



OnlyInt 知识分享平台

汇报人：第二小组

汇报时间：2024.6.20

目录

CONTENTS

01

产品特色与创新

02

架构与关键技术

03

特色与创新点

04

经验教训

05

成员贡献分工



part one

产品特色与创新

01

● 主要/基础功能 ●

文章编辑发布

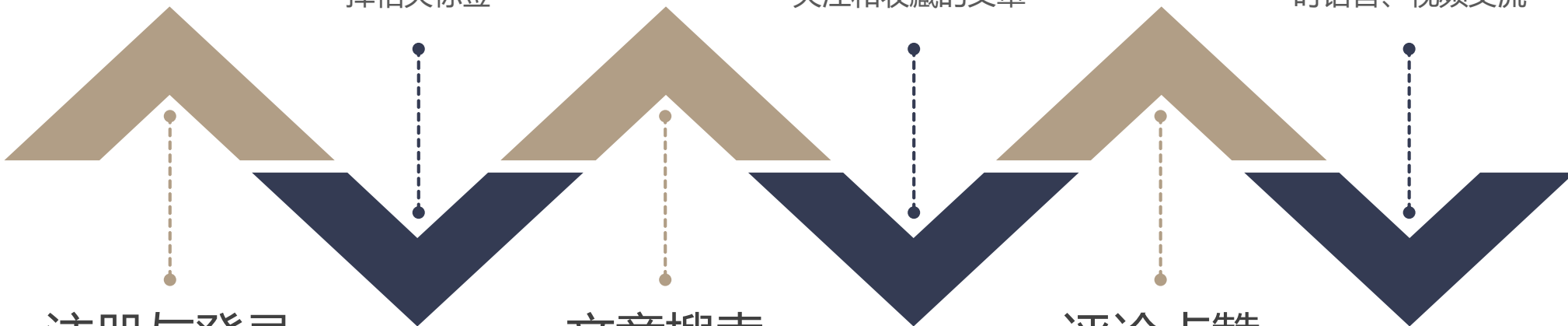
用户可以发布文章，包括文字、图片和链接，并选择相关标签

个人资料编辑

用户可以编辑个人信息、上传头像，并显示已发布、关注和收藏的文章

聊天室

创建语音视频讨论室，用户可以进入讨论室进行实时语音、视频交流



注册与登录

用户输入账号、密码进行注册与登录

文章搜索

用户可以通过关键词或标签搜索感兴趣的文章

评论点赞

用户可以对文章进行评论和点赞，展示社区活跃度

● 特色/创新功能 ●

用户排行榜

根据用户的活跃度、贡献度等指标加权计算积分，并进行排行，激励用户更积极参与。

文章编辑&Logo设计

集成了富文本编辑器，便于进行更完整的文章编辑操作；设计了软件Logo，加强了小组认同感



语言视频聊天室

用户创建/进入聊天室后，可以进行实时视频或语音聊天、屏幕共享、文字输入等操作

标签树

用户可以创建标签并添加标签关系、简介等信息，便于建立完善知识体系，梳理上下游知识

Logo

用户排行榜

根据用户的活跃度、贡献度等指标加权计算积分，并进行排行，激励用户更积极参与。

文章编辑&Logo设计

