OCP在线咨询平台

软件架构文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 21/4/2024 | 1.0 | 小组合作编写 | 齐佳怡、孙恬然、骆镒妤、朱涵、赵楷越 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

软件架构文档

# 简介

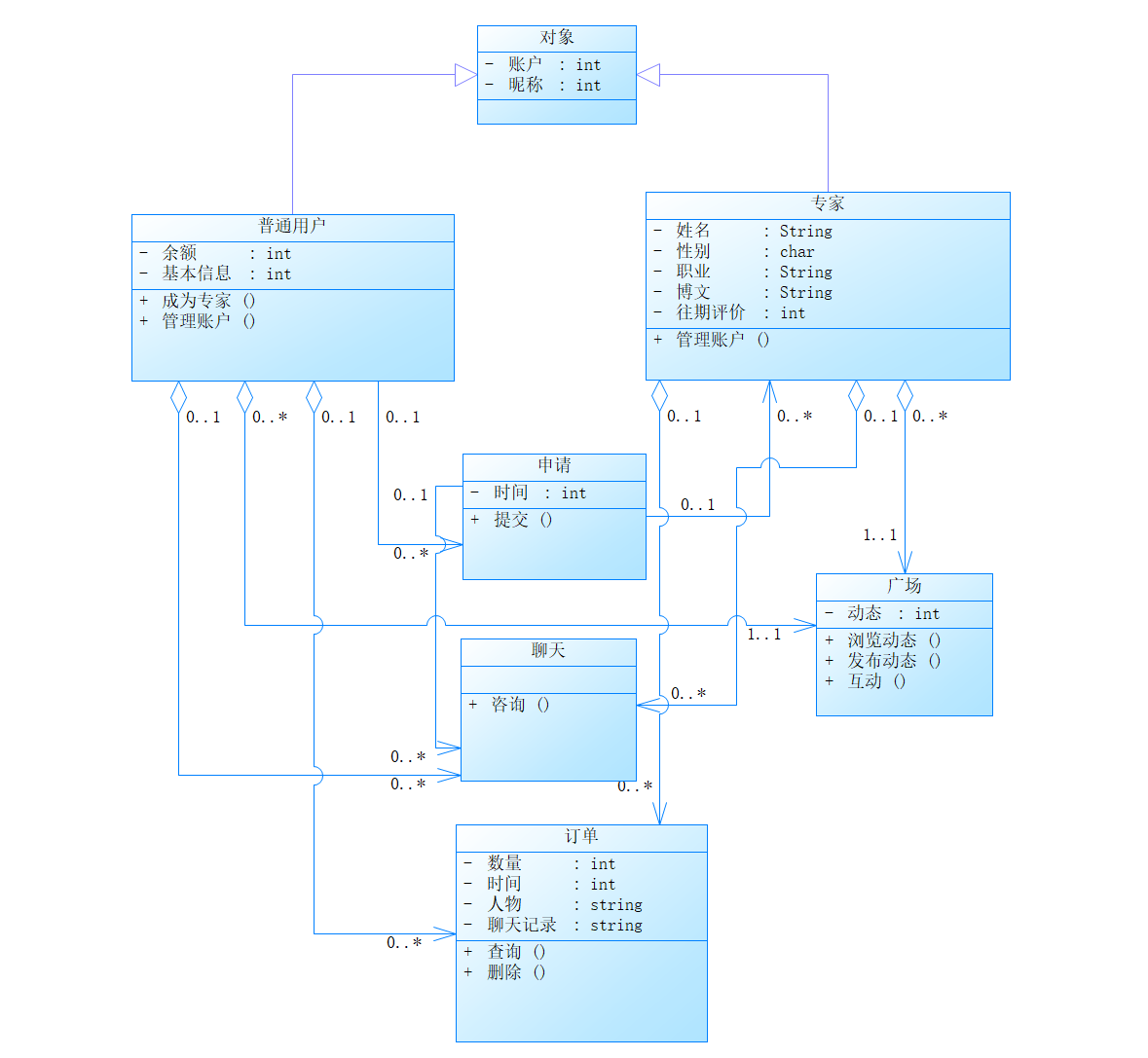
## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

