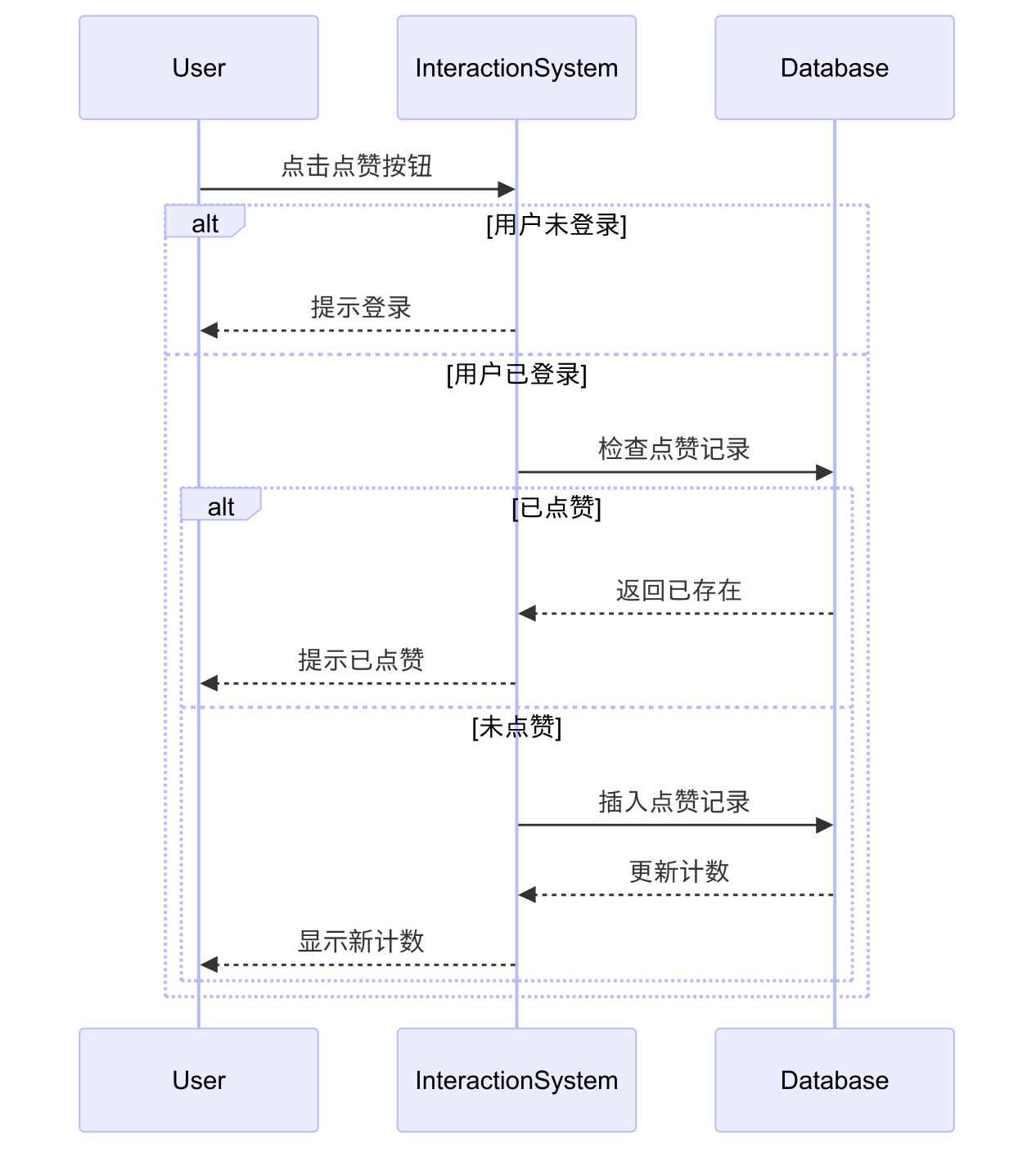
1. ****用户旅程图探讨（13min）****
   1. ****旅程步骤细化****：以 “给出具体解决流程” 为例，详细讨论用户旅程图中的各个步骤。在用户输入问题后，机器人显示 “正在思考…” 提示的时间间隔应合理设置，既不能让用户等待时间过长，也不能过早显示结果导致用户怀疑系统的准确性。对于系统返回的结构化步骤，讨论如何通过折叠 / 展开设计，提高页面的可读性和信息展示效率。
   2. ****关键触点优化****：针对关键触点，如语音输入按钮和上传图片按钮，讨论如何从设计上提高用户的操作意愿。例如，按钮的位置应方便用户点击，并且在用户点击后，给予明显的反馈提示，如按钮变色或播放提示音。对于流程步骤中的文件链接，应明确提示用户链接的内容和格式，避免用户误点击。
2. ****技术可行性分析（10min）****
   1. ****技术难点深入讨论****：开发团队详细阐述各个功能点的技术难点。在 NLP 模型方面，讨论如何提高模型对校园事务相关问题的理解和解析能力，如通过收集大量校园事务相关的语料进行训练，优化模型参数。对于知识库的结构化存储，确定采用 JSON 格式，并讨论如何建立高效的索引和查询机制，提高匹配效率。在文件链接更新维护方面，制定定期检查和更新的机制，确保链接的有效性。
   2. ****解决方案评估****：针对技术难点提出的解决方案进行评估，综合考虑技术实现难度、成本和性能等因素。例如，在选择语音识别和图片 OCR 技术时，对比不同供应商的产品性能、价格和接口兼容性，选择最适合项目需求的技术方案。

| **功能点** | **技术难点与解决方案** |
| --- | --- |
| 给出具体解决流程 | - NLP模型需准确解析流程类问题<br>- 知识库需结构化存储流程条目（如JSON格式） |
| 文件规定依据 | - 文件名称与问题的关联需建立映射表<br>- 文件链接需定期更新维护 |
| 明确找哪个老师解决 | - 部门-联系人关系需动态维护<br>- 支持一键拨号需集成通讯录API |
| 多模态能力 | - 语音识别需集成ASR（如讯飞/Google Speech）<br>- 图片OCR需调用Tesseract或百度AI |

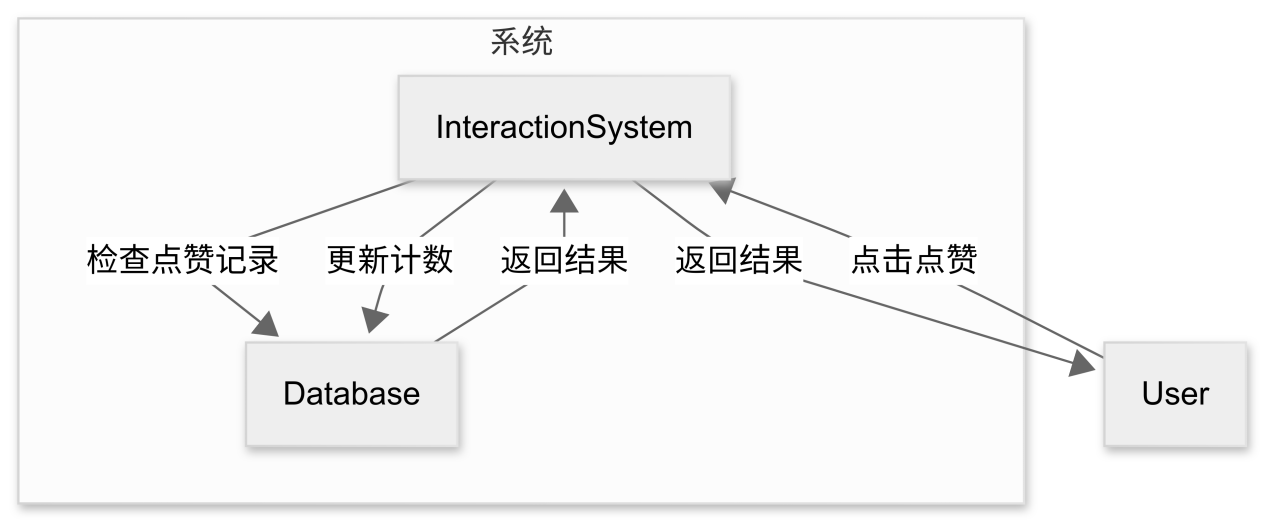
1. ****数据字典（问题实体）确认（15min）****