集成了富文本编辑器，便于进行更完整的文章编辑操作；设计了软件Logo，加强了小组认同感



语言视频聊天室

用户创建/进入聊天室后，可以进行实时视频或语音聊天、屏幕共享、文字输入等操作

标签树

用户可以创建标签并添加标签关系、简介等信息，便于建立完善知识体系，梳理上下游知识

part two

架构与关键技术

02

● 产品架构与技术栈 ●



整体架构思想

整体采用C/S架构，前端整体采用React框架，后端采用SpringBoot框架，数据库采用MySQL与Redis（Redis用于缓存用户行为数据、积分数据）

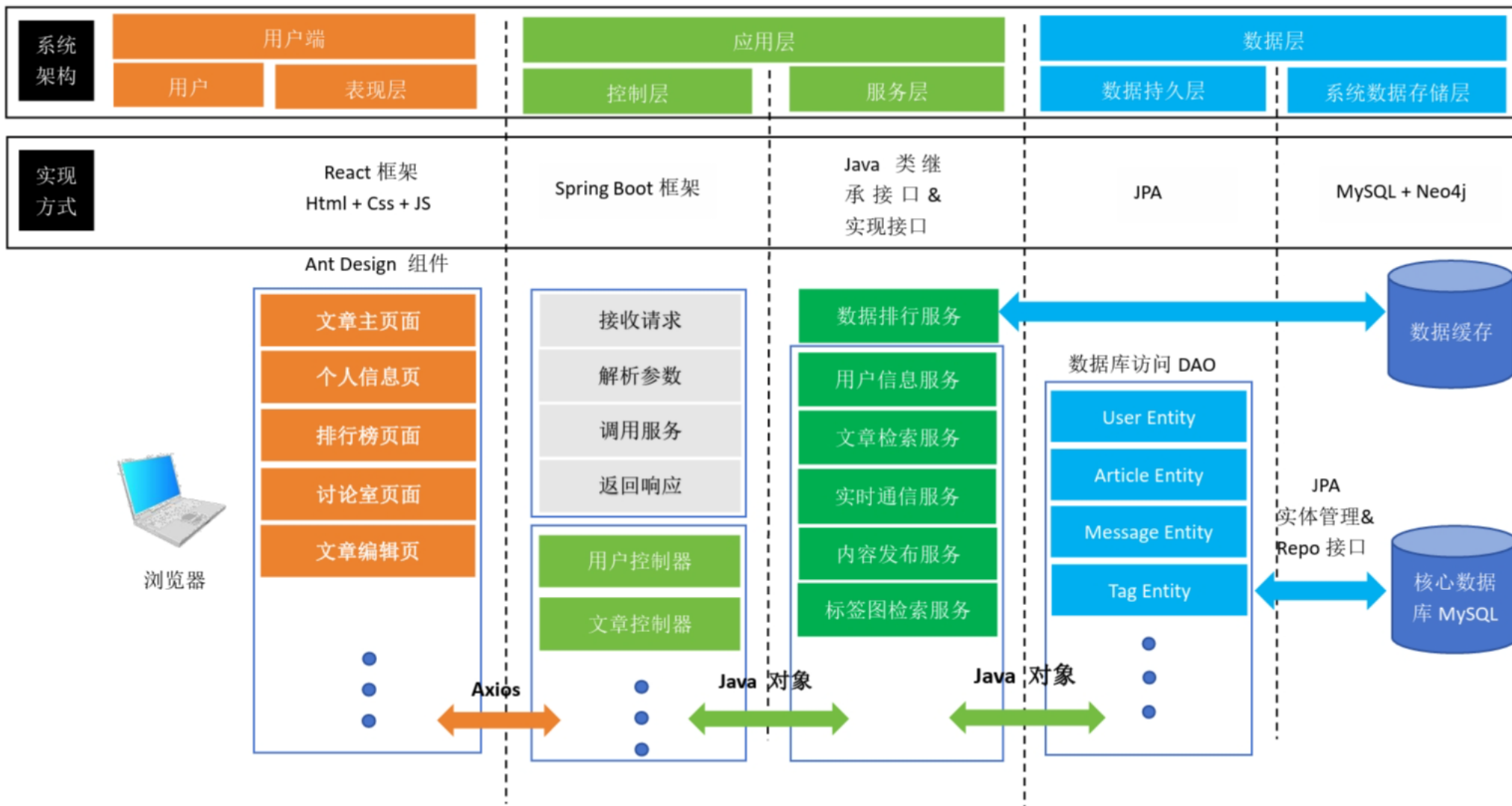
分层架构

通过层（Layer）来隔离不同的关注点（Concern Point），以此应对不同需求的变化，使得这种变化可以独立进行

组件化开发

开发和调试效率高，可维护性强，避免阻断，同时版本管理更容易

产品逻辑架构



● 产品关键技术 ●

实时通信/视频

接入了声网提供的服务器，
利用WebRTC实现了获取摄像头、
麦克风，进行屏幕共享等功能，
共同集成到聊天室组件中



图片存储

利用阿里云OSS云存储器进行
上传图片的储存

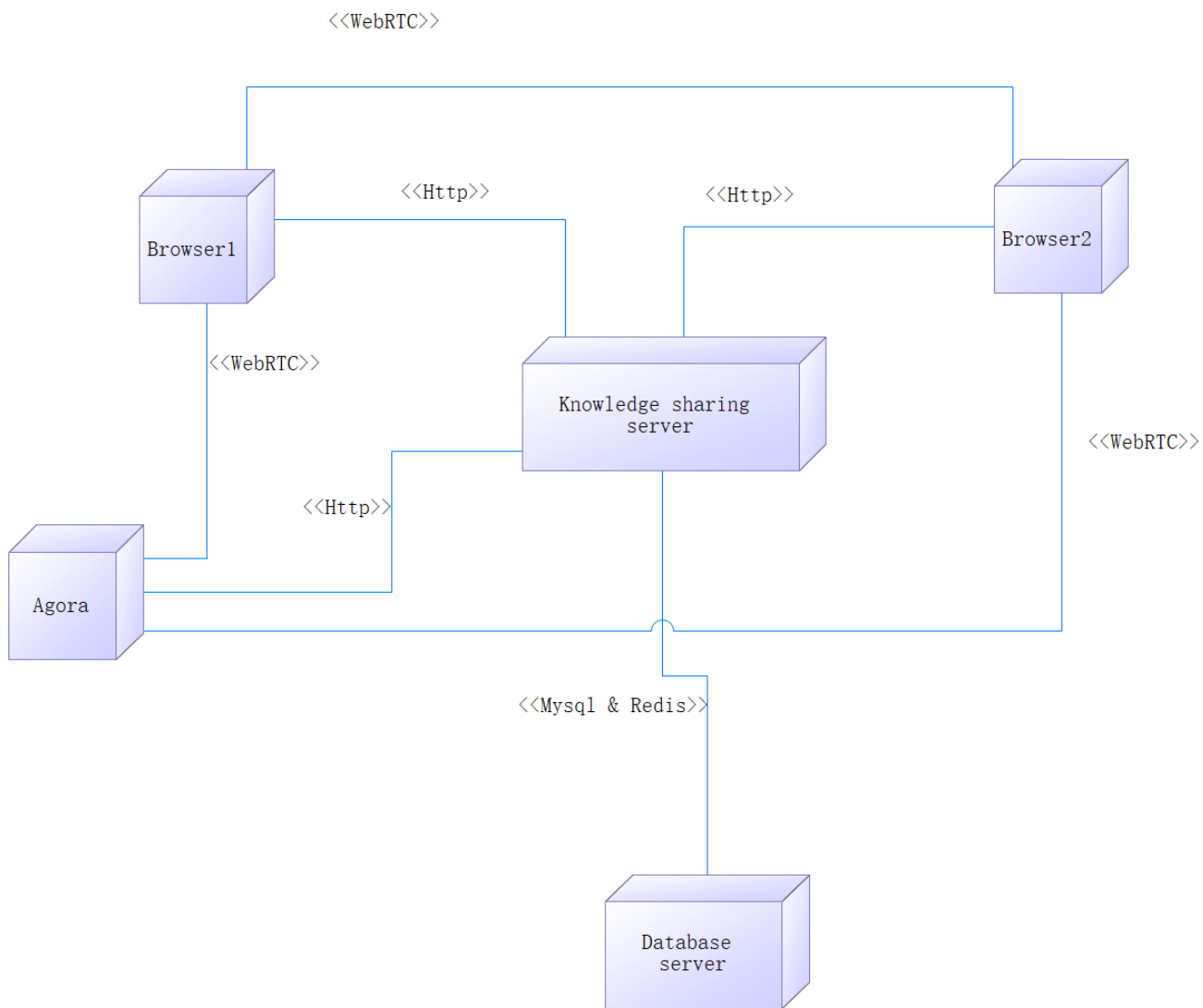


排行榜

采用Redis缓存用户的点赞数，
已发布文章等信息，更及时
地进行响应



产品物理架构



part three

特色与创新点

03



● 特色与创新点 ●

使用Redis实现缓存

针对用户排行榜这一即时功能实现，我们采用了Redis以提高响应速度

