**项目设计文档**

团队名称：咬文嚼字

文档更新记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 版本描述 | 日期 |
| V1.0 | 季灵杰&于钦 | 提供给成员阅读并评审 | 2015-3-10 |
| V1.1 | 朱静怡 | 新增排序与获得单个球队/球员信息接口 | 2015-3-13 |

引言：

|  |  |
| --- | --- |
| 内容和说明 |  |
| 1编写目的 | 本文档提供NBA球员球队查询系统的软件架构概览，采用若干架构视图描述系统的不同方面，以便表示构造系统所需要的重要架构决策 |
| 2对象与范围 | 本文档的读者是咬文嚼字团队内部的开发和管理人员，参考了RUP的《软件架构文档模板》，用于指导下一循环的代码开发和测试工作 |
| 3参考文献 | 《需求规格说明书》，咬文嚼字  《软件架构文档模板》，Rational Software Corporation,2002 |
| 4名词与术语 | 两双，场均数据，命中率，总篮板，效率，近五场的提升率，GmSc 效率值，真实投篮命中率，投篮效率，篮板率，进攻篮板率，防守篮板率，助攻率，抢断率，盖帽率，失误率，使用率，胜率，进攻回合，进攻效率，防守效率，进攻篮板效率，防守篮板效率，抢断效率，助攻率 |

系统的分层架构：

业务层

业务逻辑相关组件

表示层

界面部分，包括普通UI出错和配置界面

数据层

持久化数据；数据基本读/写逻辑

系统划分为以下3个层次。

1. 表示层：用于前台界面展示和配置的层次。
2. 业务层：包含业务控制和逻辑的层次。
3. 数据层：定义和存储系统中相关数据的层次。

管理员

用户

客户端

应用服务器

数据库服务器

访问层

业务层

数据层

系统可以部署在以下3个物理层次。

1. 访问层：用于用户访问系统的层次。
2. 业务层：部署业务控制和逻辑的层次。
3. 数据层：部署和存储系统中相关数据的层次。

系统的架构设计如下。

系统架构中的对象分为7类：

1. UI对象，负责处理系统数据的展现和用户的交互。
2. IController对象，控制器负责获取用户输入，并调用IService模块的服务。
3. IService对象，负责提供服务的抽象接口，获取从数据端组装好的数据。
4. ServiceImp对象，负责对于抽象接口的实现模块
5. IDTO（Data Transfer Object）对象，负责封装从IDAO获取的批量数据的接口。
6. IDAO（Data Access Object）对象，负责与数据库实体交互，获取数据。
7. Entity对象，该模块用来将从数据库中获取的数据封装成数据实体。

系统中的组件和组件接口：

表示层

UI

业务层

IService

IController

ServiceImp

IDTO

数据层

Entity

IDAO

|  |  |  |  |
| --- | --- | --- | --- |
| 接口ID | 连接组件 | 接口信息 | |
| 11 | 连接UI与IController | 语法 | Return(Response)  Interface(Request) |
| 前置条件 | 用户的输入正确 |
| 后置条件 | 处理控制组件处理请求并响应 |
| 不变量 | 用户请求信息 |
| 12 | 连接IController与IService | 语法 | Return(result)  Interface() |
| 前置条件 | 无 |
| 后置条件 | 对应的IService执行对应的业务逻辑 |
| 不变量 | 无 |
| 13 | 连接IService与IDTO | 语法 | Return(dataset)  Interface(command) |
| 前置条件 | 无 |
| 后置条件 | 对应的IDTO组件调用特定IDAO类获取数据层数据，并返回数据集 |
| 不变量 | 无 |
| 14 | 连接IDAO与Entity | 语法 | Return(data)  Interface(criteria) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | IDAO中的类将Entity对象写入数据库或从数据库中返回Entity对象 |
| 不变量 | 无 |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| IService.getSingleTeamInfo | 语法 | public teamInfo getSingleTeamInfo(teamInfo) |
| 前置条件 | 用户查看单个球队信息 |
| 后置条件 | 执行对应的业务逻辑 |
| IService.getSinglePlayerInfo | 语法 | public playerInfo getSinglePlayerInfo(playerInfo) |
| 前置条件 | 用户查看单个球员信息 |
| 后置条件 | 执行对应的业务逻辑 |
| IService. getTeamInfo | 语法 | public ArrayList<teamInfo> getTeamInfo() |
| 前置条件 | 无 |
| 后置条件 | 执行对应的业务逻辑 |
| IService.getPlayerInfo | 语法 | public ArrayList<playerInfo> getPlayerInfo(playerCondition pc) |
| 前置条件 | 用户的条件选择正确 |
| 后置条件 | 执行对应的业务逻辑 |
| IService.sortTeam | 语法 | Public ArrayList<teamInfoVO> sortTeam(ArrayList<teamInfoVO> original ,String s ) |
| 前置条件 | 用户的条件选择正确 |
| 后置条件 | 执行对应的业务逻辑 |
| IController.getPlayerCondition | 语法 | public playerCondition getPlayerCondition() |
| 前置条件 | 用户进行了条件选择并提交 |
| 后置条件 | 处理该组件并进行相应 |
| IDTO.getTeamOriginal | 语法 | public ArrayList<teamInfoPO> getTeamOriginal() |
| 前置条件 | 要求获得球队原始数据 |
| 后置条件 | 对应的IDTO组件调用特定IDAO类获取数据层数据，并返回数据集 |
| IDTO.getPlayerOriginal | 语法 | public ArrayList<playerInfoPO> getPlayerOriginal() |
| 前置条件 | 要求获得球员原始数据 |
| 后置条件 | 对应的IDTO组件调用特定IDAO类获取数据层数据，并返回数据集 |
| IDTO.getSingleTeamOriginal | 语法 | public teamInfoPO getSingleTeamOriginal() |
| 前置条件 | 要求获得单个球队原始数据 |
| 后置条件 | 对应的IDTO组件调用特定IDAO类获取数据层数据，并返回数据集 |
| IDTO.getSinglePlayerOriginal | 语法 | public playerInfoPO get SinglePlayerOriginal( () |
| 前置条件 | 要求获得单个球原始数据 |
| 后置条件 | 对应的IDTO组件调用特定IDAO类获取数据层数据，并返回数据集 |