   1. ****字段含义与类型明确****：对数据字典中问题实体的各个字段进行详细确认，明确每个字段的含义、数据类型和是否为主键。特别讨论了 question\_id 字段采用 UUID 字符串的唯一性和安全性，以及 question\_content 字段的长度限制和存储方式，确保能够准确存储用户输入的问题。
   2. ****数据存储与关联规划****：结合系统架构，规划问题实体数据的存储方式和与其他模块数据的关联关系。确定将问题实体数据与回答内容、用户信息等相关数据进行关联存储，方便查询和统计分析。例如，通过 user\_id 字段关联用户实体，获取提问用户的相关信息，用于用户行为分析和个性化服务。
2. ****风险评估表（问答模块）分析（8min）****
   1. ****风险识别与评估****：全面分析技术风险、数据风险和性能风险。对于 NLP 模型误判问题类型导致答案错误的技术风险，通过对历史问题和答案进行抽样分析，评估错误率对用户体验的影响程度。对于知识库更新不及时导致信息过时的数据风险，统计知识库中信息的平均更新周期和过时信息的比例，评估其对用户获取准确信息的影响。对于高并发查询导致响应延迟的性能风险，通过模拟高并发场景进行压力测试，预估系统在不同并发量下的响应时间和吞吐量。
   2. ****应对措施制定与讨论****：针对识别出的风险，制定详细的应对措施。在技术风险方面，除了增加人工审核环节和定期优化模型外，还考虑建立用户反馈机制，及时收集用户对答案准确性的反馈，用于模型的优化和改进。在数据风险方面，建立知识库版本控制机制，记录每次更新的内容和时间，方便追溯和管理。同时，制定数据更新计划，确保知识库中的信息及时、准确。在性能风险方面，部署缓存层（如 Redis）存储热门问题和答案，减少重复查询数据库的操作，提高系统响应速度。同时，优化数据库查询语句和索引结构，提高数据库的并发处理能力。

### **（四）互动模块讨论（20min）**

1. ****结构化需求详细解读（10min）****
   1. ****功能点深入剖析****：产品经理详细介绍互动模块的各个功能点，包括回答点赞、评价反馈和吐槽功能。对于回答点赞功能，明确用户登录后才能点赞，且每个用户对同一回答只能点赞一次的规则。在评价反馈功能中，强调评价内容的评分范围、文本反馈要求以及敏感词过滤的重要性。对于吐槽功能，详细说明用户可匿名或实名提交，系统自动分类吐槽类型的实现方式。