## 范围

本软件架构文档提供了多领域一对一在线咨询平台的结构概述。一对一在线咨询系统，用于用户与特定领域的专家进行一对一在线咨询，获取专业帮助。

# 概念视图



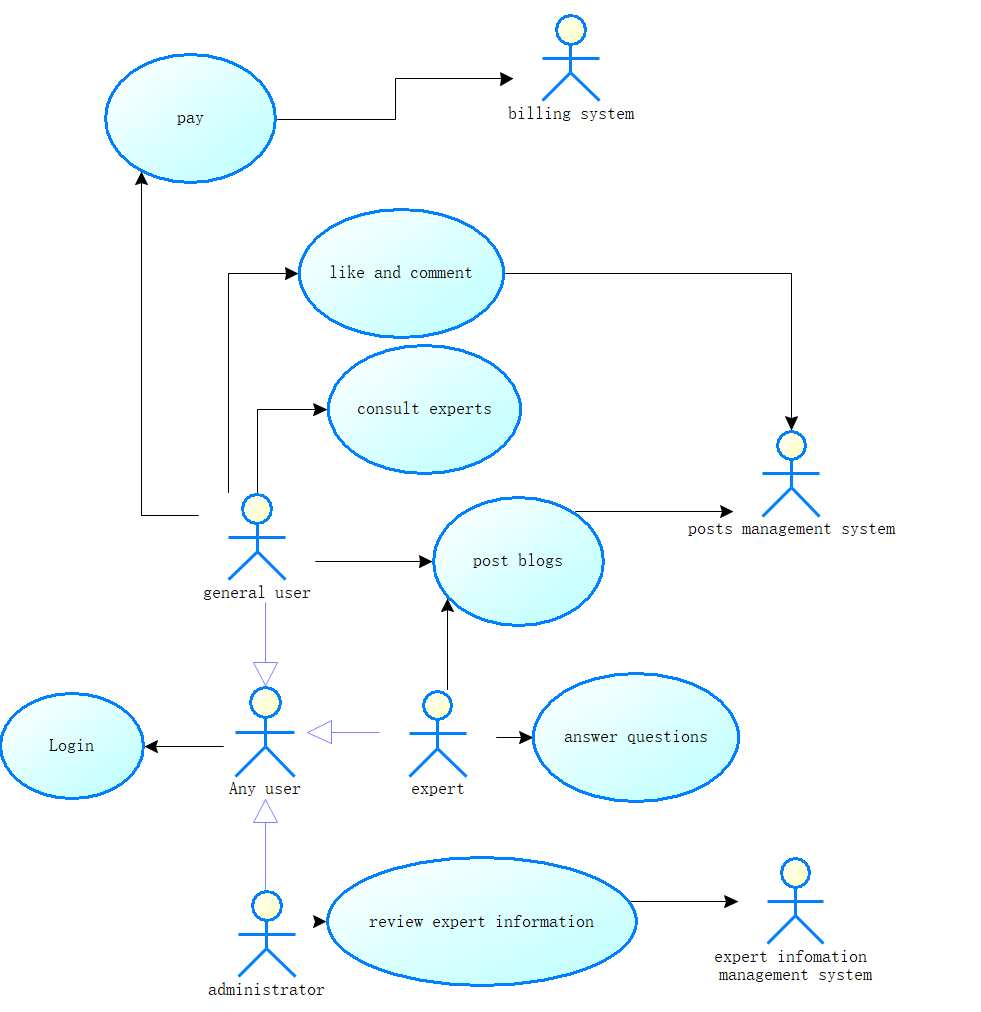
上图为线上咨询平台的概念模型。

用户类的基本属性包括账户和昵称。普通用户类和专家用户类继承自用户类。其中普通用户类的属性额外包括余额和基本信息，方法包括成为专家和管理账户；专家用户类的属性包括额外的专家信息，方法包括管理账户。普通用户通过申请可转换为开通专家账户。

普通用户和专家用户类与申请、聊天、广场和订单类都是聚合关系。申请类的属性包括时间，方法包括提交；聊天类的方法包括咨询；广场类的属性包括动态，方法包括浏览动态，发布动态以及互动；订单类的属性包括数量，时间，人物以及聊天记录，方法包括查询以及删除。

以上为对概念视图的阐释。

# 用例视图



线上咨询平台的用例包括：用户注册登录，用户维护个人主页，普通用户注册成为专家用户，平台广场，专家选择，专家排行，专家咨询，查看咨询历史。这些用例由普通用户，专家用户或是后台管理人员执行。另外，外部的支付系统以及信息验证系统也参与其中。