服务器部署

目前项目后端代码已经部署在了阿里云服务器上，便于后续维护开发

标签图谱

用户构建的标签、标签关系可视化，便于快速检索以及建立知识体系



语音视频聊天室

用户可以进行即时的聊天交流，寻找志同道合的朋友共同进行学习

文章暂存

用户还未确定好文章具体内容时，可以选择暂存将文章保存到草稿箱

人文关怀

软件整体采用了浅绿+深绿的清新风格，更具有活力与生命力

使用Redis实现缓存

通过Redis缓存高频率改动的点赞信息，利用有序集合ZSET进行用户积分排行

```









@Service
public class RedisRelationServiceImpl implements RedisRelationService {
    @Autowired
    RedisTemplate redisTemplate;

    4 个用法 Nurtibbar
    String type2String(RelationType type) { return type.toString(); }

    4 个用法 Nurtibbar
    @Override
    public void addRelation(RelationType type, Integer sender, Integer receiver) {
        String key = generateRelationKey(sender, receiver);
        redisTemplate.opsForHash().put(type2String(type), key, value: 1);
    }

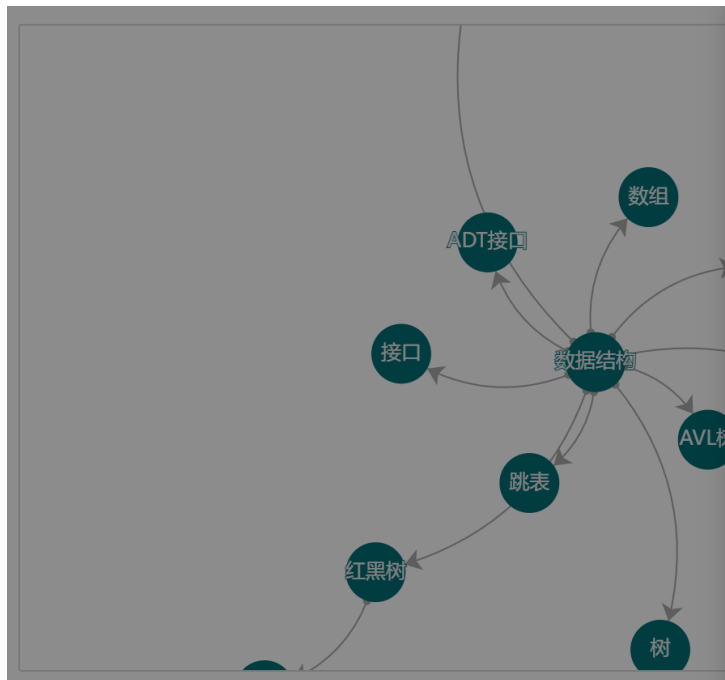
    5 个用法 Nurtibbar
    @Override
    public Boolean findRelation(RelationType type, Integer sender, Integer receiver) {
        String key = generateRelationKey(sender, receiver);
        return redisTemplate.opsForHash().get(type2String(type), key) != null;
    }
}

```

排行榜			
	WangYou 低调奢华有内涵	👍点赞:9 📄文章:4 🔥活跃度:29	查看个人主页
	吴勋豪 休言万事转头空	👍点赞:8 📄文章:3 🔥活跃度:23	查看个人主页
	nigulasiwang 晚安, 世界	👍点赞:6 📄文章:2 🔥活跃度:16	查看个人主页
	下一秒就睡着 蛤	👍点赞:6 📄文章:3 🔥活跃度:11	查看个人主页
	S1leepNextSec	👍点赞:2 📄文章:1 🔥活跃度:6	查看个人主页
	S1leepless	👍点赞:1 📄文章:1 🔥活跃度:4	查看个人主页
	LXH 啵~	👍点赞:3 📄文章:1 🔥活跃度:3	查看个人主页
	DUNHUANG	👍点赞:1 📄文章:1 🔥活跃度:1	查看个人主页

