**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 |
| V0.9 | 朱静怡 | 定义系统整体架构和部分接口 | 2015-3-28 |
|  |  |  |  |

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：

表示层

界面部分，包括普通UI出错和配置界面

业务层

业务逻辑相关组件

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。

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。

系统的体系结构逻辑设计方案如下：

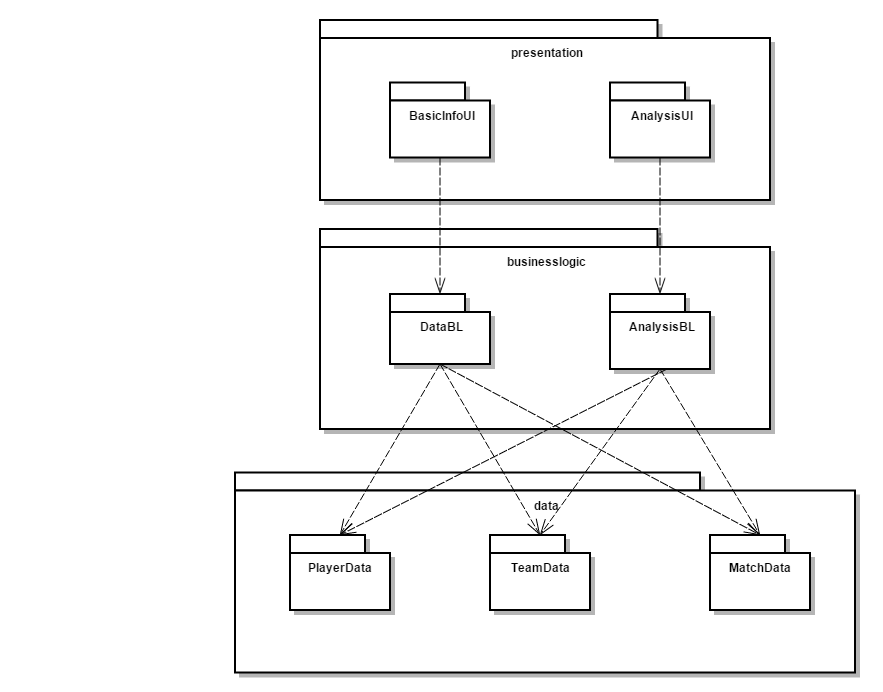


图1 系统体系结构逻辑设计方案

**表1 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接 口** | **服务调用方** | **服务提供方** |
| DataBLService  AnalysisBLService | 客户端展示层 | 客户端业务逻辑层 |
| PlayerDataService  TeamDataService  MatchDataService | 客户端业务逻辑层 | 服务器端数据层 |