## 专家选择

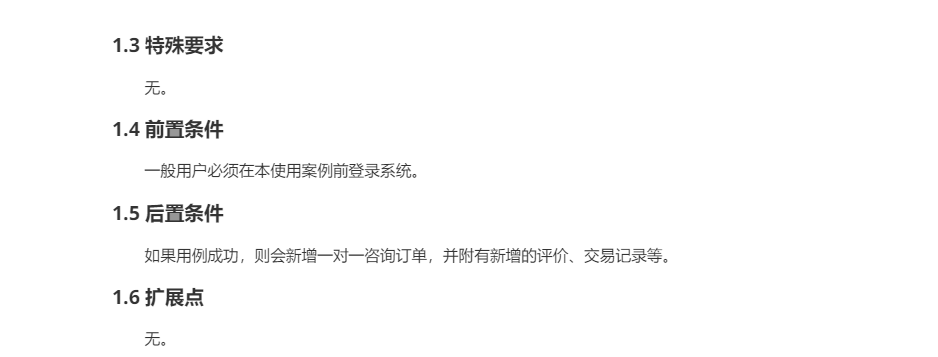
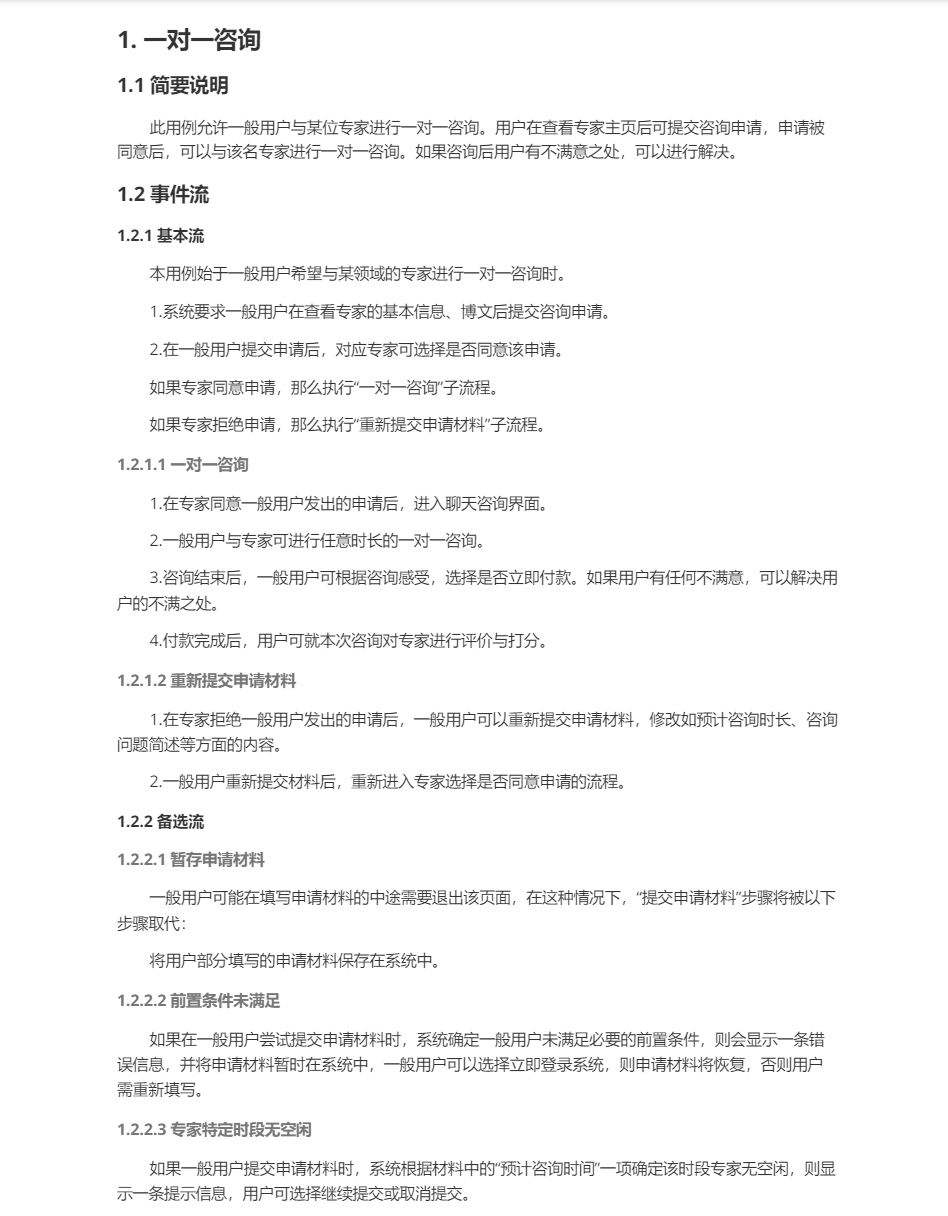
此用例允许普通用户在平台上选择合适的专家进行后续的咨询。用户可在此页面上进行搜索，可以根据目标领域进行筛选，依照不同排序方式进行排序。此用例的执行者是普通用户。

