**《宠物托运智能服务系统》**

**详细设计说明书：智能客服模块**

**课 程： 软件工程实训**

**队 伍： 深藏Blue**

**项目名称： GoGoPet**

**指导老师： 石秀金**

**成 员： 皮 超 221310417**

**赵鑫奕 221310109**

**张钰龙 221310131**

**李佳慧 221310203**

**丛梓璇 221310201**

**版本号V1.0**

**2025年4月2日**

1 引言

1.1 编写目的

本详细设计说明书用于指导**智能客服模块**的开发与实现，确保开发人员能够按照规范进行编码，同时为测试人员提供参考依据。本文档适用于开发人员、测试人员及相关项目成员。

1.2 项目概述

智能客服模块是**宠物托运智能服务系统**的重要组成部分，旨在为用户提供高效、智能的问答支持，帮助用户快速获取托运相关信息。本模块结合 LLM（大语言模型）与 RAG（检索增强生成）技术，支持自然语言交互，使用户能够通过对话方式获得精准答案。其核心功能包括实时问题解答、知识库支持、个性化推荐、多轮对话能力、突发状况应对以及多端支持。通过提供智能化的客服体验，该模块不仅提升了用户的满意度和信任度，还有效降低了人工客服的工作负担，推动宠物托运服务向更加智能、高效的方向发展。

1.3 术语定义

表 1 术语表

|  |  |
| --- | --- |
| **术语/缩写** | **定义** |
| **用户** | 使用系统的注册用户。 |
| **LLM**  **（Large Language Model）** | 大语言模型，基于深度学习的自然语言处理技术，能够理解和生成类似人类的文本。 |
| **RAG（Retrieval-Augmented Generation）** | 检索增强生成技术，通过结合知识库检索和语言模型生成，提高回答的准确性和专业性。 |
| 知识库 | 由管理员上传的行业相关资料（如托运政策、公司内部规章等），用于支持智能客服回答用户问题。 |
| FAQ（Frequently Asked Questions） | 常见问题库，系统预设的高频问题及其标准答案，便于快速解答用户常见疑问。 |
| 智能客服模块 | 本系统中提供问答支持的模块，帮助用户快速获取托运相关信息。 |

1.4 参考资料

* 《需求规格说明书》
* 《概要设计说明书》
* 《数据库设计说明书》

2 系统概述

2.1 系统架构

**智能客服模块**作为**宠物托运智能服务系统**的子模块，前端采用 **Vue 3 + Vite + Ant Design Vue** 进行界面开发，后端基于 **Spring Boot + MyBatis Plus + MySQL** 进行数据存储和接口提供。数据通过 **RESTful API** 进行交互，并结合 **LLM** 与 **RAG** 技术，实现智能问答功能。采用 **JWT** 进行身份验证，在确保用户数据安全的同时，提供更具有针对性的智能客服体验。

本模块依赖于用户管理模块，用户必须登录后才能询问客服，确保对话的安全性。系统架构如图1所示。

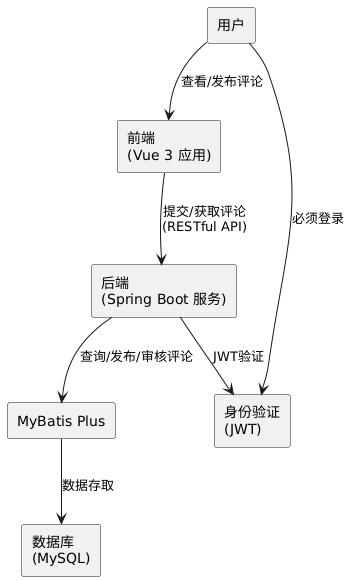


图 1 智能客服模块架构图

系统包含以下主要组件：

1. **用户（前端）**：用户通过浏览器访问系统，可以查看评论、发布评论。
2. **前端（Vue 3 应用）**：向后端发送API请求，获取或提交评论数据，并进行页面渲染。
3. **后端（Spring Boot 服务）**：处理前端请求，包括用户身份验证、评论管理（查询、发布、审核等）。
4. **数据库（MySQL）**：存储对话数据，包括对话ID、用户ID、对话内容等。
5. **身份验证（JWT 认证）**：确保用户必须登录后才能询问客服，防止匿名评价。

