**软件架构文档**

**ALIVE**

**生机**——“今天你还活着吗”

软件项目计划

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 27/06/2023 | 1.0 | 初步确定软件架构设计 | 危思越、郭俊甫、杜心敏、徐一宸 |
|  |  |  |  |
|  |  |  |  |

目录

1.简介4

1.1目的4

1.2参考资料4

2.用例视图4

3.逻辑视图4

3.1概述4

3.2在构架方面具有重要意义的设计包4

4.进程视图4

5.部署视图4

6.实现视图5

7.技术视图5

8.数据视图（可选）5

9.核心算法设计（可选）5

10.质量属性的设计5

软件架构文档

1. **简介**

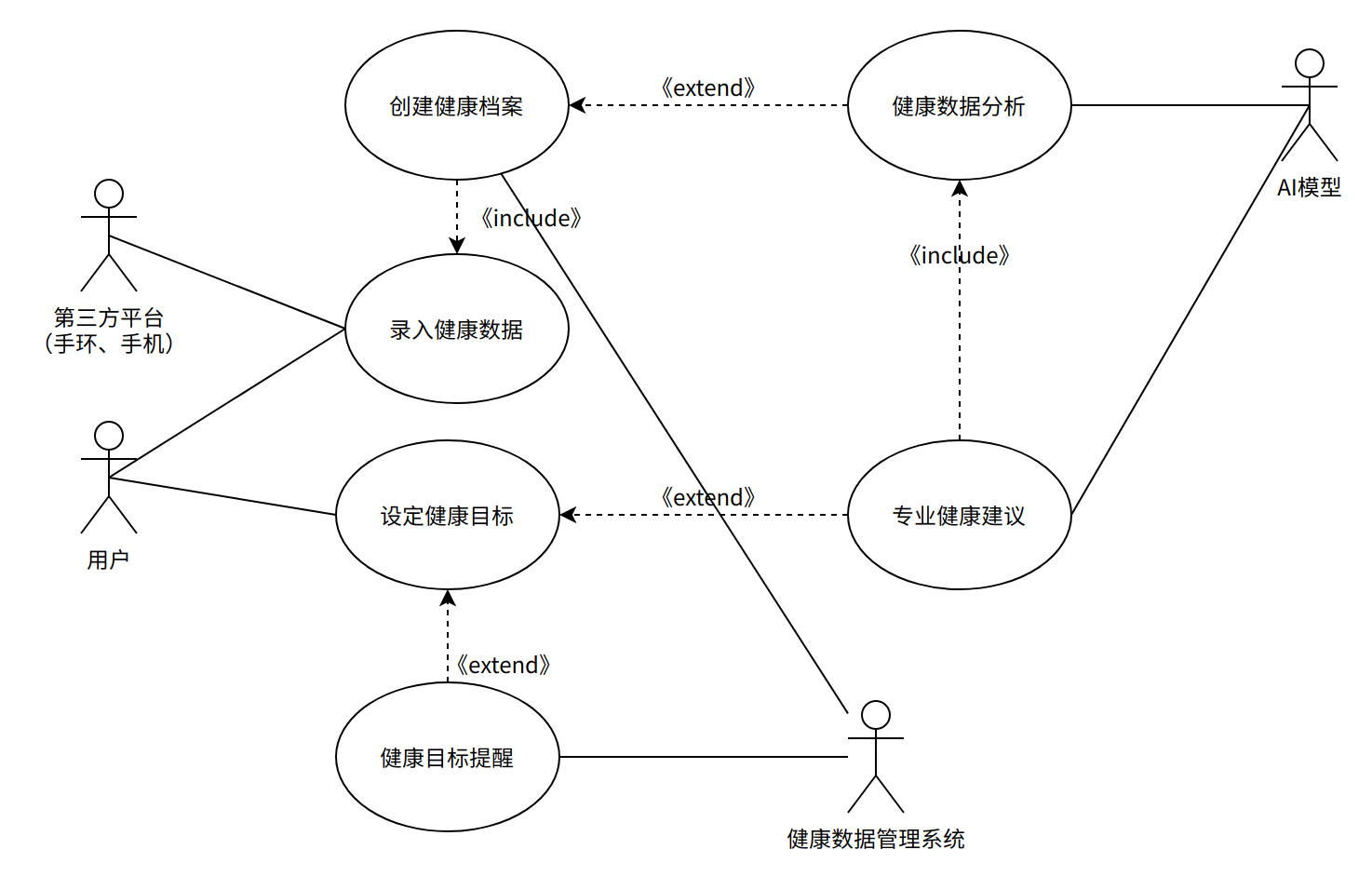