## 专家咨询

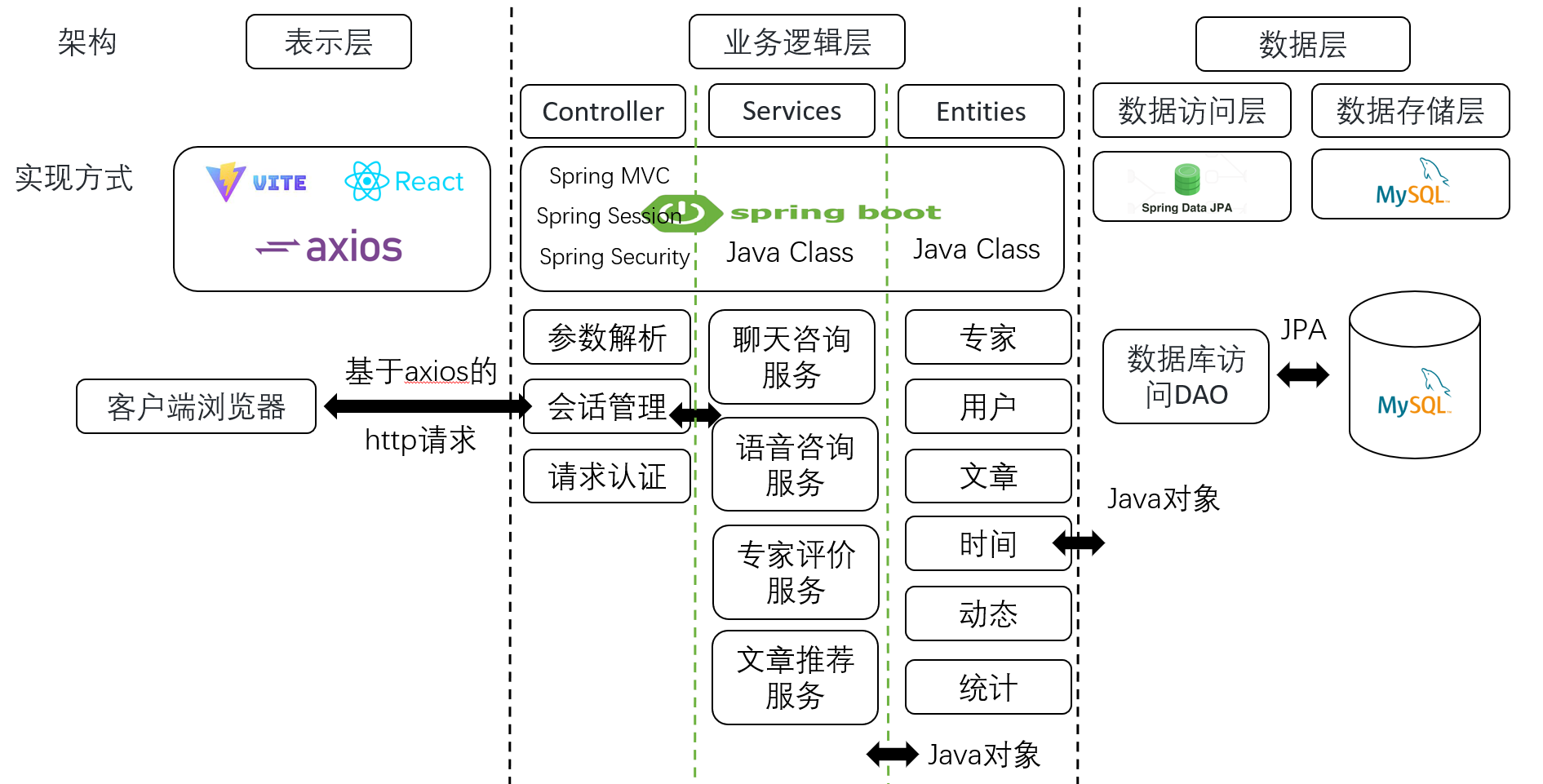
此用例允许普通用户对专家进行咨询。普通用户可以点击咨询历史，界面显示对应历史的专家评价以及相应的咨询记录。用户可以选择文字咨询或者语音咨询的形式。此用例的执行者是普通用户以及专家用户。