分层结构的架构关系如下：

* 前端层（浏览器、Vue 3）
* 业务逻辑层（Spring Boot API）
* 数据存储层（MySQL 数据库）

前端与后端通过RESTful API交互，后端与数据库通过MyBatis Plus进行数据存储和查询。

2.2 模块功能描述

**智能客服模块**主要提供以下核心功能：

1. **实时问答功能**

* 智能客服将解答用户在宠物托运过程中遇到的各类问题，提升用户体验，减少人工客服的工作负担。

1. **知识库管理功能**

* 管理员可上传托运相关政策、常见问题等文档，系统自动解析并纳入知识库，持续优化回答内容的专业性。

3 程序详细设计描述

3.1 智能客服模块

智能客服模块主要为用户提供基于自然语言的问答服务，包含以下功能点：

1. **实时问答**：用户可在系统内输入任意托运相关问题，系统基于大语言模型（LLM）与知识库检索（RAG）技术，实时生成专业、准确的回答。
2. **多轮对话**：支持上下文理解能力，能够处理连续提问的场景，为用户提供连贯、深入的交流体验。
3. **知识库管理**：管理员可上传托运相关政策、常见问题等文档，系统自动解析并纳入知识库，持续优化回答内容的专业性。
4. **个性化推荐**：系统根据用户问题智能识别需求，提供定制化建议，如托运准备清单、注意事项等。
5. **突发情况应对**：针对如航班延误、宠物突发健康问题等特殊情境，智能客服可提供应急处理建议及联系方式指引。

模块信息如表2所示。

表 2 模块信息表ChattList

|  |  |  |  |
| --- | --- | --- | --- |
| ***模块编号*** | **M\_ZNKF\_1** | **源程序文件** | **ChatBot.vue** |
| *功能* | **实时问答**：用户通过自然语言输入托运相关问题，系统基于 LLM 与 RAG 技术生成精准回答，并支持上下文理解，实现多轮对话体验 | | |
| *输入参数* | 用户 ID（pet\_id）：标识发起对话的用户  用户问题（user\_question）：用户输入的自然语言问题  历史上下文（context）：用户与系统的对话历史，用于多轮问答支持 | | |
| *要访问的表* | t\_chat（存储会话记录）  t\_user（获取用户基本信息）  知识库接口或向量检索系统（用于 RAG 技术的检索支撑） | | |

表3展示了时序图相关内容。

表 3 模块信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **模块编号** | **M\_ZNKF\_1** | **源程序文件** | **ChatBot.vue** |
| **模块时序图** | 46  图 2 智能客服模块时序图  时序图的功能描述：  **智能问答：**  用户输入托运相关问题，系统基于大语言模型（LLM）与检索增强生成技术（RAG），从知识库中获取信息，生成准确答案。系统保存完整对话，并支持多轮交互，提供连续一致的问答体验。 | | |
| **输入界面** | 图 3 输入界面效果图  主要控件说明：   * **问题输入框**：一个多行文本框，供用户以自然语言输入问题内容，支持自动换行，支持上下文联想提示。 * **发送按钮**：一个用于提交问题的按钮，点击后系统启动问答流程，按钮文字为“发送”。 * **清空上下文按钮**：提供清空当前多轮对话上下文的功能，适用于重新开始提问。 * **对话记录区**：在输入框上方展示历史提问与系统回答内容，便于多轮对话追溯。 * **验证码框**：用于防止恶意刷问，可在首次提问或频繁提问时弹出，增强系统安全性。 * **快速提问推荐**：在输入框上方或初次进入页面时显示常见问题推荐标签（如“托运流程”“费用标准”等），便于用户快速点击发问。 | | |
| **输出界面** | 图 4 输出界面效果图  主要控件说明：   * 对话展示区：以对话气泡形式展示用户提问与系统回答，每轮交互保留时间戳与标识。 * 智能回答内容：系统回答支持富文本展示，包含文本、列表、链接、推荐卡片等格式，提升可读性。 * 回答反馈按钮：在每条系统回答下方展示“满意 / 不满意”按钮，便于收集用户反馈。 * 上下文展示切换按钮：允许用户查看/隐藏上下文内容，便于聚焦当前问题或查看多轮逻辑。 * 滚动加载：当对话内容较多时，支持滚动加载展示，提升浏览效率。 | | |
| **出错处理** | 统一跳转至error.vue页面 | | |

3.2 表结构说明

本模块设计的数据库表主要为t\_chat（存储对话信息）、t\_user（获取评论用户信息），具体信息如下：

表 4 t\_chat表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 字段名 | 数据类型 | 允许空值 | 说明 |
| 对话ID | chat\_id | VARCHAR(18) | N | 主键，唯一标识符 |
| 用户ID | pet\_id | VARCHAR(18) | N | 外键，指明该会话主体为哪个客户 |
| 对话内容 | chat\_content | LONGTEXT | Y | 储存会话 |

