

大模型智能问答客服系统系统设计文档

日期： 2026-03-31
状态： 已确认
技术栈： Python / FastAPI / Vue3 / DeepSeek / PostgreSQL / Redis

一、项目背景与目标

将企业内部的报销制度、软件操作手册、财务/科研流程、常见问题汇总等文档接入大模型，构建一套智能问答客服系统。用户可以用自然语言提问，系统给出答案并注明引用来源；支持上传截图辅助提问。

核心目标：

- 文档知识库化，降低重复人工咨询成本
- 答案可追溯，每条回答标注来源文档、章节、页码或段落
- 支持截图上传，辅助描述操作类问题
- 管理员可自助维护文档，无需开发介入

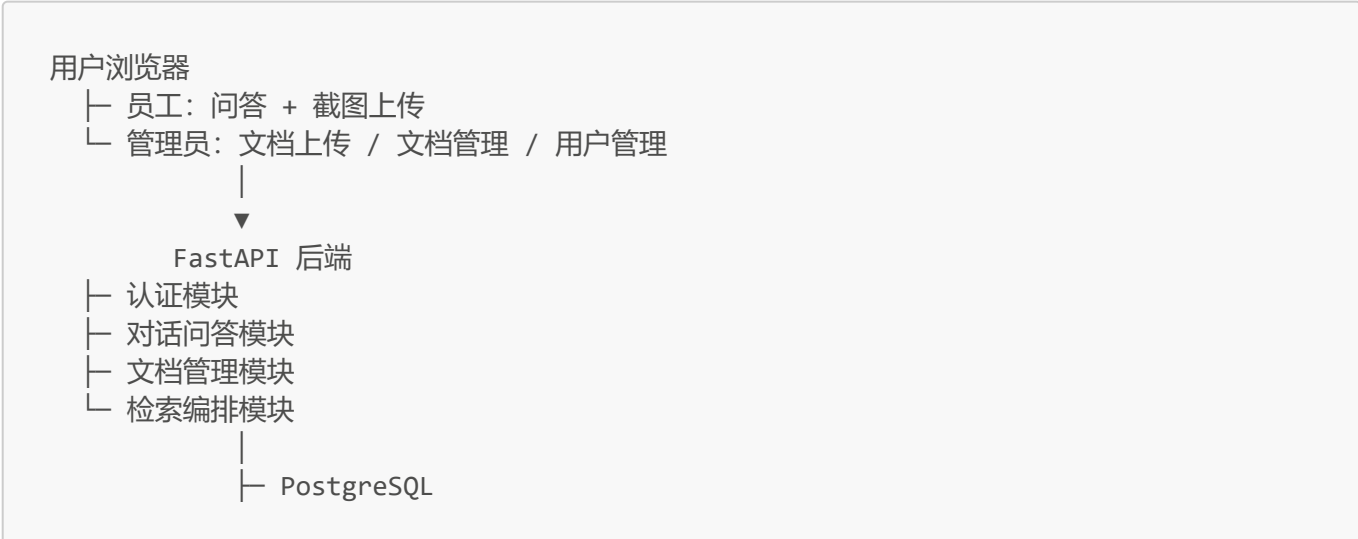
本期检索策略目标：

- 前期不引入向量数据库与 embedding
- 以“先选文档，再选章节，再定位段落”为主链路，优先保证准确率和可解释性
- 允许用大模型对候选文档和章节做重排，但不让大模型承担全库检索职责

二、用户角色

角色	权限
employee (员工/外部合作方)	对话问答、查看本人历史会话
admin (管理员)	文档上传/删除/管理、用户管理 (创建/角色分配/禁用)

三、整体架构





四、核心模块设计

4.1 认证模块

- **账号名：** 邮箱地址
- **注册管控：** 关闭完全开放的自助注册；支持两种方式创建账号：
 - 管理员后台手动创建（指定角色）
 - 邮箱自助注册（仅限白名单域名，如 `@company.com`，可配置），默认角色 `employee`
- **登录方式：**
 - 邮箱 + 密码
 - 邮箱 + 验证码（6位数字，5分钟有效，存 Redis）
- **验证码防刷：**
 - 同一邮箱 60 秒内不可重复发送
 - 同一 IP 每小时限制 10 次
- **Token 机制：**
 - Access Token（JWT）：有效期 2 小时
 - Refresh Token：有效期 7 天，存 Redis，可主动吊销
 - 管理员禁用账号时，将该账号现有 Token 加入 Redis 黑名单，立即失效
- **邮件服务：** QQ 邮箱 SMTP（`smtp.qq.com:465`，SSL，授权码鉴权）