该系统的开发包图如图2 所示：

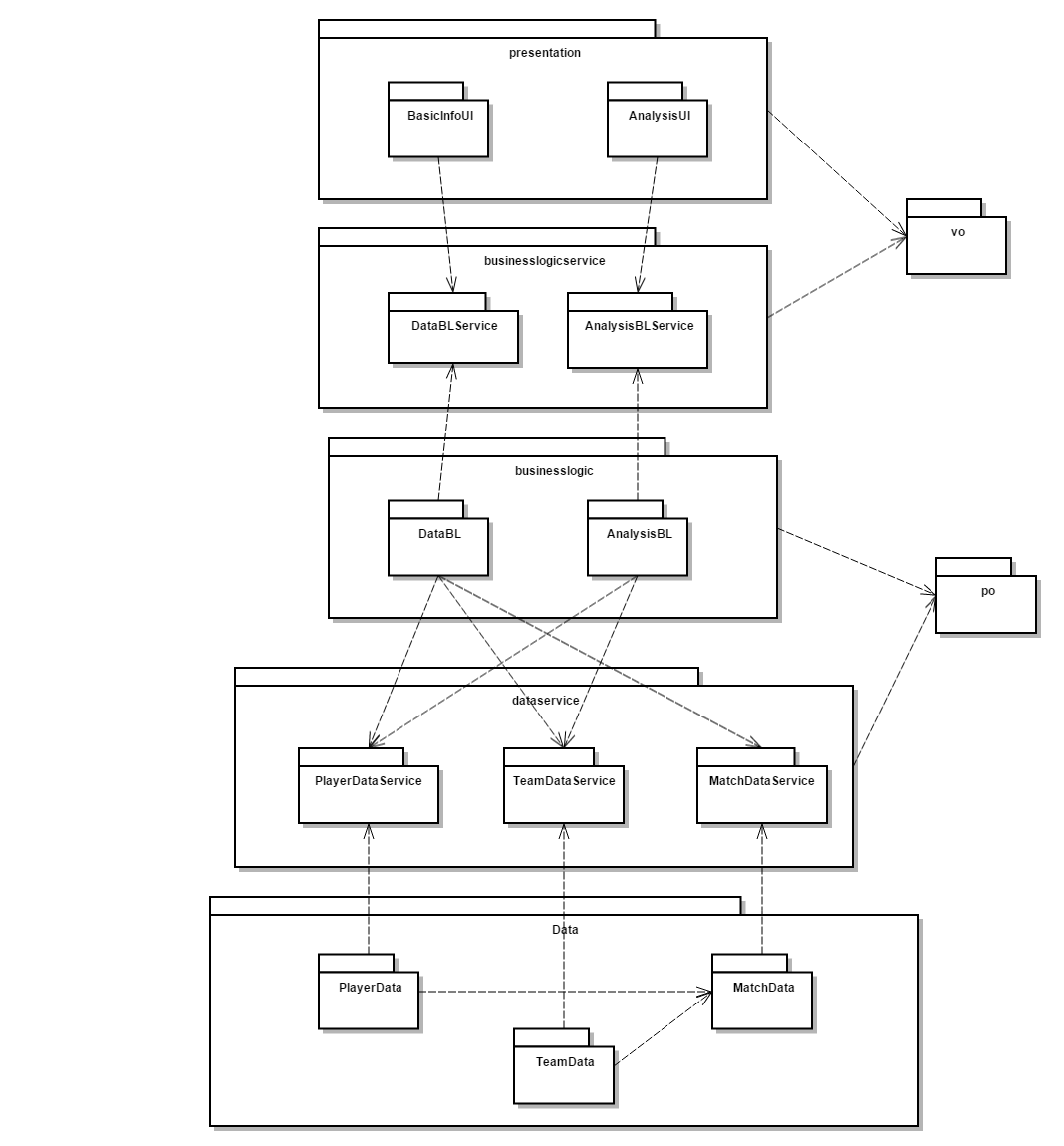


图2 系统开发包图

**业务逻辑层的分解**

业务逻辑层包括多个针对界面的业务逻辑处理对象。

1. 业务逻辑层模块的职责

业务逻辑层模块的职责如下表所示。

|  |  |
| --- | --- |
| **模 块** | **职 责** |
| DataBL | 负责实现所有基础信息显示所需要的服务 |
| AnalysisBL | 负责实现数据分析（包括筛选、排序、热点功能）的服务 |