   2. ****业务规则细化****：进一步细化业务规则，如在回答点赞功能中，讨论如何防止用户通过非法手段绕过登录限制进行点赞，以及如何在高并发点赞情况下确保点赞计数的准确性。在评价反馈功能中，明确敏感词过滤的具体实现方式，如采用第三方敏感词库进行匹配过滤，并定期更新敏感词库。在吐槽功能中，确定管理员回复的时间限制和回复格式要求，确保用户的反馈能够得到及时有效的处理。
2. ****Mermaid 顺序图与协作图分析（10min）****
   1. ****顺序图讲解****：以 “回答点赞” 为例，开发团队详细讲解顺序图中用户、互动系统和数据库之间的交互流程。分析在检查点赞记录环节，如何优化数据库查询效率，减少查询时间。例如，采用缓存技术存储点赞记录，减少对数据库的直接查询。同时，讨论在高并发场景下，如何通过分布式锁机制确保点赞操作的原子性，避免重复点赞。



* 1. ****协作图分析****：通过协作图，展示系统各部分在互动过程中的协作关系。讨论如何加强各模块之间的协作，如在用户提交点赞、评价或吐槽后，如何将数据及时准确地存储到数据库，并更新相关统计信息，如点赞计数、平均评分等。



1. ****功能优先级划分讨论（10min）****
   1. ****优先级确定依据****：根据 MoSCoW 法则，结合用户参与度和系统运营需求，确定 “回答点赞” 为 “Must Have” 功能。业务专家解释，点赞功能可以直观反映用户对回答的认可程度，有助于提高用户参与度和优质内容的传播。“评价反馈” 列为 “Should Have” 功能，因为该功能对于提升服务质量具有重要意义，但实现难度相对较高，需要一定的开发时间。“吐槽功能” 列为 “Could Have” 功能，虽然对系统改进有一定帮助，但考虑到管理员的处理工作量和开发资源，可根据项目实际情况决定是否实现。
2. ****用户旅程图探讨（10min）****
   1. ****旅程步骤细化****：以 “评价反馈” 为例，详细讨论用户旅程图中的各个步骤。在用户查看问答机器人的回答后，点击 “评价” 按钮进入评分页面的交互设计应简洁明了，评分星级选择器的视觉效果应突出，方便用户操作。对于用户填写反馈文本时，实时过滤敏感词的提示应及时准确，避免用户输入违规内容。
   2. ****关键触点优化****：针对关键触点，如评分星级选择器和文本输入框，讨论如何从设计上提高用户的操作体验。例如，评分星级选择器可以采用点击变色和悬停提示的方式，让用户更清楚自己的选择。文本输入框应提供实时字数统计和自动换行功能，提高用户输入的便利性。
3. ****技术可行性分析（20min）****
