# 软件架构设计文档

团队名称：TBD

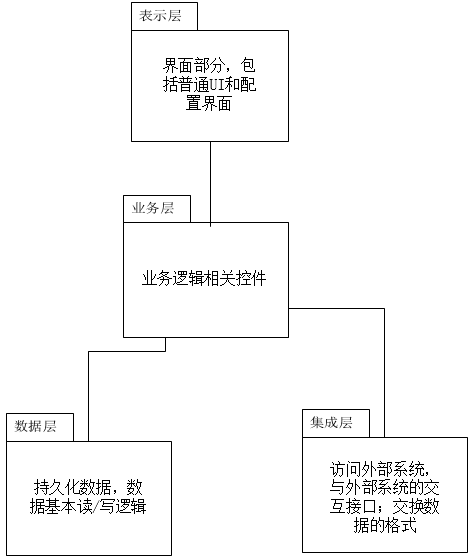
## 文档更新记录：

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **作者** | **版本描述（应该说明做了哪些修改及修改的标识等）** | **日期** |
| V1.0 | 陈元亮 | 新建文档，发布到服务器SVN上 | 2015/05/05 |
| V2.0 | 李嘉麒 | 迭代三更新 | 2015/05/15 |

## 引言：

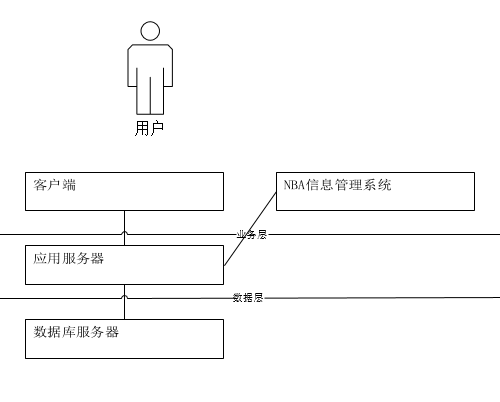
|  |  |
| --- | --- |
| 内容和说明 |  |
| 1.编写目的 | 本文档提供NBA数据分析系统的软件架构概览，采用若干架构视图描述系统的不同方面，以便表示构造系统所需要的重要架构决策。 |
| 2.对象和范围 | 本文档的读者是( TBD )团队内部的开发和管理人员，参考了RUP的《软件架构文档模板》，用于指导下一循环的代码开发和测试工作。 |
| 3.参考文献 | 《软件需求规格说明书》  《软件架构文档模板》 |
| 4.名词和术语 | UI BL Data |

## 系统的分层架构:



系统划分为以下4个逻辑层次。

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：定义和存储系统相关数据的层次。
4. 集成层：定义和集成与外部系统交互接口的层次。



系统可以部署在下面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：部署和存储系统中的相关数据的层次

## 系统架构设计如下：

系统架构中的对象分为5类：

1. UI对象，负责处理处理系统数据的展现与用户之间的人机交互
2. BLService对象：负责提供UI服务的抽象接口，获取从数据端组装好的数据。
3. BLservice对象：负责实现BLservice所申明的接口，分配任务给其他controller
4. DataService对象:负责提供给BL层服务的抽象接口，从数据库中读取数据
5. DataController对象：负责实现DataService所申明的接口

