<PKBG和平战场：决斗>

软件架构文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 08/7/2019 | 1.0 | 完成软件架构文档的初步编写 | 王浩宇 |
| 12/7/2019 | 1.1 | 完善软件架构文档 | 谢宜含 |
|  |  |  |  |
|  |  |  |  |

**目录**

**软件架构文档 （简化版）**

1. 简介

目的

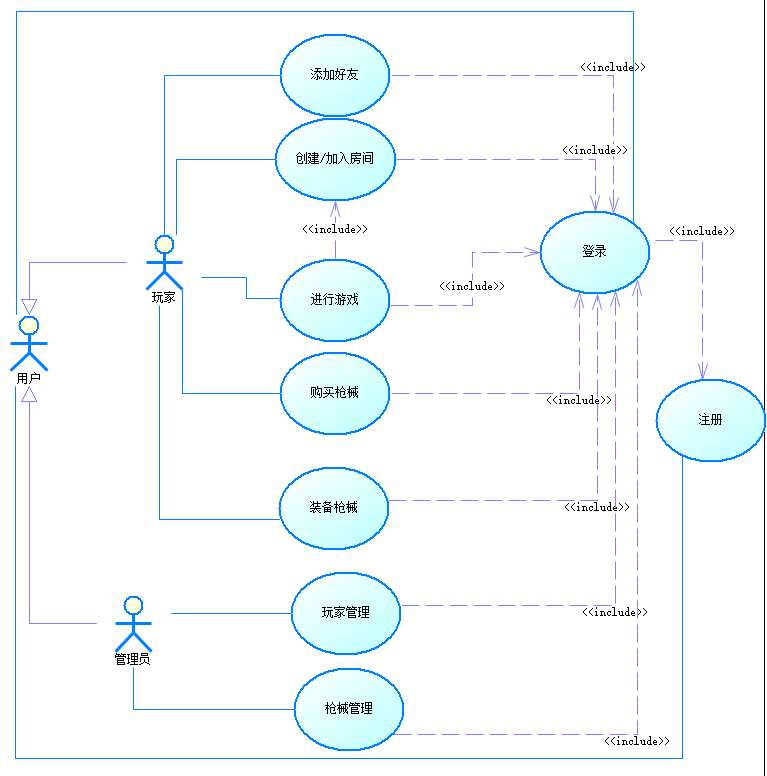
本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