**目的**

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。

**参考资料**

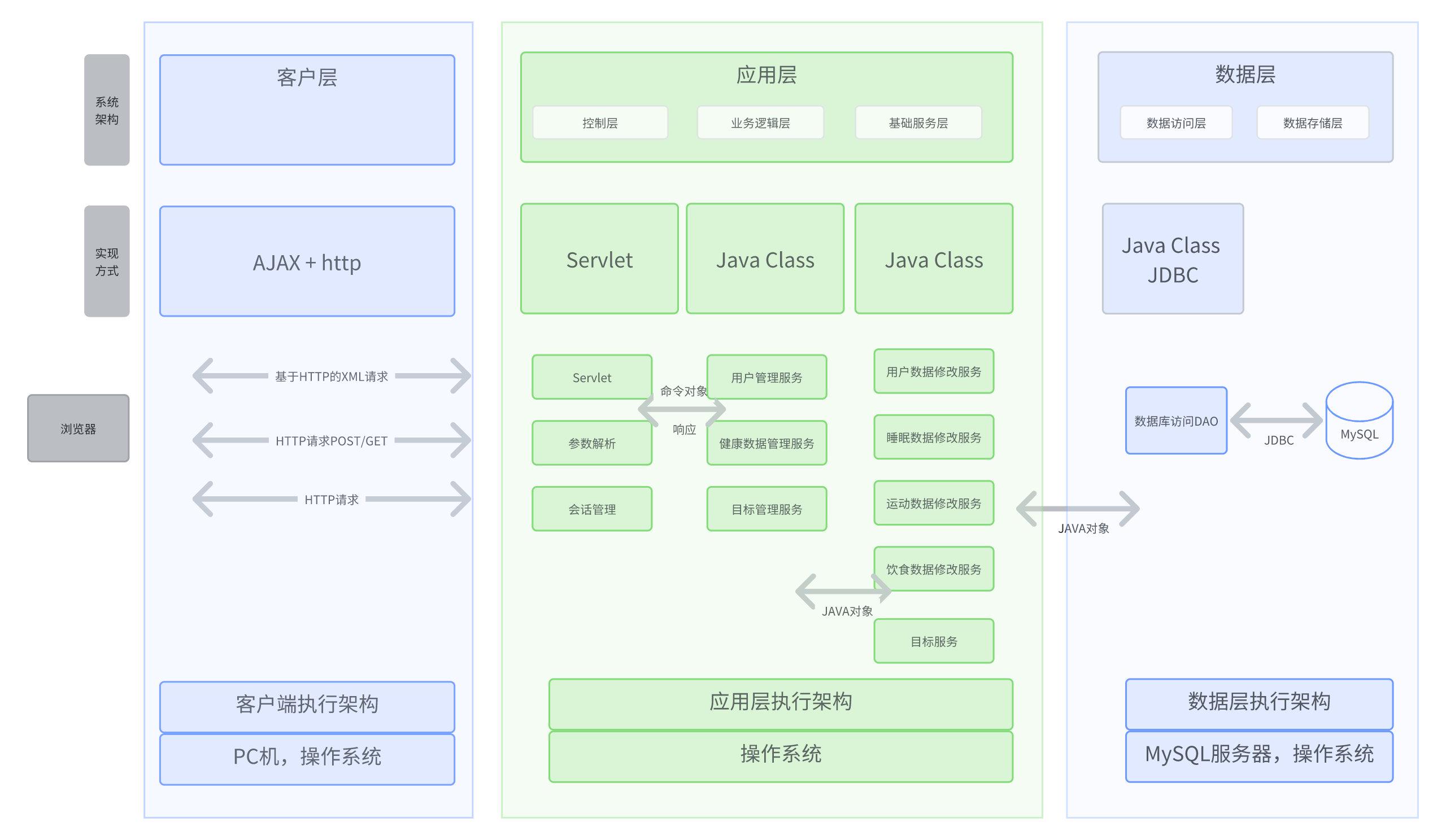
暂无

2. **用例视图**



3. **逻辑视图**

**概述**



采用MVC架构，客户层负责向用户呈现界面，并接收用户请求发送给应用层；应用层负责执行业务逻辑以处理用户请求，并调用数据层提供的持久性操作；数据层：负责执行数据库持久性操作。

**在构架方面具有重要意义的设计包**

**数据存储**



整个数据存储分为User, UserAuth, Goal, HealthRocord, SleepRecord, ExerciseRecord, DietRecord, xxxRecord, reord。

