# 智能客服系统大作业报告

目录

[智能客服系统大作业报告 1](#_Toc26824)

[一、项目背景与意义 1](#_Toc19430)

[二、需求分析 1](#_Toc2919)

[三、系统设计 1](#_Toc12612)

[3.1 总体架构 1](#_Toc30143)

[3.2 主要模块说明 2](#_Toc7302)

[四、详细实现 2](#_Toc18492)

[4.1 系统界面与交互（黄政博3220411019） 2](#_Toc12013)

[4.2 知识库设计与问答逻辑（王俊杰3220411020） 2](#_Toc8081)

[4.3 API集成与系统测试（仇天翔3220411048） 2](#_Toc30958)

[五、系统运行与测试 3](#_Toc24235)

[六、项目分工与总结（黄政博3220411019） 3](#_Toc15590)

[七、参考文献 3](#_Toc1294)

## 一、项目背景与意义

随着电子商务的快速发展，客户服务需求日益增长。传统人工客服存在响应慢、成本高等问题，智能客服系统应运而生。通过自然语言处理和知识库问答，智能客服能够高效解答用户常见问题，提升用户体验，降低企业运营成本。本项目以淘宝电商场景为背景，设计并实现了一个基于知识库和大模型API的智能客服系统，支持多种文件格式的知识导入，具备实时对话、常见问题快捷回复等功能。

## 二、需求分析

1. 支持多种格式的知识库文件（TXT、PDF、DOCX、CSV、JSON）。

2. 提供简洁友好的对话界面，支持实时问答。

3. 能够根据知识库内容优先回答用户问题，未命中时调用大模型API智能回复。

4. 支持常见问题快捷提问、对话历史记录、清除对话等功能。

5. 系统易于扩展和维护，便于后续知识库更新。

## 三、系统设计

### 3.1 总体架构

系统采用模块化设计，主要包括界面交互层、知识库管理层、问答逻辑层和API集成层。整体架构如下：

界面交互层：基于Tkinter实现，负责用户输入输出、快捷按钮、对话展示等。

知识库管理层：负责知识库文件的加载、解析和结构化存储。

问答逻辑层：实现用户问题与知识库内容的匹配、模糊检索、答案生成等。

API集成层：对接Deepseek大模型API，补充知识库未覆盖的问题。

### 3.2 主要模块说明

`app.py`：主程序，包含界面、消息处理、知识检索、API调用等核心逻辑。

`knowledge\_base.json`：知识库文件，结构化存储常见问题、产品信息、服务指南等。

`format\_json.py`：知识库格式化工具，保证JSON文件规范。

`requirements.txt`：依赖包说明，便于环境搭建。

## 四、详细实现

### 4.1 系统界面与交互（黄政博3220411019）

- 使用Tkinter库设计主窗口，包含标题栏、对话区、输入区、快捷问题按钮等。

- 支持用户输入问题、点击快捷问题按钮自动填充并发送。

- 对话区采用滚动文本框，自动显示历史消息，区分用户与客服角色。

- 提供“发送”“清除对话”“退出”等功能按钮，提升用户体验。

- 绑定回车键快速发送消息，窗口关闭前弹窗确认。

### 4.2 知识库设计与问答逻辑（王俊杰3220411020）

- 设计`knowledge\_base.json`，包含“常见问题”“产品信息”“服务指南”三大板块，结构清晰，便于扩展。

- 实现知识库加载、解析，支持多种问题类型的检索。

- 采用`difflib.SequenceMatcher`实现用户问题与知识库内容的模糊匹配，提升问答准确率。

- 优先在知识库中查找答案，支持常见问题、产品信息、服务指南多层次检索。

- 未命中时返回None，交由API层处理。

### 4.3 API集成与系统测试（仇天翔3220411048）

- 集成Deepseek大模型API，配置API密钥与基础URL，保证安全调用。

- 未命中知识库时，将用户问题与知识库摘要作为上下文发送给API，获取智能回复。

- 负责依赖环境搭建（见`requirements.txt`），如openai、langchain、PyQt5等。

- 设计并执行系统测试用例，覆盖常见问题、产品检索、服务指南、API兜底等场景，保证系统稳定性和准确性。

- 优化系统异常处理，提升用户体验。

## 五、系统运行与测试

1. 安装Python 3.8及以上版本，执行`pip install -r requirements.txt`安装依赖。

2. 运行`app.py`，启动智能客服系统。

3. 在界面左侧上传知识库文件，或直接使用默认`knowledge\_base.json`。

4. 输入问题或点击快捷问题按钮，系统自动检索知识库并回复。

5. 若知识库无答案，系统自动调用大模型API智能回复。

6. 测试覆盖常见问题、产品信息、服务指南、异常输入等，系统均能正确响应。

## 六、项目分工与总结（仇天翔3220411048）

**1. 你们小组的这个项目，具有哪些功能模块？**

项目主要功能模块有：

图形化界面与交互模块；

知识库管理与检索模块；

智能问答API集成模块；

对话历史与快捷操作模块；

系统测试与异常处理模块。

1. **你参与了其中哪些功能模块？是通过什么方式实现的？**

我主要负责API集成与系统测试。具体包括：

集成Deepseek大模型API，配置API密钥和基础URL，保证安全高效调用；

实现知识库未命中时，将用户问题与知识库摘要作为上下文发送给API，获取智能回复；

负责依赖环境的配置与管理，编写requirements.txt，确保项目可复现；

设计并执行系统测试用例，覆盖常见问题、产品检索、服务指南、API兜底等场景；

优化系统异常处理逻辑，提升系统健壮性和用户体验。

采用自动化测试和日志记录，确保系统稳定可靠。

1. **你对这个项目未来有什么改进想法？**

未来我计划：

引入多模型融合机制，根据问题类型智能选择最佳API或本地模型；

增加API调用的容错与降级策略，提升系统可用性；

优化依赖管理和自动化部署流程，支持云端一键部署；

丰富测试用例，加入压力测试和用户行为模拟，持续提升系统质量。

## 七、参考文献

[1] LangChain官方文档：https://python.langchain.com/

[2] Deepseek API文档：https://platform.deepseek.com/docs

[3] Python官方文档：https://docs.python.org/3/

[4] Tkinter官方教程：https://docs.python.org/3/library/tkinter.html

[5] 相关高校智能客服系统课程资料