## 专家排行

此用例允许平台对专家进行排行。用户可选择全部领域或特定领域，根据不同时间范围查看在特定领域以及时间段内的专家排行。其中前三名将会被特别的展示。后台将综合多个指标为专家进行排序。此用例的执行者是普通用户和后台管理人员。



# 逻辑视图

****

对于我们的逻辑架构，我们采用了3 Tiers的三层式架构风格，主要分为表示层，业务逻辑层以及数据层。

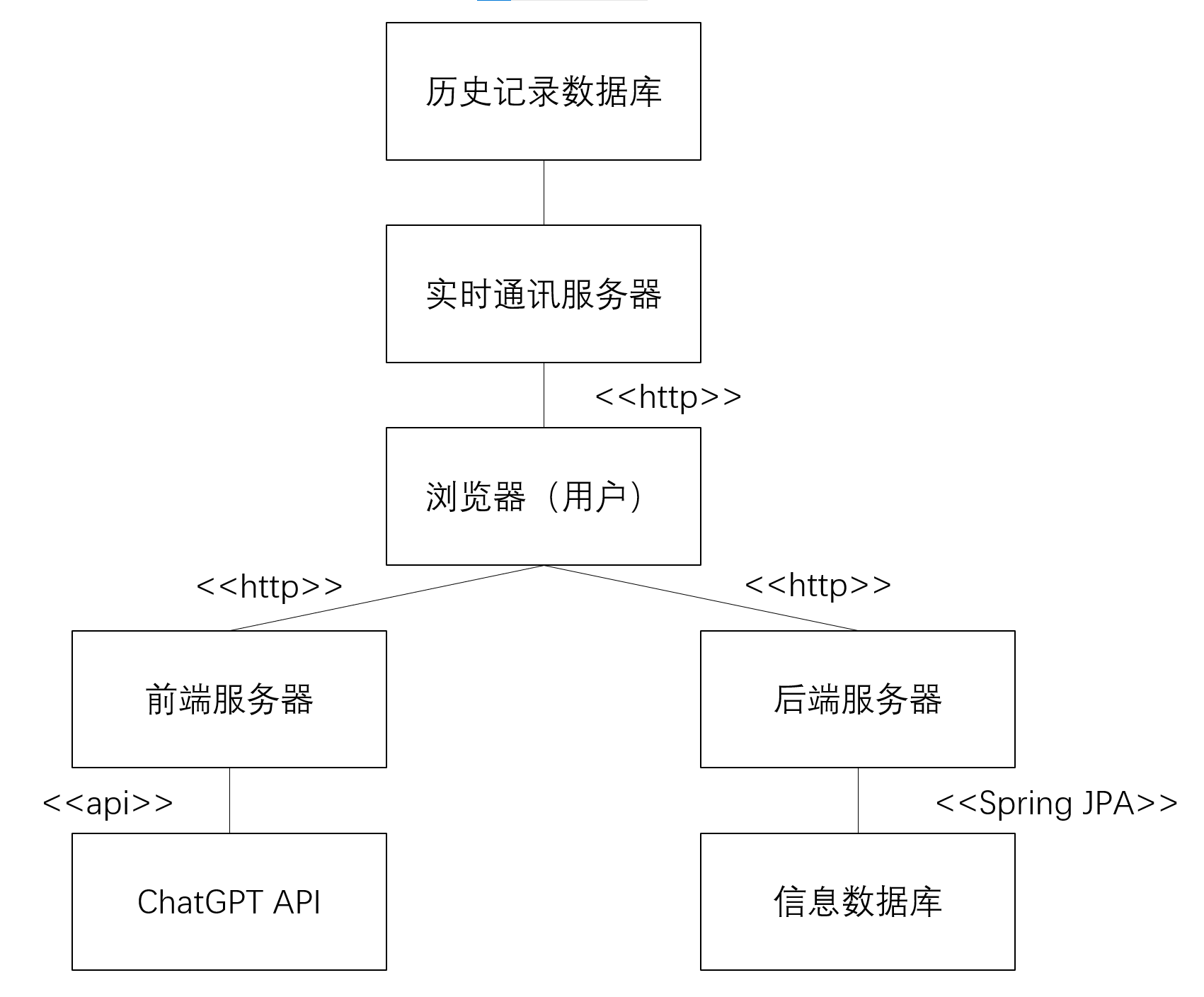
在表示层中，我们基于vite+react的框架进行前端的实现，而客户端浏览器通过基于axios的http请求与业务逻辑层进行连接。

