生活记录仪

软件架构文档

版本 1.0

、

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <16日/5月/2023年> | <1.0> | <编辑完成第一稿内容> | <魏世泽><张宸玺> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 5

3. 逻辑视图 6

3.1 概述 6

3.2 在构架方面具有重要意义的设计包 6

4. 进程视图 7

5. 部署视图 7

6. 实现视图 8

7. 技术视图 8

8. 数据视图（可选） 8

9. 核心算法设计（可选） 8

10. 质量属性的设计 8

软件架构文档

# 简介

## 目的

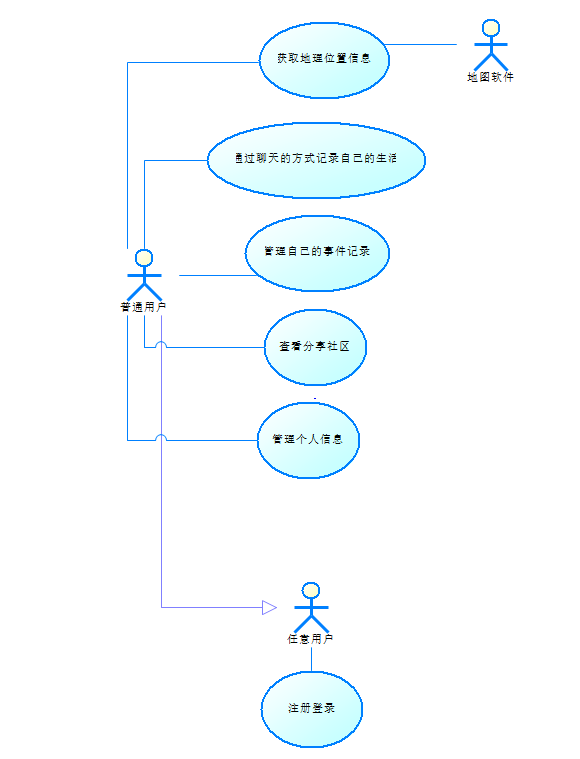
本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