4.2 文档管理模块

功能	说明
上传	支持 PDF、Word(.docx)、Excel(.xlsx)，单文件限制 50MB；服务端验证 Magic Bytes + 扩展名双重校验
原始文件存储	原始文件保存到本地文件系统，按文档 ID / 版本号分目录存放

功能	说明
解析	PDF 用 <code>pdfplumber</code> ，Word 用 <code>python-docx</code> ，Excel 按 Sheet 转为规范化文本或 Markdown 表格
结构化	解析后生成三级结构： <code>document</code> 、 <code>section</code> 、 <code>chunk</code> ；章节标题、层级、页码范围、段落顺序均保留
规范化文本落盘	每份文档额外保存解析后的文本产物，便于调试、离线检查和兜底检索
索引入库	文档标题、文件名、标签、章节标题、段落正文、来源位置全部写入 PostgreSQL
异步处理	Celery + Redis；上传后后台解析、结构化、索引化，前端显示状态
状态机	<code>pending -> processing -> active / failed</code>
删除	先将文档标记 <code>deleting</code> ，再删索引、删解析文本、删原始文件，任一步失败均可幂等重试
更新（版本切换）	新版本先入库为 <code>pending</code> ；索引就绪后原子切换为 <code>active</code> ，旧版本转 <code>superseded</code>

结构化原则：

- 文档级：文件名、别名、标签、摘要
- 章节级：目录项、一级/二级/三级标题、页码范围、Sheet 名称
- 段落级：正文、表格文本、所属章节、顺序号、来源位置

4.3 问答检索模块

本期不使用向量检索，主链路采用“规则召回 + 大模型重排 + 段落精定位”。

问答流程（SSE 流式输出）：

用户输入（文本 + 可选截图）

-> OCR 识别截图文字（阿里云 OCR）

-> 合并为最终 Query 文本

-> Query 归一化（术语、别名、关键词、数字、表单名）

-> 文档候选召回（文件名 + 标签 + 文档摘要 + FTS）

-> 章节候选召回（目录项/章节标题/小节标题 + FTS）

-> 大模型重排候选文档与章节

-> 在最终候选章节内做段落精定位

-> 取命中段落及相邻段扩展上下文

-> 组装 Prompt（系统提示 + 引用片段 + 历史 5 轮 + 用户问题）

-> 调用 DeepSeek Chat API（SSE 流式输出）

-> 返回答案 + 引用来源（文档名 + 章节标题 + 页码/段落号）

设计原则：

- 大模型只负责候选重排，不负责全库检索
- 先缩小范围，再精确定位，优先提升召回准确率

- 只在少量明确候选章节中做正文定位，避免把无关文档片段带进 Prompt
- 若未命中足够可信的文档/章节，明确回答“未在当前知识库中找到依据”

查询归一化：

- 术语同义词映射，如“报销/报账”“出差/差旅”“附件/上传材料”
- 抽取数字、金额、日期、制度名、表单名
- 去除无意义停用词，保留关键短语

候选控制：

- 文档候选最多 5 个
- 章节候选最多 8 到 12 个
- 大模型重排后仅保留 1 到 3 个章节
- 最终送入生成模型的正文片段控制在 6 到 12 段

Prompt 设计原则：

- 模型只能基于提供的引用片段回答
- 依据不足时必须明确说不知道或未找到
- 在答案末尾输出引用片段编号，系统映射成文档名、章节、页码或段落号
- 多轮对话只携带最近 5 轮历史

后期扩展点：

- 检索层抽象为 `RetrieverInterface`
- 重排层抽象为 `RerankerInterface`
- 后期可平滑增加 BM25 增强、RRF 融合、向量检索、Reranker 模型

4.4 数据模型（核心表）

```
users
  id, email, hashed_password, role, is_active, created_at

refresh_tokens
  id, user_id, token_hash, expires_at, revoked

token_blacklist
  jti, expires_at

documents
  id, name, file_path, normalized_text_path, status, version_num,
  supersedes_id, created_by, created_at, summary, tags_json

document_aliases
  id, document_id, alias

document_sections
  id, document_id, parent_section_id, title, level,
  page_start, page_end, section_order, title_tokens
```

```
doc_chunks
  id, document_id, section_id, chunk_index, text,
  source_location, page_no, paragraph_no, tsv

conversations
  id, user_id, title, created_at

messages
  id, conversation_id, role, content, sources_json, created_at
```