在业务逻辑层中，我们主要分为三个子层：Controller，Services和Entities。主体后端服务基于了Spring Boot框架进行了实现。在Controller中，我们通过Spring Security，Spring Session的方法，基于Spring MVC的架构进行了参数解析，会话管理，用户的请求认证等功能。在Services中，我们以Java类对聊天咨询服务，语音咨询服务，专家评价服务，文章推荐服务进行了实现。在Entities中，我们同样以Java类对专家，用户，文章，时间，动态，统计等实体进行了实现。Services和Entities通过Java对象传递的方式进行连接。

在数据层中，我们主要分为数据访问层和数据存储层。数据访问层通过Spring Data JPA进行实现，数据存储层使用了MySQL数据库。通过数据库访问DAO模式，以JPA方法对MySQL数据库进行访问后，再以Java对象的形式与Entities子层进行连接。

以上就是我们基于3 Tiers的三层式架构风格的逻辑架构图的阐释。

# 部署视图



对于我们的物理架构，由于我们采取的是前后端分离的模式，因此用户在PC端的浏览器会对前端服务器和后端服务器发起http请求进行信息交换。其中，前端服务器也会调用ChatGPT API在前端中直接实现对用户的智能推荐功能。而在后端，我们通过Spring Data JPA的方式建立后端服务器与信息数据库的链接。而为了实现实时咨询的功能，我们也引入了实时通讯服务器，用户通过http请求对服务器进行链接。而为了存储实时通讯的历史记录（聊天记录等数据），我们也引入了历史记录数据库这一模块，但由于具体实现方式仍正在研究中，因此没有画出实时通讯服务器与历史记录数据库之间的链接方式。

# 质量属性的设计

1. 易用性：
2. 平台风格简约，图标清晰，常用功能易于寻找。
3. 平台提供详尽的用户新手指南，帮助初次使用的新用户尽快熟悉平台。
4. 在涉及到重要操作的地方，提供取消或撤消操作的选项，让用户可以回退到之前的状态。
5. 实现即时的反馈机制，例如在用户提交表单或操作后显示成功或失败的消息，以及相应的错误信息。
6. 通过用户调研和用户行为分析，建立用户模型，了解用户的需求、偏好和行为习惯，为设计提供指导和参考。
7. 可靠性：
8. 实时监控系统运行状况，通过日志记录和报警系统，追踪系统运行过程中的关键事件和异常情况，及时进行处理。
9. 设计良好的系统架构和组件隔离策略，确保系统中各个模块之间的独立性，降低故障扩散的风险。
10. 从服务中删除不稳定的功能，减少系统出错的可能性。
11. 设计检查点和回滚机制，保证系统在发生错误时能够快速恢复到之前的稳定状态。
12. 性能：
13. 优化平台的算法和代码结构，减少不必要的计算步骤，提高系统的运行效率。
14. 对于队列中的请求或任务进行大小限制，防止队列过大导致系统响应延迟或资源竞争。
15. 采用并发处理技术，充分利用系统的多核处理器和多线程能力，提高系统的并发处理能力，增加用户的响应速度。
16. 根据平台的负载情况和用户需求，适时增加服务器资源，如增加服务器数量或提升服务器配置，以应对高并发访问和大规模数据处理。
17. 将常用的数据或计算结果保存多个副本，减少对于相同数据或计算的重复计算，提高系统的响应速度和吞吐量。
18. 采用先进先出的任务调度算法，确保请求按照先后顺序进行处理，避免任务堆积和不公平竞争。
19. 安全性：
20. 实施严格的用户权限管理，用户被分为专家和普通用户两种不同的角色，应当并限制其访问和操作权限，确保用户只能访问其权限范围内的内容和功能。
21. 对于敏感信息采取加密存储和传输措施，限制对于敏感数据的访问权限，避免敏感信息被未授权的用户获取。
22. 记录用户的操作行为和系统事件，建立审计日志，并定期审查和分析日志数据，及时发现和应对潜在的安全威胁。
23. 定期进行安全漏洞扫描和渗透测试，发现并修复系统中存在的安全漏洞和弱点。平台上的专家都经过审核人员的严格审核，咨询的质量有所保障。