标签图谱

用echarts绘制标签图谱，对标签之间的关系进行可视化展示，让用户对知识之间的联系一目了然



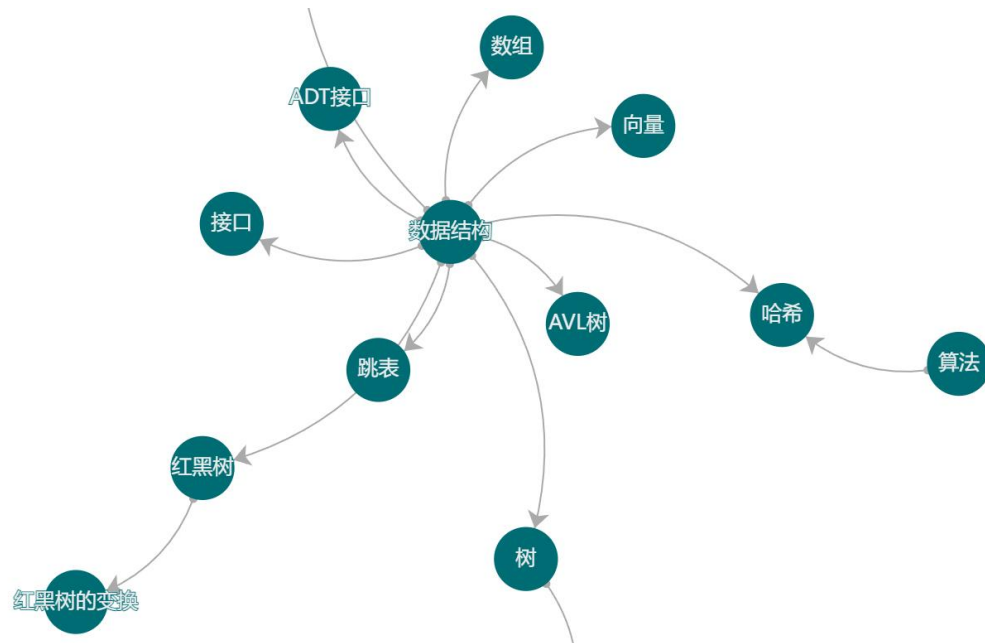
【数据结构】史上最好理解的红黑树讲解，让你彻底搞懂红黑树

什么是二叉树

二叉树遍历就是这么简单

数据结构 -- 图解AVL树(平衡二叉树)

什么是向量



文章暂存

标题

标签 + Java X JPA X

OnlyInt

正文 默认字号 默认字体 默认行高

之前介绍了Mybatis数据库ORM框架，也介绍了使用Spring Boot的jdbcTemplate操作数据库。其实Spring Boot还有一个非常实用的数据操作框架：Spring Data JPA。

Spring Data JPA是Spring Data框架下的一个基于JPA标准操作数据的模块，简化了操作持久层的代码，只需要编写接口就可以。如果你第一次使用Spring Data JPA，你一定会惊呼这东西简直就是神器，几乎不需要写什么关于数据库访问的代码一个基本的CURD的功能就出来了。使用非常简单方便，下面我们就来讲述JPA使用的基本操作。

一、Spring Data JPA介绍

JPA (Java Persistence API)，Java持久层API的简称，是JDK 5.0注解或XML描述对象- 关系表的映射关系，并将运行期的实体对象持久化到数据库中。JPA是一个基于O/R映射的标准规范。

