<项目名称>

软件架构文档

版本 <1.0>

[注：用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File>Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit>Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2/7/2024 | 1.0 | 初次撰写该文档 | 周泓宇、吴坤臻 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 5

3.1 概述 5

3.2 在构架方面具有重要意义的设计包 5

4. 进程视图 6

5. 部署视图 6

6. 实现视图 7

7. 技术视图 7

8. 质量属性的设计 7

软件架构文档

# 简介

## 目的

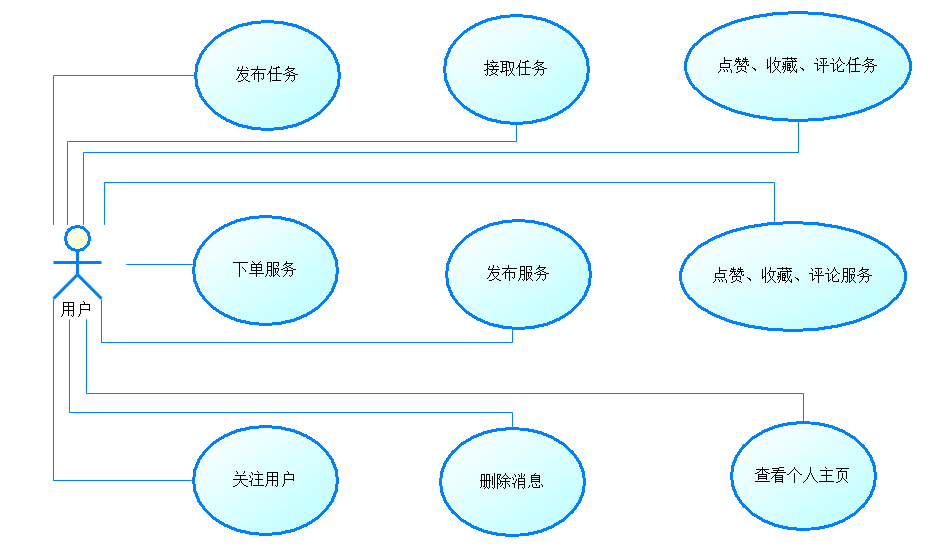
本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

## 参考资料

《软件项目计划》2024.6.25

《软件需求规约》2024.6.26

# 用例视图



此图为该项目的用例视图，其中包含了该项目的核心功能，包括发布和接取任务、服务，对任务、 服务进行点赞评论收藏，以及与其他用户聊天等功能。

# 逻辑视图

## 概述

## 逻辑视图

本项目采用了微服务风格，整个项目从上到下分为三层，分别是用户界面层、业务逻辑层和基础服务层。每一层中有不同功能的包，用户界面层与业务逻辑层之间通过http协议进行数据交互，而业务逻辑层和基础服务层之间通过jpa进行数据交互。