表 5 t\_user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 字段名 | 数据类型 | 允许空值 | 说明 |
| 用户ID | user\_id | VARCHAR(18) | N | 主键，唯一标识符 |
| 用户名 | user\_name | VARCHAR(63) | N | 注册信息 |
| 密码 | password | VARCHAR(63) | N | 注册信息 |
| 用户身份 | user\_type | CHAR(1) | N | 类型（用户/托运司机/管理员/……） |
| 用户地址 | user\_address | VARCHAR(63) | Y | 用户地址，限于用户、司机填写 |
| 用户头像 | picture\_id | VARCHAR(18) | Y | 外键，指明用户头像 |
| 公司ID | company\_id | VARCHAR(18) | Y | 外键，如果身份为司机则指明公司 |

4 公用接口

4.1 全局变量

在此模块中，定义了一些全局变量以便于各个页面和功能之间的共享和协调。以下是定义的一些常用全局变量：

1. **currentUserId**: 存储当前登录用户的ID，供各个功能点检查和访问用户数据。
2. **currentChatSessionId**: 存储当前进行中的聊天会话，便于后端保存/更新对话数据。
3. **chatContext**:存储当前会话中的全部上下文信息，用于向 LLM 提供多轮对话支持。
4. **feedbackStatusMap**: 存储用户对每条回答的反馈状态（满意/不满意/未反馈）。

4.2 公用界面

公用界面是指在多个模块或页面中重复使用的界面组件，帮助提高开发效率并保持界面一致性。以下是几个公用界面设计：

1. **登录界面：**

* 用户输入用户名和密码进行登录。
* 提供“忘记密码”和“注册”功能。
* 验证用户身份，成功后进入系统，初始化用户 ID 与会话历史。

1. **消息提示框：**

* 用于展示系统的提示信息（如“发送成功”、“存在敏感词”、“连接超时”）。
* 提供不同类型的提示（成功、错误、警告等）。

1. **聊天窗口组件：**

* 聊天消息按时间自动排序。
* 可区分用户提问和智能客服回复的气泡式消息框。
* 自动滚动到底部，提升用户体验。

1. **多轮上下文面板：**

* 显示用户与客服之间的上下文问答记录。
* 支持查看最近若干轮交互，便于用户回顾。
* 支持上下文清空重启。

1. **推荐问题区块：**

* 展示常见或推荐问题，如“如何准备宠物证件？”
* 点击后直接填入输入框并自动发送。
* 支持关键词动态联想。

4.3 公用函数和过程

公用函数和过程是为提高代码复用性，所有模块可调用的函数和过程。以下是部分公用函数和过程示例：

1. **validateChatInput：**

* 功能：验证用户对话信息是否为空、字符长度是否合法。
* 参数：问题文本 questionText: string。
* 返回值：布尔值，若输入合法返回true，否则返回false。

1. **submitChat：**

* 功能：将用户对话信息提交至后端进行智能问答处理。
* 参数：用户ID、对话ID、对话内容。
* 返回值：智能客服返回的答案。

1. **getChatHistory：**

* 功能：获取当前用户的历史问答记录。
* 参数：用户ID。
* 返回值：问答历史数组 ChatMessage[]。

1. **parseAnswerFromLLM：**

* 功能：解析从 LLM 模型返回的答案内容，可能包含建议、列表、FAQ等格式。
* 参数：responseText: string。
* 返回值：结构化显示的 HTML 或组件渲染数据。

1. **handleEmergencyCases：**

* 功能：根据提问识别是否属于突发事件（如“宠物晕机了怎么办”），并推荐应急方案。
* 参数：问题文本。
* 返回值：应急响应信息、联系方式等。

1. **logUserFeedback：**

* 功能：记录用户对回答的满意度反馈（如点赞、踩、文本建议等）。
* 参数：对话ID、反馈类型。
* 返回值：布尔值，若反馈成功返回true，否则返回false。

4.4 公用表辞典

公用表辞典是系统中统一使用的数据字典表，用于维护系统中常见的状态、分类等信息。以下是部分公用表辞典的设计：

1. **对话状态（t\_chat\_status）**

描述：对话的状态，如“已完成”，“待回复”等。

字段：

* status\_id：状态ID。
* status\_name：状态名称（如“已完成”、“等待回复”）。
* status\_description：状态描述（如“AI已生成回答”）。

1. **问题分类字典（t\_question\_category）**

描述：提供常见问题类型，用于分类问题。

字段：

* category\_id：分类 ID。
* category\_name：分类名称（如“证件办理”、“托运准备”）。
* category\_description：用于推荐问题分类。

1. **模型反馈类型（t\_feedback\_type）**

描述：定义反馈的类型，反馈类型分为满意、不满意、无响应等。

字段：

* type\_id：反馈类型 ID。
* type\_name：类型名称（满意、不满意、无响应等）。

这些表辞典能够确保系统的一致性和数据规范化。