1.2参考资料

1.《软件工程导论》

2. RUP的软件架构文档模板

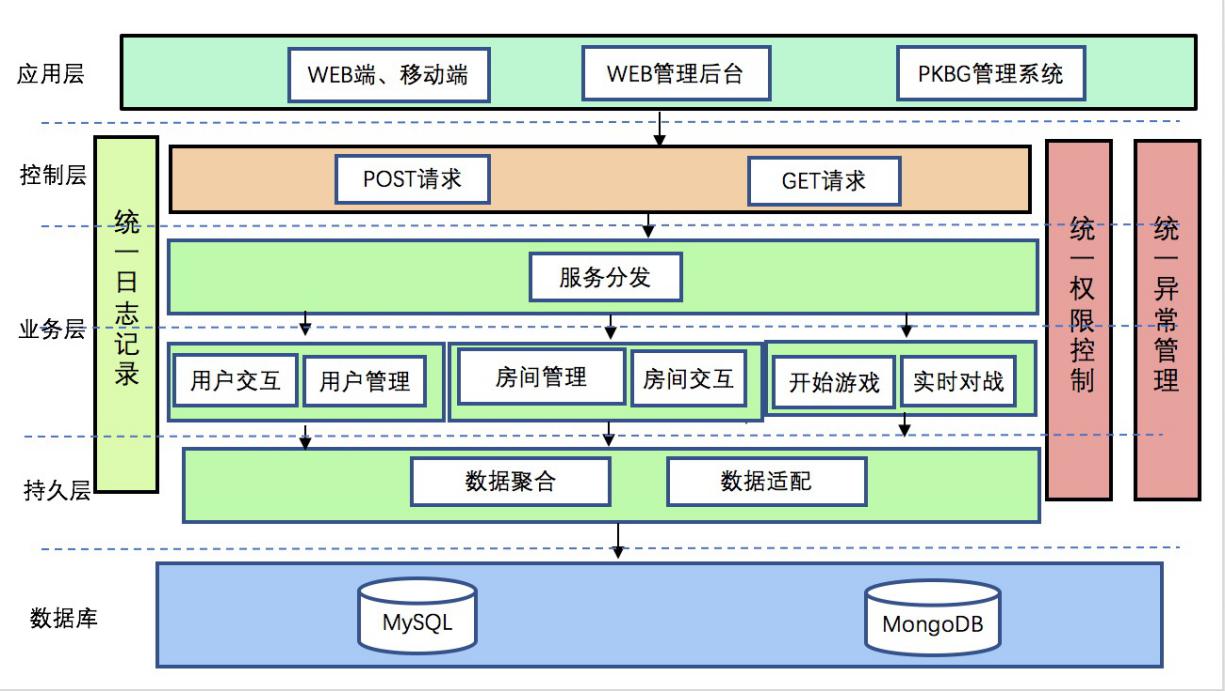
1. 用例视图



1. 逻辑视图

本章是对软件架构的逻辑视图的描述。主要内容包括描述重要的类，类的分包，子系统以及子系统的分层等。另外还包括一些重要用例的实现。

3.1概述



PKBG系统的逻辑视图主要由五层组成，分别是应用层、控制层、业务层、持久层、数据库。

* + 应用层

实现基于web端和移动端的应用，前端用React和React native框架，采用JavaScript语言；通过Spring boot搭建后台框架，采用Java语言，实现PKBG系统

* + 控制层

接收来自用户的请求并通过相应接口对相应的请求进行回应

* + 业务层

采用微服务架构，服务分发器通过服务注册中心联通三个子服务器，分别是：1.user-client负责用户交互 、用户管理（注册、登录、封禁用户、解禁用户）；2.room-client负责房间交互和管理（创建房间、加入房间、退出房间、准备、开始游戏）；

3.game-client负责开始游戏、实时对战。

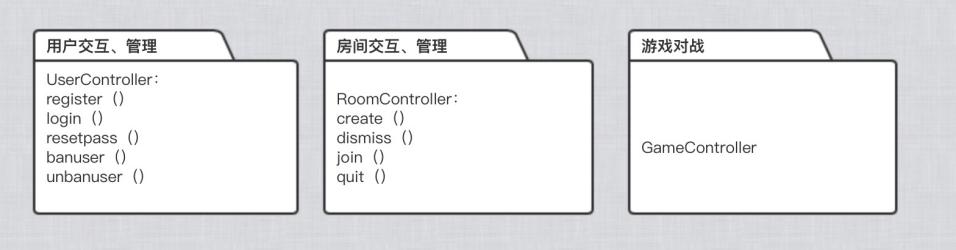
* + 持久层

主要将收集的数据进行聚合、转化、抽取、适配，最后传入到数据库

* + 数据库

主要收集用户所反馈的数据、传感器/监控器等所记录的数据

3.2控制层



控制层主要由3个子包组成，这3个子包如图所示。控制层主要实现对前端发送到后端的请求进行处理的过程。

UserController处理用户登录和用户管理操作的请求（注册、登录、封禁用户、解禁用户）。

RoomController（改为web-socket后更名为myHandler）处理对房间交互和管理操作的请求（创建房间、加入房间、退出房间、准备、更换队伍、开始游戏）。