## 在构架方面具有重要意义的设计包

**3.2.1展示任务、服务**

该包是用户界面层的核心内容，用于向用户展示数据库中的任务和服务，并提供搜索功能。核心是任务类和服务类，他们包含了任务发布人、接收人等信息。

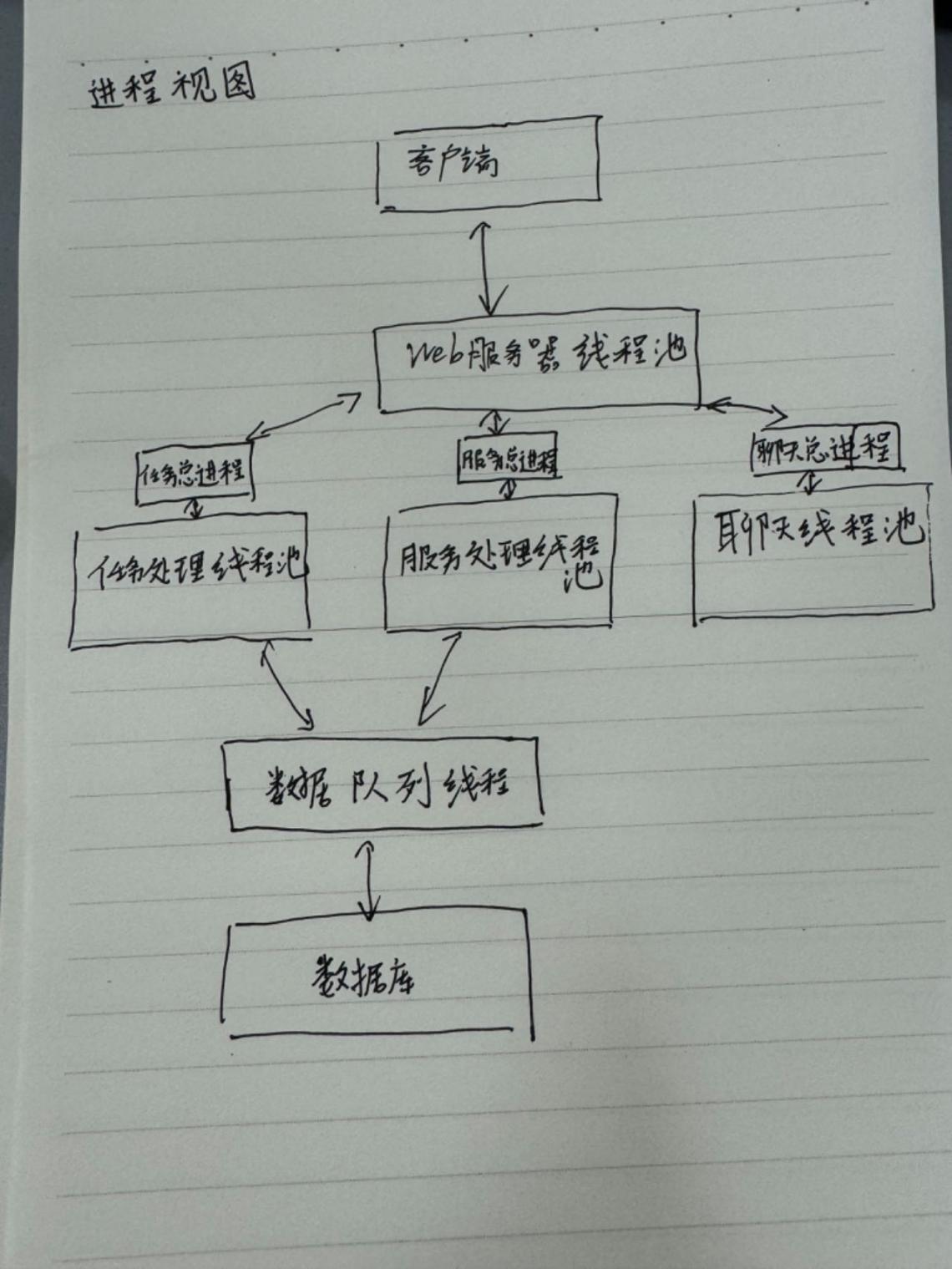
**3.2.2聊天界面**

该包用于实现用户之间的聊天界面，用户之间可以通过这个包看到显示的聊天文本。核心是一个聊天类，包含聊天对象、聊天内容和时间等信息。

**3.2.3聊天服务器**

该报用于处理用户聊天的数据并进行实时分发，

# 进程视图



当用户界面发送请求到后端时，后端通过不同的进程相应请求，然后分配对应的线程池，当需要从数据库中获取数据时，再将获取数据的请求插入数据队列线程中，从数据库获取信息并返回

# 部署视图

本软件将部署在华为云服务器上，通过项目组成员的电脑进行连接调试，这些电脑具有标准的商用硬件，例如多核CPU，足够的RAM和高速硬盘等。本项目的各个服务会部署在不同的物理机或虚拟机上，用过云平台的内部协议和http等协议进行数据传输。

# 实现视图

在软件的实现上采用分层的思想，主要分为用户界面层、业务逻辑层和基础服务层。用户界面层负责与用户交互的界面展示，使用React框架实现。业务逻辑层主要负责处理业务规则和用户请求，使用Spring Boot实现。基础业务层用于与数据库和其他持久化存储交互，本软件中使用的数据库

# 技术视图

本软件前端采用基于JavaScript语言的React框架，后端采用基于Java语言的Spring Boot框架，在Visual Studio以及Intellij Idea中进行开发。数据库则使用MySQL以及redis，最终整个软件在华为云服务器上部署运行。

# 质量属性的设计

性能：本软件通过分层，可以独立优化每一层的运行逻辑和性能，并且可以在层内使用异步处理和并发机制提升本软件的性能。

可扩展性：本软件采用微服务架构，将应用程序分解为一组小的、松耦合的服务，每个服务实现特定的业务功能。这种架构使得可以独立扩展各个服务，从而提高整个应用程序的可扩展性。

可靠性：在基础服务层实现严格的事务管理，确保数据的一致性和完整性，从而提高系统的可靠性。

易用性：在用户界面层采用现代的前端框架React，可以提高用户界面的交互性和响应性，从而提高整体的用户体验。

可移植性：遵循通用的编程标准和协议，如HTTP/HTTPS、JSON等，确保开发的软件容易在不同的平台和环境中部署和运行。

安全性和保密性：通过实现JWT安全协议，可以保护数据传输的安全和用户的认证信息，实现基于角色的访问控制，确保用户只能访问他们被授权的资源和数据。