2. 业务逻辑模块的接口规范

databl模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| DataBLService.getSinglePlayerInfo | 语法 | public PlayerVO getSinglePlayerInfo (String playerName) |
| 前置条件 | playerName符合输入规范 |
| 后置条件 | 查找是否存在相应的球员，返回该球员所有信息 |
| DataBLService.getSingleTeamInfo | 语法 | public TeamVO getSingleTeamInfo(String teamName) |
| 前置条件 | teamName符合输入规范 |
| 后置条件 | 查找是否存在相应的球队，返回该球队所有信息 |
| DataBLService.getSingleMatchInfo | 语法 | public MatchVO getSingleMatchInfo(String matchDate, String team) |
| 前置条件 | 比赛时间和球队符合输入规范 |
| 后置条件 | 查找是否存在相应比赛，返回该场比赛所有信息 |
| DataBLService.findMatchByPlayer | 语法 | public ArrayList<MatchVO> findMatchByPlayer (String playerName) |
| 前置条件 | playerName符合输入规范 |
| 后置条件 | 查找该球员参与的最近五场比赛，返回这五场比赛所有信息 |
| DataBLService.findMatchByTeam | 语法 | public ArrayList<MatchVO> findMatchByTeam (String teamName) |
| 前置条件 | teamName符合输入规范 |
| 后置条件 | 查找该球队的最近五场比赛，返回这五场比赛所有信息 |
| DataBLService.findMatchByDate | 语法 | Public ArrayList<MatchVO> findMatchByDate (String matchDate) |
| 前置条件 | 比赛时间符合输入规范（因为不知道数据里的最后一天是哪天matchDate可以设为today然后就返回数据里最后一天的所有比赛信息） |
| 后置条件 | 查找是否存在相应比赛，返回当日比赛所有信息 |
| DataBLService.getAllPlayerInfo | 语法 | public ArrayList<PlayerVO> getAllPlayerInfo () |
| 前置条件 | 无 |
| 后置条件 | 返回所有球员信息 |
| DataBLService.getAllTeamInfo | 语法 | public ArrayList<TeamVO> getAllTeamInfo() |
| 前置条件 | 无 |
| 后置条件 | 返回所有球队信息 |
|  | | |
| **需要的服务（需接口）** | | |
| DataService.getPlayerInfo | 语法 | public ArrayList<PlayerPO > getPlayerInfo () |
| 前置条件 | 无 |
| 后置条件 | 返回所有球员信息 |
| DataService.getTeamInfo | 语法 | Public ArrayList<TeamPO> getTeamInfo() |
| 前置条件 | 无 |
| 后置条件 | 返回所有球队信息 |
| DataService.getMatchInfo | 语法 | public ArrayList<MatchPO> getMatchInfo() |
| 前置条件 | 无 |
| 后置条件 | 返回所有比赛信息 |

analysisbl模块的接口规范

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **提供的服务（供接口）** | | | | | |
| AnalysisBLService.sortPlayer | | 语法 | | | public ArrayList<PlayerVO> sortPlayer (ArrayList<PlayerVO> playerlist, String keyword, sortType type); |
| 前置条件 | | | 用户输入合法待排序球员列表、排序关键词和排序方式 |
| 后置条件 | | | 返回排序后的球员列表 |
| AnalysisBLService.sortTeam | | 语法 | | | public ArrayList<TeamVO> sortTeam(ArrayList<TeamVO> teamlist, String keyword, sortType type); |
| 前置条件 | | | 用户输入合法待排序球队列表、排序关键词和排序方式 |
| 后置条件 | | | 返回排序后的球队列表 |
| AnalysisBLService.getTopFiftyPlayer | | 语法 | | | public ArrayList<PlayerVO> getTopFiftyPlayer (PlayerPosition position, playerPartition partition, String keyword); |
| 前置条件 | | | 用户合法球员位置、球员联盟和排序依据 |
| 后置条件 | | | 返回筛选得到的前50名球员 |
| AnalysisBLService.getTodayHotSpotPlayer | | 语法 | | | public ArrayList<PlayerVO> getTodayHotSpotPlayer (String keyword); |
| 前置条件 | | | 用户输入合法筛选条件 |
| 后置条件 | | | 返回今日参加比赛的前五名热点球员 |
| AnalysisBLService.getSeasonHotSpotPlayer | | 语法 | | | public ArrayList<PlayerVO> getSeasonHotSpotPlayer (String keyword); |
| 前置条件 | | | 用户输入合法筛选条件 |
| 后置条件 | | | 返回所有参加过比赛的前五名热点球员 |
| AnalysisBLService.getSeasonHotSpotTeam | | 语法 | | | public ArrayList<TeamVO> getSeasonHotSpotTeam (String keyword); |
| 前置条件 | | | 用户输入合法筛选条件 |
| 后置条件 | | | 返回所有参加过比赛的前五名热点球队 |
| AnalysisBLService.getProgressivePlayer | | 语法 | | | public ArrayList<PlayerVO> getProgressivePlayer (String keyword); |
| 前置条件 | | | 用户输入合法筛选条件 |
| 后置条件 | | | 返回符合相应条件的五名进步最快球员 |
| **需要的服务（需接口）** | | | | | |
| **服务名** | | | | **服务** | |
| DataService.getPlayerInfo | 语法 | | public ArrayList<PlayerPO > getPlayerInfo () | | |
| 前置条件 | | 无 | | |
| 后置条件 | | 返回所有球员信息 | | |
| DataService.getTeamInfo | 语法 | | Public ArrayList<TeamPO> getTeamInfo() | | |
| 前置条件 | | 无 | | |
| 后置条件 | | 返回所有球队信息 | | |
| DataService.getMatchInfo | 语法 | | public ArrayList<MatchPO> getMatchInfo() | | |
| 前置条件 | | 无 | | |
| 后置条件 | | 返回所有比赛信息 | | |