系统中的组件和组件接口：

表示层

UI

业务层

BL

BLService

数据层

DataService

Data

## 5. 接口规范

### 5.1 BL模块接口规范

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **提供的服务（供接口）** | | | | | |
| **BLservice.** **getAlliancePlayerAverageData ()** | | 语法 | | Double[] [getAlliancePlayerAverageData](file:///C:\Users\zgw\Documents\Tencent%20Files\719276265\FileRecv\doc\BLservice\BLservice.html#getAlliancePlayerAverageData-java.lang.String-boolean-)(java.lang.String season, boolean isPlayOff); | |
| 前置条件 | | 服务器已开启 | |
|  | | 后置条件 | | 调用BLservice的[getAlliancePlayerAverageData](file:///C:\\Users\\zgw\\Documents\\Tencent%20Files\\719276265\\FileRecv\\doc\\BLservice\\BLservice.html" \l "getAlliancePlayerAverageData-java.lang.String-boolean-)方法，并返回联盟平均水平 | |
| **BLservice.getALLPlayer()** | | 语法 | | ArrayList<Playervo> [getAllPlayers](file:///C:\Users\zgw\Documents\Tencent%20Files\719276265\FileRecv\doc\BLservice\BLservice.html#getAllPlayers-java.lang.String-boolean-)(java.lang.String Season, boolean isPlayOff); | |
| 前置条件 | | 服务器已开启 | |
| 后置条件 | | 调用BLservice的**getALLPlayers**方法，并返回所有的Players全部信息 | |
| **BLservice.getALLTeams()** | | 语法 | | Public ArrayList<Teamvo>  **getALLTeams** (); | |
| 前置条件 | | 服务器已开启 | |
| 后置条件 | | 调用BLservice的**getALLTeams**方法，并返回所有的Teams全部信息 | |
| **BLservice. getHotZone ()** | | 语法 | | public HotZonevo **getHotZone** (); | |
| 前置条件 | | 服务器已开启 | |
| 后置条件 | | 调用BLservice的**getHotZone**方法，得到某个球队某赛季常规赛或季后赛的热区数据 | |
| **BLservice.** **getLiveEvent ()** | | 语法 | | Public ArrayList<[EventVo](file:///C:\\Users\\zgw\\Documents\\Tencent%20Files\\719276265\\FileRecv\\doc\\vo\\EventVo.html" \o "vo中的类)> **getLiveEvent();** | |
| 前置条件 | | 服务器已开启，调用initLive()方法 | |
| 后置条件 | | 得到当前正在直播的即时直播事件（即为文字直播时所见到的描述） | |
| **BLservice.** **getMatchSimpleInfo ()** | | 语法 | | ArrayList<MatchSimpleInfovo> getMatchSimpleInfo(java.lang.String teamName, java.lang.String season) **;** | |
| 前置条件 | | 服务器开启 | |
| 后置条件 | | 得到一场比赛的简略信息 | |
| **BLservice.** **getPlayerById ()** | | 语法 | | Public getPlayerById(String Id) | |
| 前置条件 | | 服务器开启 | |
| 后置条件 | | 根据某个球员的ID，得到该球员全部历史记录 | |
| **BLservice. getPlayerById ()** | | 语法 | | Public Playervo [getPlayerById](file:///C:\Users\zgw\Documents\Tencent%20Files\719276265\FileRecv\doc\BLservice\BLservice.html#getPlayerById-java.lang.String-java.lang.String-boolean-)(String Id,String season, boolean isplayoff) | |
| 前置条件 | | 服务器已开启 | |
| 后置条件 | | 根据某个球员的ID，得到该球员某个赛季的属性 | |
| **BLservice. getPlayerByNameAndTeam( ()** | | 语法 | | Public Playervo [getPlayerByNameAndTeam](file:///C:\Users\zgw\Documents\Tencent%20Files\719276265\FileRecv\doc\BLservice\BLservice.html#getPlayerByNameAndTeam-java.lang.String-)(String playerName) | |
|  | 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 根据球员姓名得到球员信息 | |
| **BLservice. getSeasonKingPlayer ()** | 语法 | | | **ArrayList<Playervo> getSeasonKingPlayer(String transferField, int number, String Season, boolean isplayoff)** | |
|  | 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 得到数据王 | |
| **BLservice.** **getTeamAbility()** | 语法 | | | double[] getTeamAbility(java.lang.String teamNameEn) | |
|  | | | | 前置条件 | 服务器已开启 |
| 后置条件 | 得到球队能力 |
| **BLservice.** **getTeamByPlayerId()** | 语法 | | | Public Teamvo getTeamByPlayerId(String playerId, String season, boolean isPlayOff ) | |
|  | 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 根据球员ID得到球队信息 | |
| **BLservice.** **getTeamByTeamName()** | 语法 | | | Public getTeamByTeamName(String teamNameEn, String season, boolean isPlayOff) | |
| 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 返回球队名称对应的球队信息 | |
| **BLservice.** **getTeamWithLatest10Data** | 语法 | | | Teamvo getTeamWithLatest10Data(java.lang.String teamNameEn, java.lang.String season, boolean isPlayOff) | |
| 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 返回最近十场比赛信息 | |
| **BLservice.** **getTodayKingPlayer()** | 语法 | | | ArrayList<TodayPlayervo> getTodayKingPlayer(String Date, String transferField, int number) | |
| 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 返回指定日期的数据王 | |
| **BLservice.** **initNBALive()** | 语法 | | | Public Team **getTeamByName(String teamName)** | |
| 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 直播初始化 | |
| **BLservice.** getTodayMatches **()** | 语法 | | | Collection<? extends Matchvo> getTodayMatches(java.lang.String date) | |
| 前置条件 | | | 服务器已开启 | |
| 后置条件 | | | 获取指定日期的比赛 | |
| **需要的服务（需接口）** | | | | | |
| **DataService. getALLPlayers** | | | 从数据端中得到所有的球员的部分信息 | | |
| **DataService. getALLTeams** | | | 从数据端中得到所有的球队部分信息 | | |
| **DataService. getALLMatches** | | | 从数据端中得到所有的比赛部分信息 | | |