该软件架构文档通过用例视图，逻辑视图等视图多方面展现软件的架构，便于开发人员进行开发。

## 参考资料

软件工程原理 沈备军，陈昊鹏，陈雨亭 著 高等教育出版社

# 用例视图



登录基本流：  
1、用户进入“生活记录仪”登录界面；

2、用户输入账号和密码；

3、用户提交输入的信息；

4、系统对账号和密码进行检查；

5、系统记录并显示当前登录用户；

6、用户可以正常使用软件的各种功能：记录并上传相关生活记录，分享、查看或修改相关记录内容。

事件记录基本流：  
1、用户点击首页上的“+”号进入记录界面；

2、用户选择相关的默认标签；

3、用户在对话框输入自己的感想或拍摄、上传照片；

4、用户点击发送将相关信息发送到对话界面；

5、用户重复若干次3、4操作直到记录结束；

6、用户确认记录结束。

统计管理基本流：  
1、用户点击“回顾”标签进入统计管理界面；

2、用户选择相关标签；

3、用户查看该标签下的所有记录；

4、用户点击“统计”查看数据图表及详细数据统计；

5、用户按记录内容进入记录查看界面；

6、用户通过记录查看界面右上角的红点进入记录编辑内容；

7、用户可编辑（删改）相关记录内容；

8、用户确认此次删改，可保存。

社区基本流：  
1、用户点击“回顾”标签进入统计管理界面；

2、用户点击待分享的事件；

3、用户点击“编辑”对待分享记录进行编辑；

4、用户编辑完成后点击“分享”；

5、用户在分享界面编辑相关信息如地理位置等；

6、用户完成生活记录的分享。

7、用户点击“社区”标签进入社区界面。

8、用户看到一系列事件的标题，简介和图片，可点击其进入查看详情。

9、用户看完后，可选择点赞评论，以及右上角的退出。

# 逻辑视图

## 概述

## 

一共分为三层，客户层，应用层和数据层。客户端发请求到后端，经过处理发送到业务层，通过

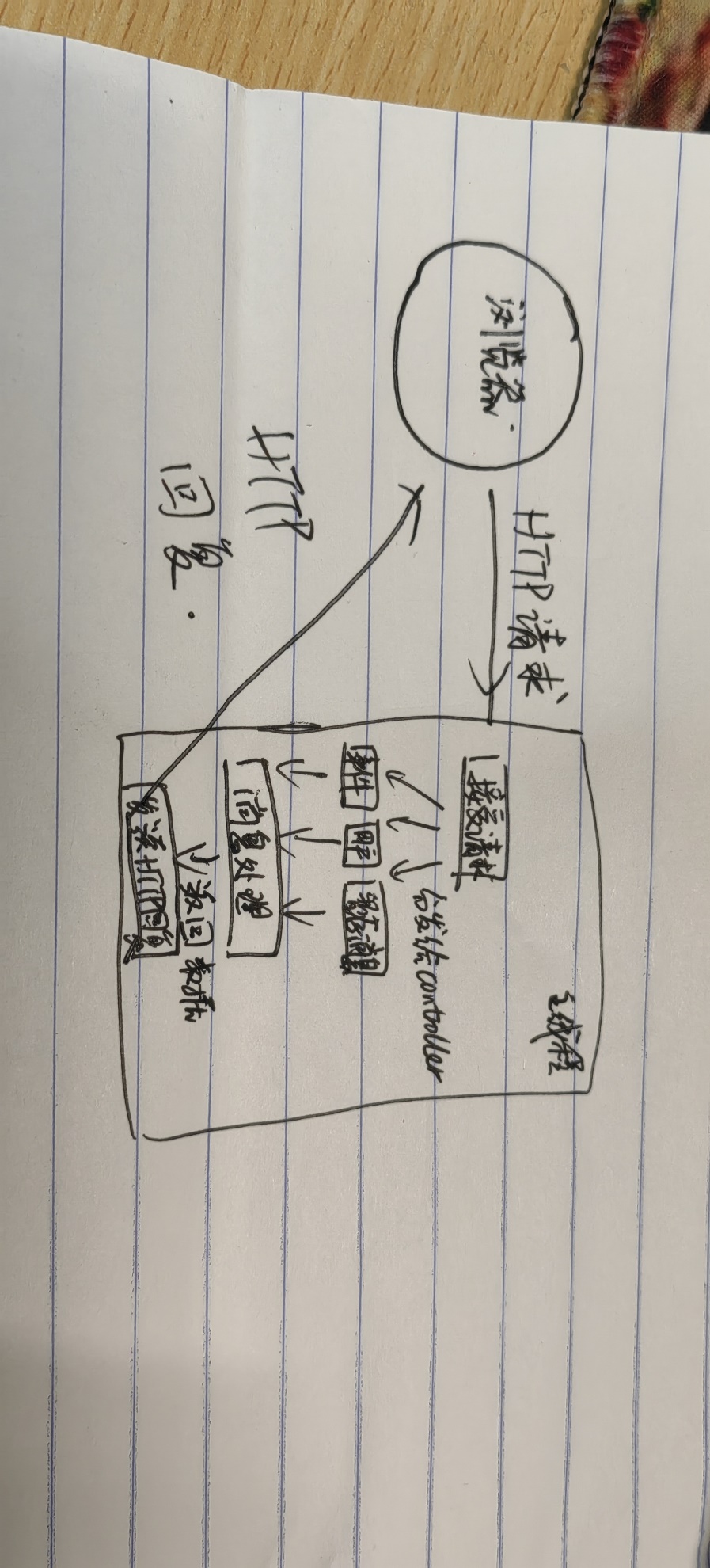
Java类通信分发到服务，再分发到DAO层，DAO层通过JDBC访问数据库。

## 在构架方面具有重要意义的设计包

Controller包中包含了所有的控制器类，这些类负责处理来自客户端的请求并将请求分发到相应的服务层。控制器类通过接收客户端请求，对请求数据进行解析和验证，然后调用对应的服务类进行业务处理。控制器类还负责将处理结果封装成响应数据，并将响应数据返回给客户端。

Service包中包含了业务逻辑层的所有服务类。服务类负责处理具体的业务逻辑，如数据校验、业务规则实施等。服务类将业务处理结果传递给DAO层，以便持久化数据。服务类还可以调用其他服务类，以实现跨领域的业务逻辑处理。

DAO包（数据访问对象包）包含了所有的数据访问对象类。这些类负责与数据库进行交互，执行增删改查等操作。DAO类通过JDBC技术访问数据库，并将数据库查询结果封装成Java对象，以便在服务层进行进一步处理。

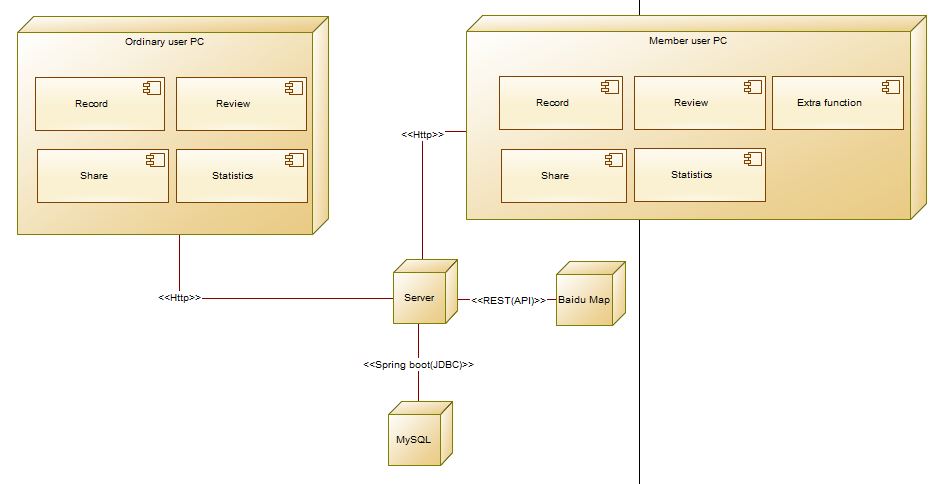
通过这些设计包，我们的软件架构实现了良好的分层和模块化，有助于提高系统的可扩展性、可维护性。