GameController处理对游戏实时对战等一系列操作的请求。

3.3 业务层



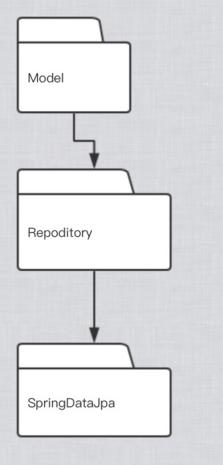
业务层主要由3个子包组成，如图所示。3个子包提供了对对应的业务进行处理的功能。

UserController提供了用户登录和用户管理操作（注册、登录、封禁用户、解禁用户）的功能。

RoomController提供了对房间交互和管理操作（创建房间、加入房间、退出房间、准备、更换队伍、开始游戏）功能。

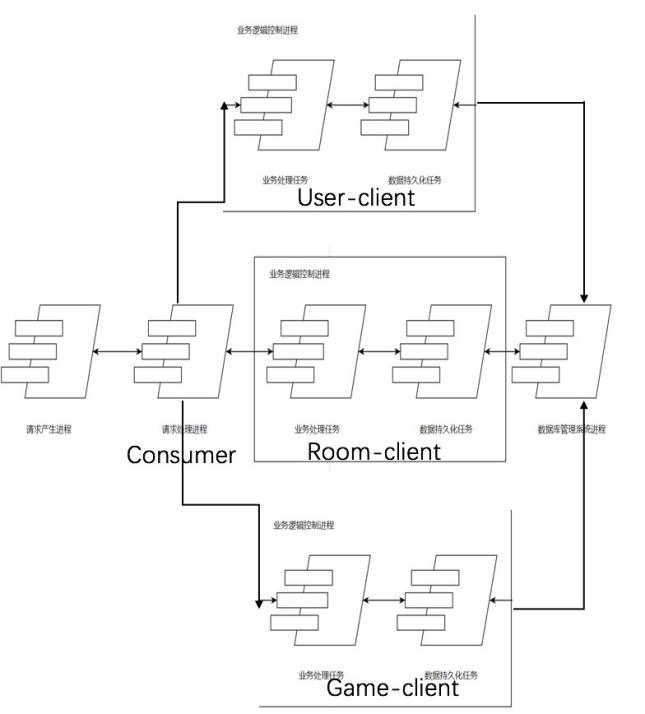
GameController提供了对游戏实时对战等一系列操作的功能。

3.4 持久层



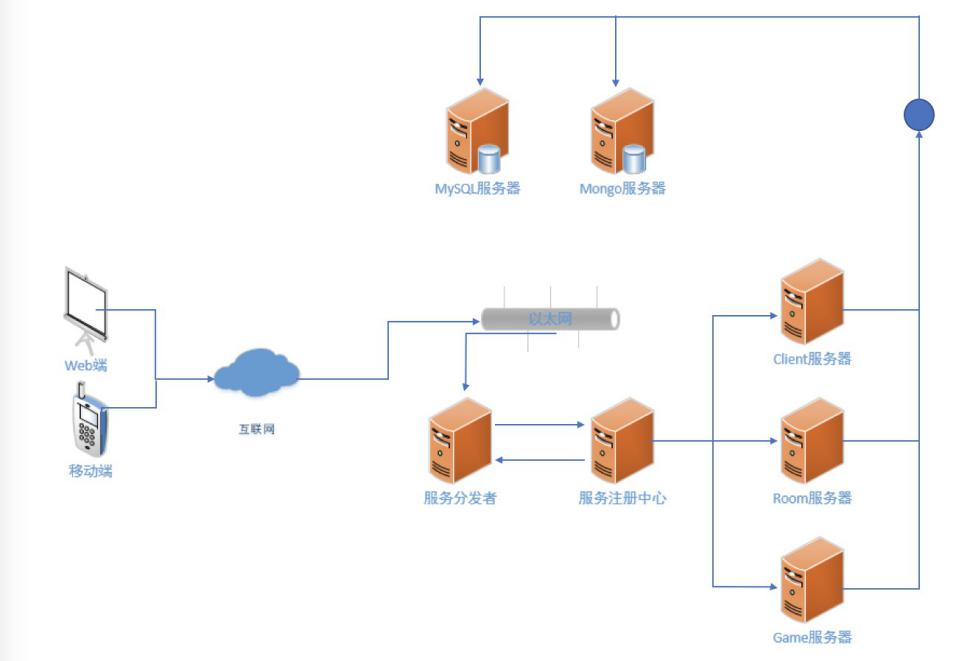
持久层主要由3个子包组成，如图所示。其中model包规定了数据格式和相互之间的外键依赖，repository包则利用SpringDataJPA自定义了对数据库的一些操作。SpringDataJPA则是由Spring官方提供的数据库访问包，里面包含了对于数据库的基本操作的实现。

1. 进程视图



用户在手机App上操作产生一系列请求，管理员在web端进行一系列请求。后端服务分发器接受这些请求后分发到对应的服务器user-client、room-clientgame-client，创建新的进程对其进行处理，并根据对业务的处理结果给出对数据库的操作请求，该请求又将交由数据库服务器进行处理。