说明:

- documents.summary 用于文档候选粗筛和大模型重排输入
- document_sections 是本期准确率的关键索引层
- doc_chunks.tsv 用于 PostgreSQL 全文检索
- source_location 统一映射为“页码 / 段落号 / Sheet 名称 / 章节路径”

4.5 前端页面

页面	访问角色	核心功能
登录/注册页	所有人	双模式登录、邮箱注册（白名单域名）、验证码发送
对话页	employee / admin	左侧会话列表、右侧 SSE 流式对话、截图上传、引用来源展示
文档管理页	admin	文档列表、上传、删除、解析状态查看、版本切换状态
用户管理页	admin	用户列表、创建账号、角色分配、禁用（即时生效）

五、技术选型

层次	选型	说明
前端	Vue3 + Element Plus	成熟后台组件生态
后端	FastAPI + Python 3.11	异步支持较好，适合 AI 服务编排
关系数据库	PostgreSQL	存用户、会话、文档元数据、章节索引、段落索引；支持 FTS
缓存 / 队列	Redis	验证码、黑名单、限流、Celery Broker
文件存储	本地文件系统 (Docker Volume)	存原始文档与解析后的规范化文本
异步任务	Celery + Redis	文档解析、结构化、索引化后台执行
检索方式	PostgreSQL FTS + 规则打分	前期不引入向量库，优先准确率与可解释性
重排方式	DeepSeek Chat	对候选文档与章节做小范围重排
OCR	阿里云 OCR	截图文字识别

层次	选型	说明
生成模型	DeepSeek Chat	基于引用片段生成答案
流式输出	SSE (Server-Sent Events)	FastAPI 原生支持，前端 EventSource 接收
邮件	QQ 邮箱 SMTP (465/SSL)	发送验证码，额外成本低
容器化	Docker + Docker Compose	开发与生产环境统一
敏感配置	.env + Docker Compose env_file	API Key、SMTP 授权码、OCR Key 等全部环境变量注入

六、云端部署架构



推荐初期服务器配置：

- 应用服务器：4核8G ECS/CVM（运行 FastAPI + Celery + Nginx）
- 数据库：RDS PostgreSQL 2核4G
- Redis：云托管版 1G

七、人日估算

模块	工作内容	人日
项目初始化	目录结构、Docker Compose、开发环境、 .env 规范	2
认证模块	邮箱注册/登录、验证码、防刷、JWT、Refresh Token、角色权限	6
文档管理后端	上传、校验、解析、结构化、章节索引、段落索引、状态机、补偿删除、版本切换	8

模块	工作内容	人日
问答检索后端	Query 归一化、文档召回、章节召回、LLM 重排、段落精定位、Prompt 组装、SSE 输出	7
OCR 集成	OCR 接入、与 Query 合并	2
前端：登录/注册	双模式登录、注册、验证码交互	2
前端：对话页	会话列表、SSE 流式渲染、截图上传、引用来源展示	7
前端：文档管理页	上传、列表、处理状态轮询、删除	3
前端：用户管理页	用户列表、创建、角色分配、禁用	2
集成测试 & 部署	云端部署、域名、SSL、容器网络、联调	5
Buffer	调试、需求微调、第三方 API 联调	6
合计		50

工期参考：

- 1 人独立开发：约 9 到 10 周
- 前后端 2 人并行：约 5 到 6 周可上线基础版本
- 后期升级到混合检索：额外约 5 到 8 人日

八、后期演进路线

1. **检索增强**：从 PostgreSQL FTS 升级到 BM25 / 更细粒度规则打分
2. **混合检索**：在现有 `RetrieverInterface` 基础上增加向量检索，与词法检索融合
3. **重排增强**：用专门的 reranker 模型替代通用聊天模型做候选重排
4. **文件存储迁移**：本地文件系统迁移到阿里云 OSS
5. **SSO 集成**：对接企业 LDAP / 统一认证
6. **用量统计**：统计问答次数、热门问题、未命中问题、低置信问题