# 进程视图

本程序为单线程程序。

浏览器发出请求给后端，后端接收到请求后发给相应的控制器进行处理，在控制器处理完毕后相应的数据将被打包成HTTP请求的标准格式发回给浏览器，浏览器重新渲染并更新页面。

# 部署视图



1.用户PC

用户PC分为普通用户和会员用户，具有记录，回顾，分享，统计的功能，其中会员PC具有额外功能。用户通过HTTP信号连接服务器。

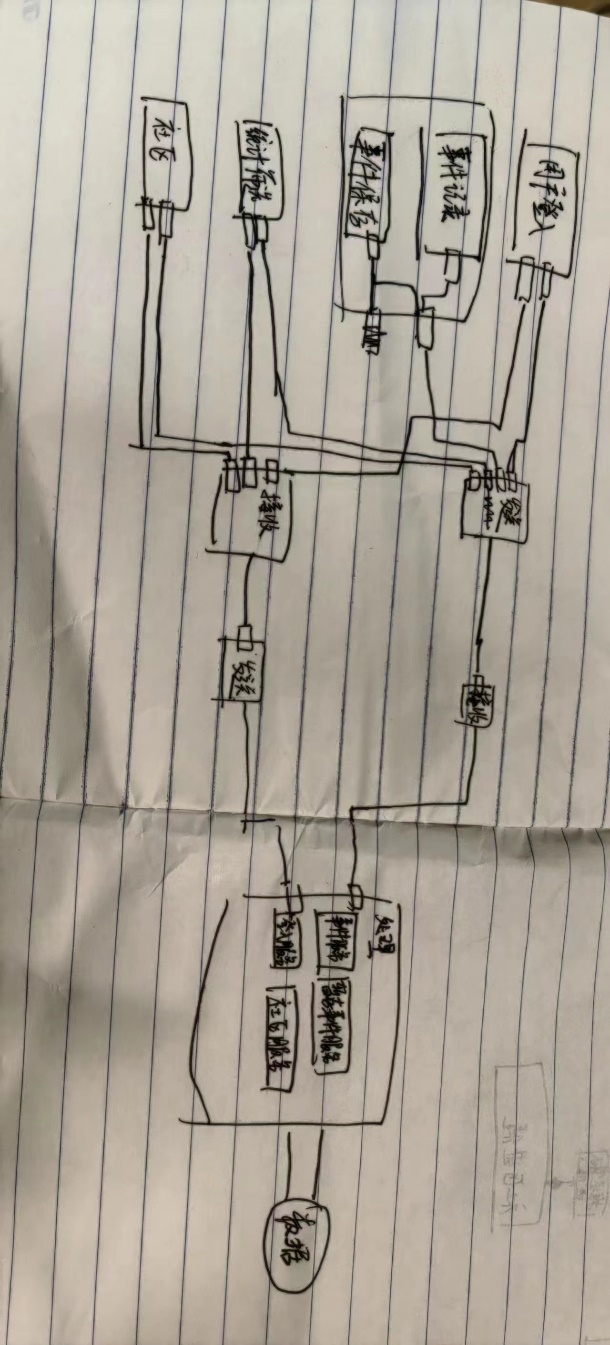
2.服务器

服务器相应用户的请求，通过REST调用Baidu Map的API。

3.数据库

MySQL数据库通过Spring Boot，JDBC方式与服务器连接。

# 实现视图



前端分为用户登入，事件，统计，社区四个模块，用ajax方法统一发送请求，后端统一接收分发给各控制层模块处理，与数据库交互得到结果后统一发送HTTP请求给前端。前端收到后统一处理，将数据分发给各模块。

# 技术视图

前端使用React框架开发，使用HTML+JS开发；后端采用Springboot框架，使用Java开发；数据库选用mysql数据库，通过JDBC连接。

# 数据视图（可选）

尚未完成

# 核心算法设计（可选）

尚未完成

# 质量属性的设计

1.性能：

1.1优化数据库查询，通过合理的索引设计和查询优化，降低数据库的查询时间。

1.2异步处理技术，对耗时的操作进行异步处理，提高系统的响应速度。

2.可扩展性：

我们的软件架构采用微服务架构，允许快速、独立地开发和部署新功能。通过容器化部署，可以轻 松实现系统的横向扩展。

3.可靠性：

3.1数据备份：定期备份数据库和系统文件，确保在发生意外情况时可以迅速恢复。

3.2容错处理：设计和实现容错机制，使得在某个组件发生故障时，系统仍能正常运行。

4.易用性：

4.1采用直观的用户界面设计，使用户能够快速上手。

4.2提供详细的用户手册和帮助文档，方便用户随时查阅。

5.可移植性

我们的软件将采用跨平台技术，使其能够在不同的操作系统和设备上运行。通过使用如HTML5、 CSS3和JavaScript等跨平台技术，我们确保了软件在多个平台上的一致性和可移植性。

6.安全性

采取严格的安全措施，确保用户数据的安全和保密。