JPA是Spring Boot官方推荐的数据库访问组件，充分体现了面向对象编程思想，有点像asp.net的EFCore，JPA也是众多ORM的抽象。

JPA的总体思想和原有Hibernate、TopLink、JDO等ORM框架大体一致。主要包括以下3方面的技术：

1. ORM映射元数据 将实体对象持久化到数据库表中
2. API，用来操作实体对象，执行CRUD操作
3. 查询语言，通过面向对象，而非面向数据库的查询语言查询数据

OnlyInt



- 主页
- 排行榜
- 聊天室
- 个人资料
- 文章
 - 编辑
 - 草稿箱
 - 已发布

吴勋豪

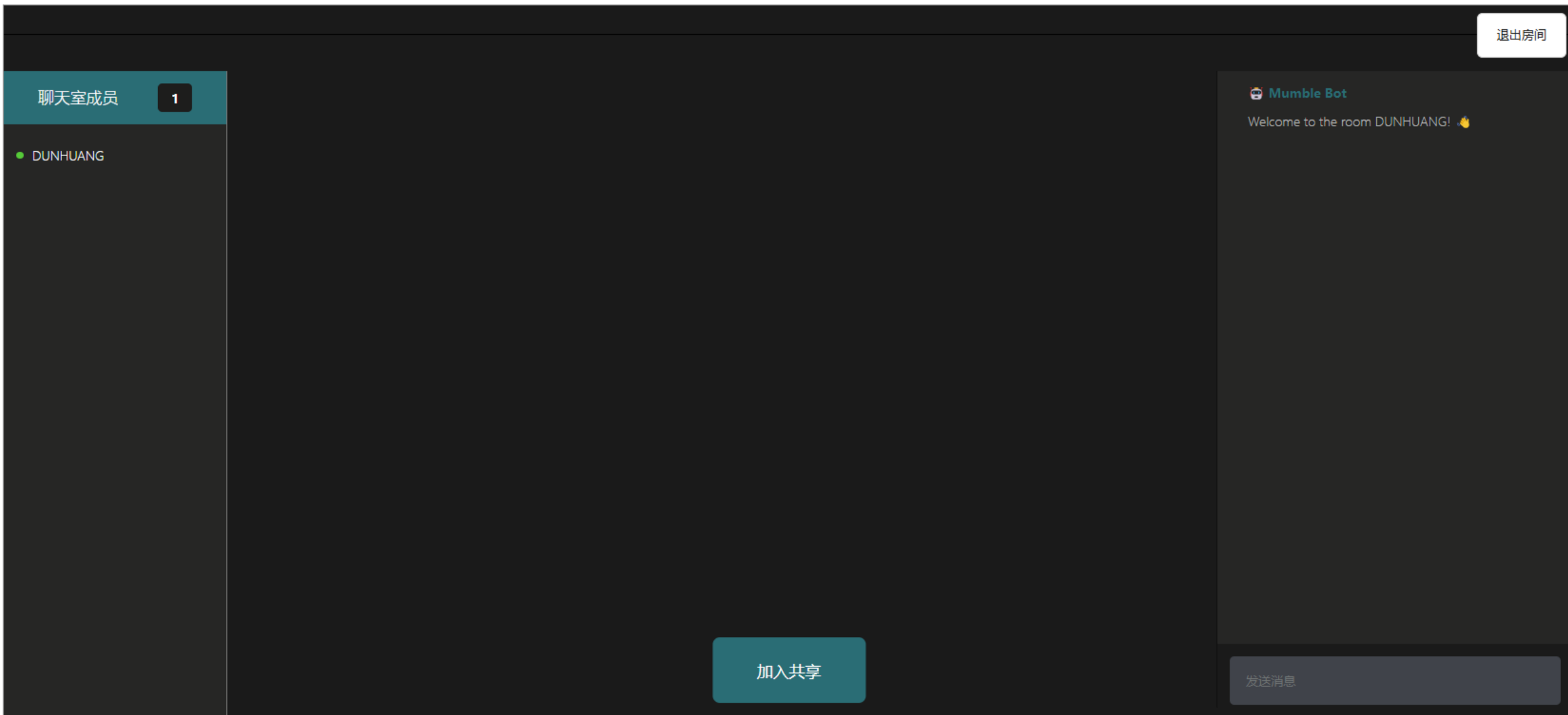


红黑树的变换

编辑 | 删除

数据结构 | 红黑树 | 红黑树的变换

● 语音视频聊天室 ●



part four

系统演示

04

part five

经验教训

05



经验教训

技术难关

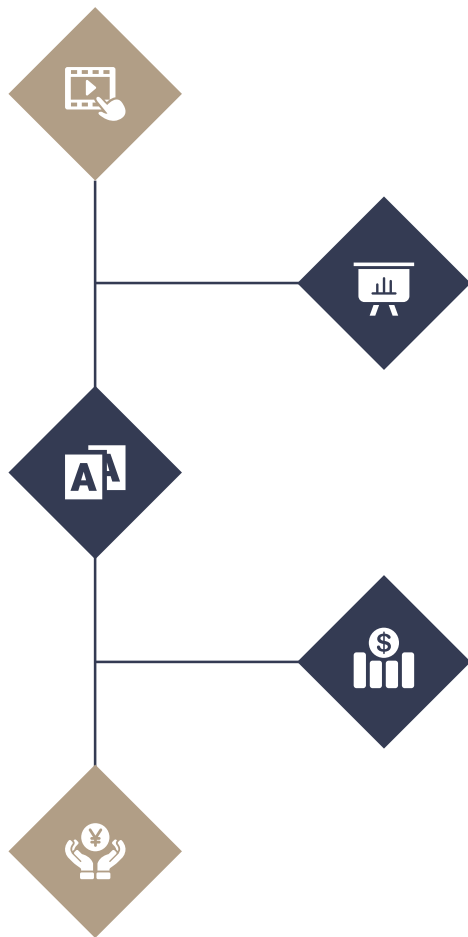
学习技术是持续发展地过程，特别是对于缺乏相关知识与经验的初级开发者，技术问题是一项挑战，需要付出更多精力攻克

定期的验收