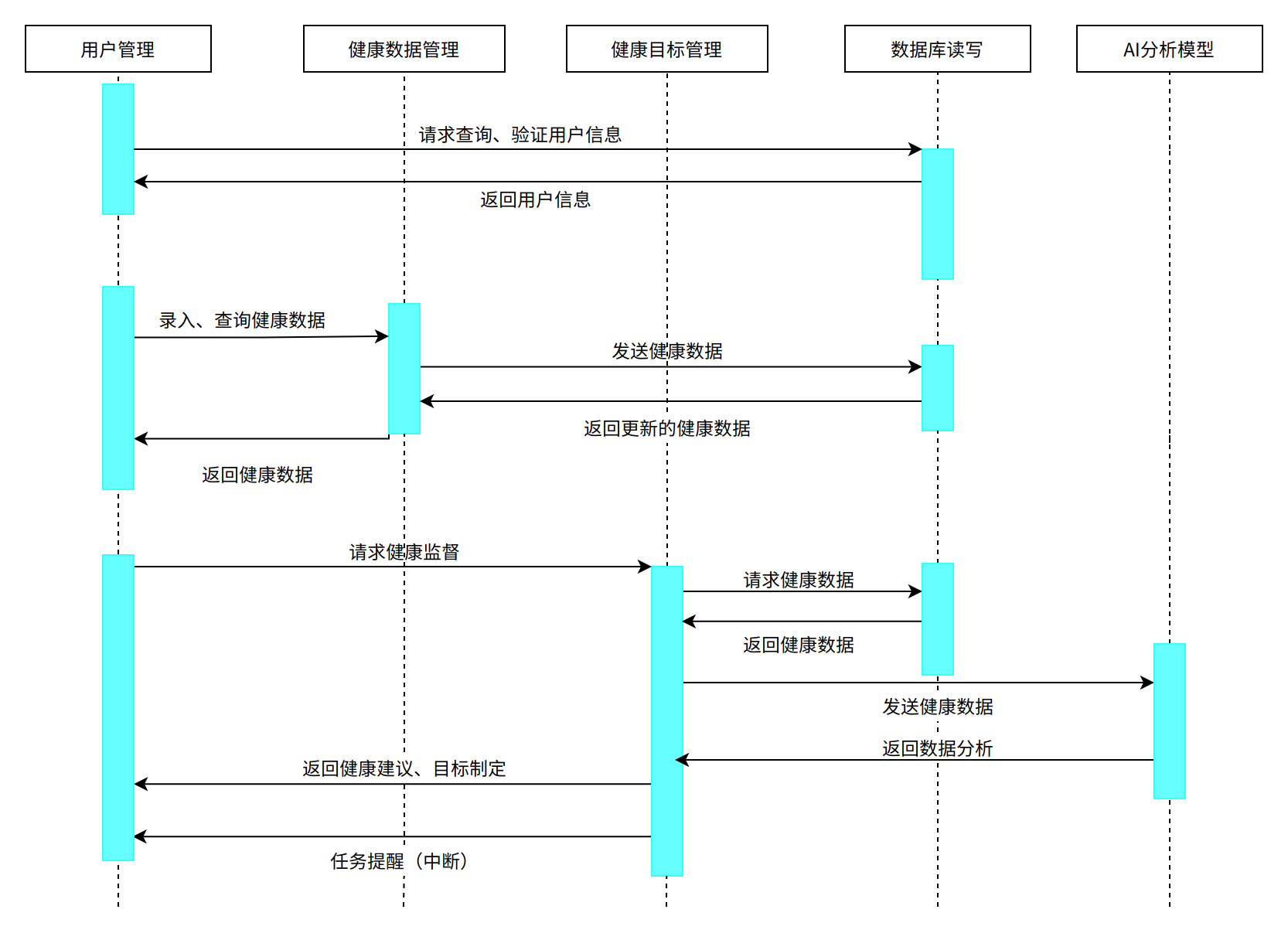
SleepRecord, ExerciseRecord, DietRecord, xxxRecord继承record。分别代表睡眠记录，运动记录，饮食记录，以及其他各项指标记录，如心率血压等，因为内容过多所以统一用xxxRecord表示。数据记录都可以增加和获取。