**数据层模块的分解**

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。数据层模块的具体描述如图12所示。

图12数据层模块的描述

1. 数据层模块的职责

数据层模块的职责如下表所示。

|  |  |
| --- | --- |
| **模 块** | **职 责** |
| PlayerData | 持久化数据库的接口，提供球员数据载入、保存、增、删、改、查等服务 |
| MatchData | 持久化数据库的接口，提供比赛数据载入、保存、增、删、改、查等服务 |
| TeamData | 持久化数据库的接口，提供球队数据载入、保存、增、删、改、查等服务 |

2. 数据层模块的接口规范

PlayerData模块接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务(供接口)** | | |
| PlayerDataService.getSinglePlayerInfo | 语法 | Public playerPO getSinglePlayerInfo(String playerName) |
| 前置条件 | 输入合法球员姓名 |
| 后置条件 | 查找并返回该球员信息 |
| PlayerDataService.getAllPlayerInfo | 语法 | Public ArrayList<playerPO> getAllPlayerInfo() |
| 前置条件 | 查询所有球员信息 |
| 后置条件 | 返回所有球员信息 |
| PlayerDataService.getTodayPlayerInfo | 语法 | Public ArrayList<todayPlayerPO> getTodayPlayerInfo(String date)(如：2015-4-8) |
| 前置条件 | 输入合理查询比赛日期 |
| 后置条件 | 返回当日比赛球员信息 |

TeamData模块接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务(供接口)** | | |
| TeamDataService.getSingleTeamInfo | 语法 | Public TeamPO getSingleTeamInfo(String teamName) |
| 前置条件 | 输入合法球队名 |
| 后置条件 | 查找并返回该球队信息 |
| TeamDataService.getAllTeamInfo | 语法 | Public ArrayList<teamPO> getAllTeamInfo() |
| 前置条件 | 查询所有球队信息 |
| 后置条件 | 返回所有球队信息 |

MatchData模块接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务(供接口)** | | |
| MatchDataService.getSingleMatchInfo | 语法 | Public MatchPO getSingleMatchInfo(String teamName1,String teamName2,string date) |
| 前置条件 | 输入合法两只球队名及比赛日期 |
| 后置条件 | 查找并返回该场比赛信息 |
| MatchDataService.getAllMatchInfo | 语法 | Public ArrayList<MatchPO> getAllMatchInfo() |
| 前置条件 | 无 |
| 后置条件 | 返回所有比赛信息 |
| MatchDataService.getTeamRecentMatchInfo | 语法 | Public ArrayList<MatchPO> getTeamRecentMatchInfo(int number,String teamName) |
| 前置条件 | 输入合法场数及球队名 |
| 后置条件 | 返回筛选的比赛信息 |
| MatchDataService.getPlayerRecentMatchInfo | 语法 | Public ArrayList<MatchPO> getPlayerRecentMatchInfo(int number,String playerName) |
| 前置条件 | 输入合法场数及球员名 |
| 后置条件 | 返回筛选的比赛信息 |