   1. ****技术难点深入讨论****：开发团队详细阐述各个功能点的技术难点。在回答点赞功能中，重点讨论高并发场景下数据库写入压力和分布式锁实现的技术细节。对于评价反馈功能，探讨实时敏感词过滤集成第三方 API 的接口调用方式和性能优化问题。在吐槽功能中，研究图片上传压缩处理的算法和自动分类 NLP 模型的选择和训练方法。
   2. ****解决方案评估****：针对技术难点提出的解决方案进行评估，综合考虑技术实现难度、成本和性能等因素。例如，在选择分布式锁实现方式时，对比 Redis 分布式锁和 Zookeeper 分布式锁的优缺点，根据项目的实际需求和架构特点选择合适的方案。在图片上传压缩处理方面，评估不同压缩算法对图片质量和文件大小的影响，选择最佳的压缩方案。
4. ****数据字典（评价实体）确认（10min）****
   1. ****字段含义与类型明确****：对数据字典中评价实体的各个字段进行详细确认，明确每个字段的含义、数据类型和是否为主键。特别讨论了 feedback\_id 字段采用 UUID 字符串的唯一性和安全性，以及 feedback\_content 字段的长度限制和存储方式，确保能够准确存储用户的评价反馈内容。
   2. ****数据存储与关联规划****：结合系统架构，规划评价实体数据的存储方式和与其他模块数据的关联关系。确定将评价实体数据与用户信息、问题信息、回答信息等相关数据进行关联存储，方便查询和统计分析。例如，通过 user\_id 字段关联用户实体，获取提交评价反馈的用户信息；通过 question\_id 字段关联问题实体，了解评价所针对的问题；通过 answer\_id 字段关联回答实体，分析回答的质量和用户满意度。
5. ****风险评估表（互动模块）分析（10min）****
   1. ****风险识别与评估****：全面分析安全风险、性能风险和用户体验风险。对于匿名吐槽内容包含恶意信息的安全风险，通过对历史吐槽数据的分析，评估恶意信息的比例和可能造成的危害。对于高并发点赞导致数据库崩溃的性能风险，通过模拟高并发场景进行压力测试，预估数据库在高并发情况下的负载能力和可能出现的故障情况。对于评价反馈提交后无即时提示的用户体验风险，通过用户调研和实际操作测试，评估用户对即时提示的需求程度和无提示情况下对用户体验的影响。
   2. ****应对措施制定与讨论****：针对识别出的风险，制定详细的应对措施。在安全风险方面，启用人工审核队列和设置关键词过滤，对匿名吐槽内容进行严格审核，确保内容合规。同时，建立风险预警机制，对恶意信息进行实时监测和处理。在性能