长期的迭代需要做好定期的验收检查，设定好每一小阶段需要完成的功能，并及时进行定期评审，避免前后端开发进度拖延

交流反馈

协调团队工作比个人编码更重要，及时的沟通与跟进，统一的编码标准，成员任务的调整与分配，有助于提高团队效率和项目完成成果



开发进度/风险把控

进度风险很高，尤其是中期出现了人员变动后，即使经过紧急讨论，进度风险也在急剧升高；这个时候需要把握开发进度，果断舍弃优先级低的功能，集中冲刺完成核心功能

细粒度任务

在项目的最早期界面原型设计时，就应该规划出各个页面应当完成的具体任务，甚至是前后端接口以及需要的相关数据，明确的细粒度任务有助于加速项目高质量完成

part six

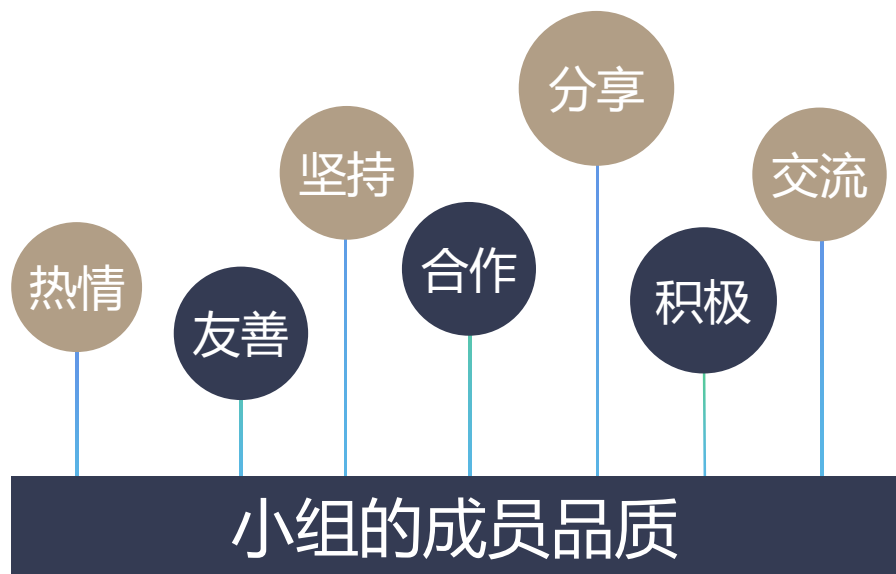
成员贡献分工

06

● 成员贡献分工 ●

共同完成的内容

项目的构思以及功能的设计，前端的界面原型设计，项目单元测试等



陈祺悦（组长）

负责后端整体开发，数据库设计，前端页面跳转的实现，以及图片上传、排行榜业务逻辑的实现等



王优

主要完成了前端的大部分业务逻辑，进行了部分页面风格优化，完成了项目的系统测试等



吴勋豪

主要完成了前端的大部分业务逻辑，进行了部分页面风格优化，完成了PPT的制作等

感谢您观看聆听

汇报人：第二小组

汇报时间：2024.6.20