HealthReocrd记录最新一次的数据，包括睡眠，运动，饮食等方面，但记录内容更为简略，如只记录睡眠平均时长。

Goal记录每个用户的个人目标，可以设置修改，并发送提醒。

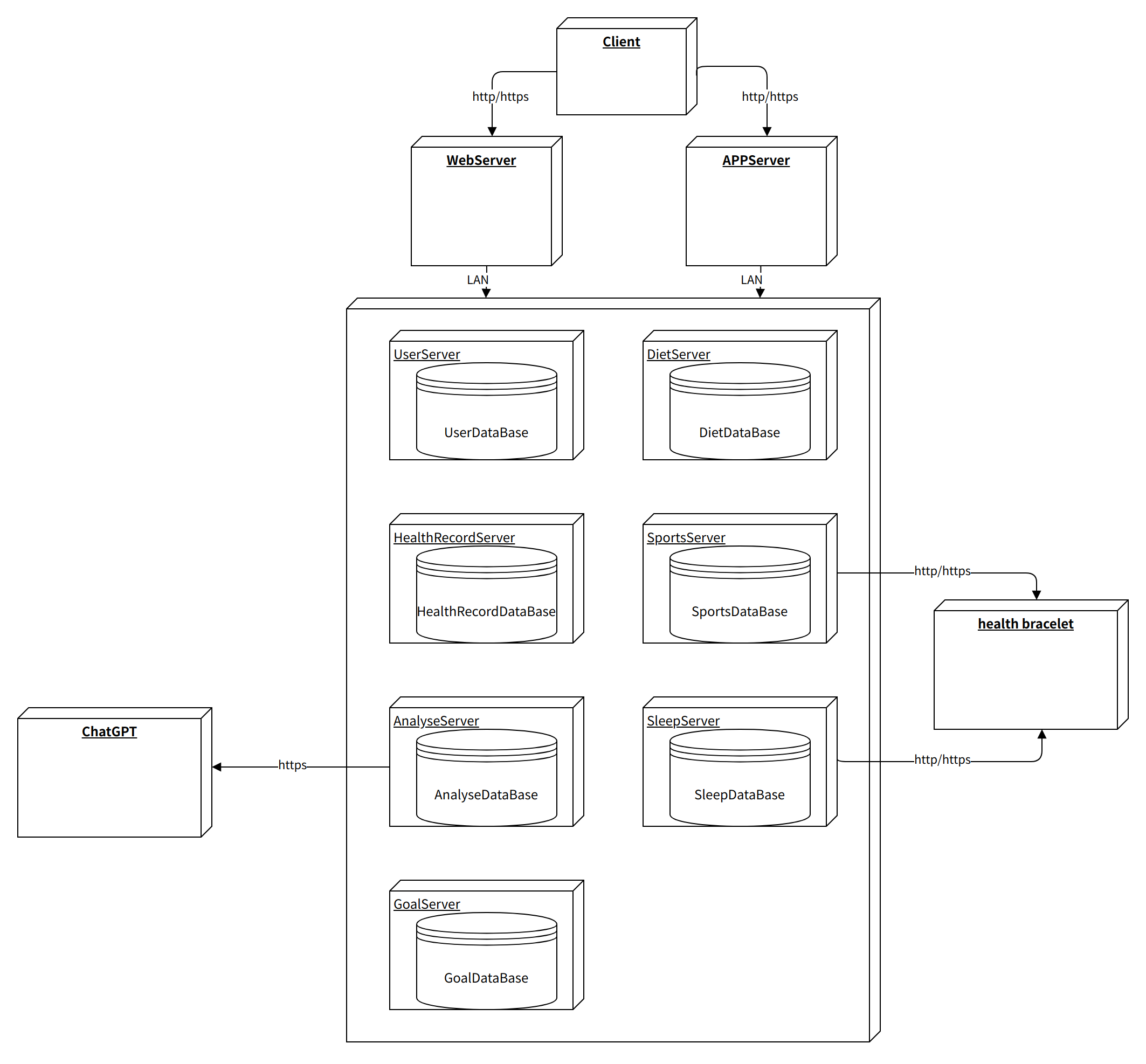
User记录每个用户的个人信息，每个用户都有一个UserAuth记录密码等私密信息。每个用户有自己的数据记录，记录可以有多条也可以没有。每个用户有个人目标记录，目标可以有多条也可以没有。