1. 部署视图



5.1客户端

用户主要通过移动App来访问系统，管理员通过web网页访问系统支持的浏览器包括IE 9+和Chrome 60+，客户端和服务端的连接可以是局域网或互联网。

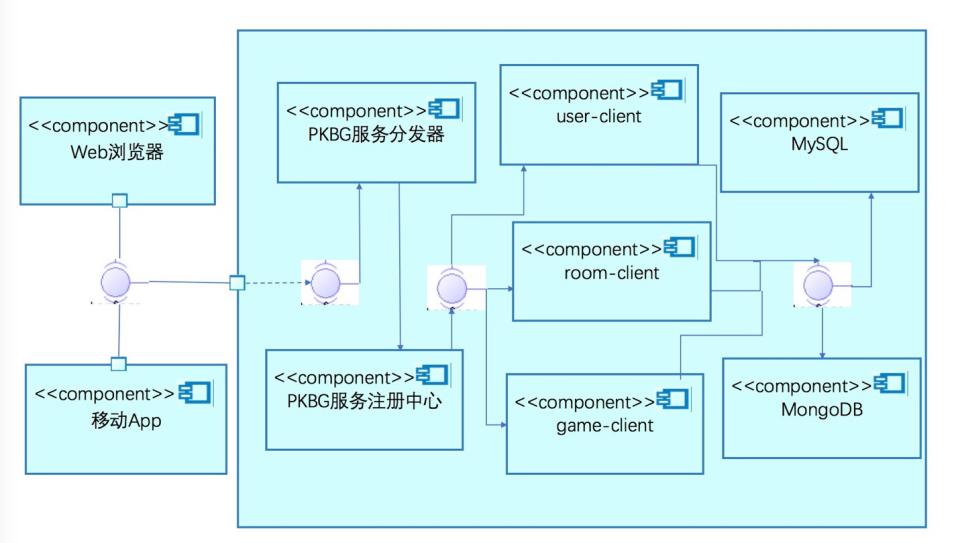
5.2 服务端

服务端采用微服务架构，采用SpringCloud通过consumer联通注册到服务注册中心的三个服务user-client、room-client以及game-client。系统部署在Tomcat9.0容器上，通过jdbc与数据服务器连接。

5.3 数据库服务器

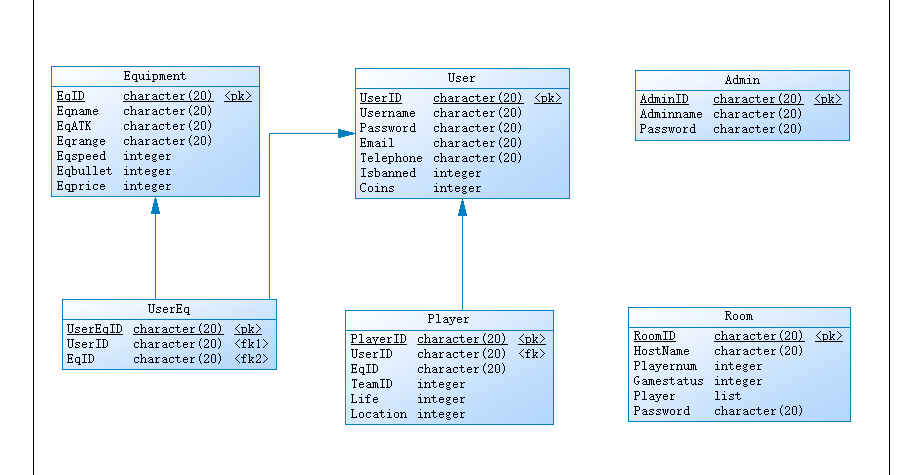
数据库服务器运行MySQL8.0数据库、MongoDB4.0数据库。

1. 实现视图



在实现过程中，我们将系统分为了客户端和服务端两大组件。客户端包括Web端与App端，这两个组件都通过请求与服务分发器联系。服务分发器联通注册中心，调用三个服务器处理业务逻辑，三个服务器与两种数据库联通。

1. 数据视图（可选）



1. 核心算法设计（可选）

[对系统中的核心算法进行设计。如果没有什么重要的算法，那么本节就为可选。]