4. **进程视图**

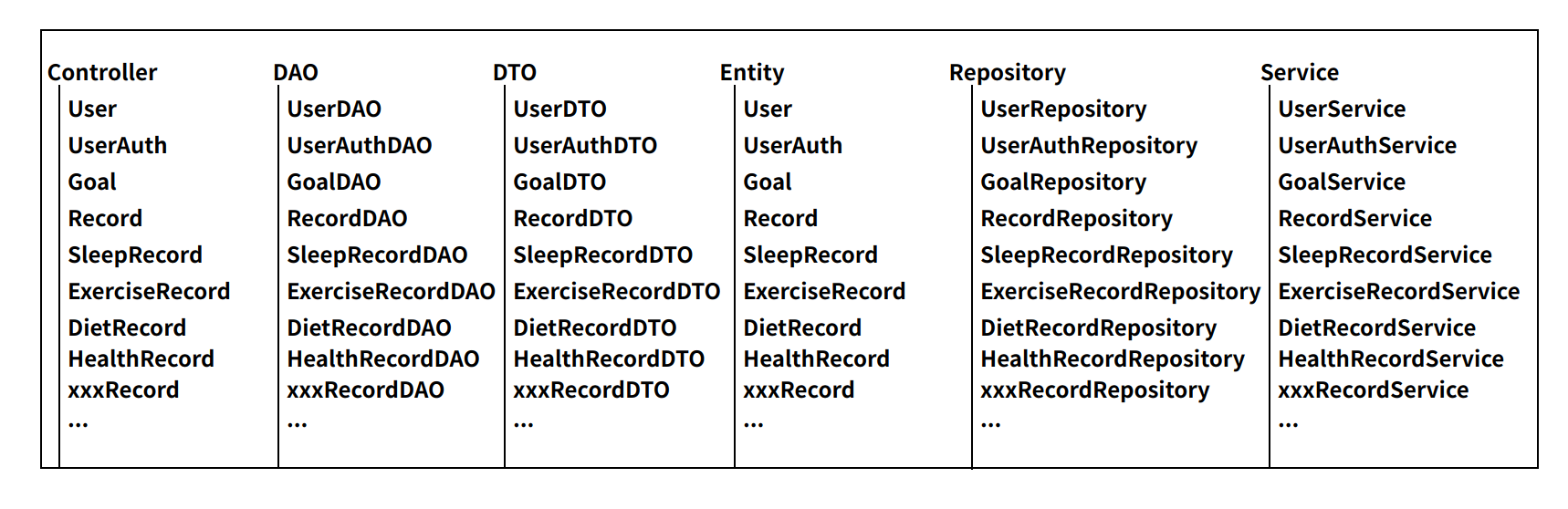


其中交互方式主要是消息传递，“任务提醒”处标注了是“中断”。

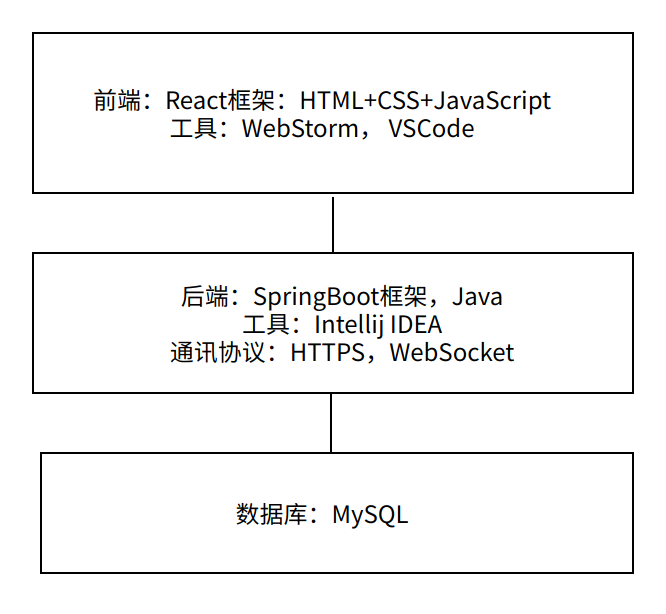
5. **部署视图**



6. **实现视图**



7. **技术视图**



8. **质量属性的设计**

[说明软件架构如何促成诸如性能、可扩展性、可靠性、易用性、可移植性等所有系统能力（而非功能）的实现。如果这些特征具有特殊的意义（例如在安全性或保密性方面的意义），则应该对它们进行详细的说明。]

* 性能：软件架构通过优化资源利用、减少响应时间和提高系统吞吐量来提升性能。
* 可扩展性：软件架构通过松耦合的组件设计和分布式系统架构，实现系统的可扩展性，比如通过微服务架构实现模块化和可伸缩的系统组件。
* 可靠性：软件架构应该具备高可用性和容错性，以保证系统的稳定运行。通过使用冗余和故障恢复机制可以增加系统的可靠性。
* 易用性：软件架构考虑用户体验和界面设计，以提供易于使用和友好的用户界面。
* 可移植性：软件架构支持在不同的硬件和操作系统平台上运行，并支持跨平台的部署。通过使用标准化的接口和协议。并避免与特定平台相关的依赖，可以实现软件的可移植性。