### 5.2 Data模块接口规范

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MatchDataService提供的服务（供接口）** | | | | |
| **MatchItemReader.** **getMatchItemById** **()** | 语法 | | public ArrayList<matchItem> getMatchItemById(String id); | |
| 前置条件 | | 已创建一个**MatchItemReader**对象 | |
| 后置条件 | | 根据指定的比赛Id返回matchItem | |
| **MatchBLcontroller.** getMatchItemByPlayerId **()** | 语法 | | public ArrayList<matchItem> getMatchItemByPlayerId(String playerId) | |
| 前置条件 | | 已创建一个**MatchItemReader**对象 | |
|  | | 后置条件 | | 根据球员Id返回matchItem |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MatchReader.** **getgetAllMatch** **()** | 语法 | | public ArrayList<String> getAllMatch(); | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | | |
| 后置条件 | | 返回所有比赛信息 | | |
| **MatchReader.** **getMatchesBySeason** **()** | 语法 | | public ArrayList<match> getMatchesBySeason(String season, boolean isplayoff); | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | | |
| 后置条件 | | 根据指定的赛季返回match | | |
| **MatchReader.** **getMatchesBySeason** **()** | 语法 | | public ArrayList<matchItem> getMatchItemById(String id); | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | | |
| 后置条件 | | 根据指定的比赛Id返回matchItem | | |
| **MatchReader.** getMatchesByTime **()** | 语法 | | public ArrayList<matchItem> getMatchItemById(String id); | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | | |
| 后置条件 | | 根据指定的比赛Id返回matchItem | | |
| **MatchReader.** **getMatchesById** **()** | 语法 | | public ArrayList<String> getMatchesByTime(String time) | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | | |
| 后置条件 | | 根据指定的时间返回match | | |
| **MatchReader.** getMatchesByTeam **()** | 语法 | | Public ArrayList<String> getMatchesByTeam(String teamNameEn, String season, boolean isPlayOff, int number); | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | | |
| 后置条件 | | 根据指定的赛季球队信息返回match | | |
| **MatchBLcontroller.** getMatchSimpleByTeam **()** | 语法 | | public ArrayList<String> getMatchesByTeam(String teamNameEn, String season, boolean isPlayOff, int number) | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | |
|  | | 后置条件 | | | 返回球队指定赛季比赛简略信息 |
| **MatchReader.** **getMatchSimpleByTeam** **()** | 语法 | | public ArrayList<match> getMatchSimpleByTeam(String teamName, String season); | | |
| 前置条件 | | 已创建一个**MatchReader**对象 | | |
| 后置条件 | | 返回球队比赛简略信息 | | |

|  |
| --- |
| **NBALiveDataService提供的服务（供接口）** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NBALiveReader.** **getAllNBALivePlayer** **()** | 语法 | | | ArrayList<NBALivePlayer> getAllNBALivePlayer; |
| 前置条件 | | | 已创建一个NBALiveReader对象 |
| 后置条件 | | | 得到所有直播的球员信息 |
| **NBALiveReader.** getAllNBALiveTeam **()** | 语法 | | | public ArrayList<NBALiveTeam> getAllNBALiveTeam() |
| 前置条件 | | | 已创建一个NBALiveReader对象 |
|  | | 后置条件 | 得到所有的直播球队信息 | |
| **pointsItemReader.** getpointsItemById **()** | 语法 | | | public ArrayList<pointsItem> getpointsItemById(String id) |
| 前置条件 | | | 已创建一个**pointsItemReader**对象 |
| 后置条件 | | | 得到所有球员得分的信息 |

|  |
| --- |
| **PlayerDataService提供的服务（供接口）** |

|  |  |  |
| --- | --- | --- |
| **PlayerItemReader.** getPlayerItemById **()** | 语法 | public ArrayList<playerItem> getPlayerItemById(String id) |
| 前置条件 | 已创建一个PlayerItemReader对象 |
| 后置条件 | 根据球员id得到球员条目 |
| **PlayerItemReader.** getPlayerItemBySeason **()** | 语法 | public ArrayList<playerItem> getPlayerItemBySeason(String season) |
| 前置条件 | 已创建一个PlayerItemReader对象 |
| **PlayerNameReader.** getpointsItemById **()** | 语法 | public ArrayList<PlayerName> getAllPlayerName() |
| 前置条件 | 已创建一个PlayerNameReader对象 |
| 后置条件 | 得到所有球员姓名 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **PlayerReader.** **getPlayerById** **()** | 语法 | public player getPlayerById(String id) |
| 前置条件 | 已创建一个PlayerReader对象 |
| 后置条件 | 根据球员id得到球员条目 |
| **PlayerReader.** **getAllPlayer** **()** | 语法 | public ArrayList<player> getAllPlayer() |
| 前置条件 | 已创建一个PlayerReader对象 |
| **PlayerReader.** getPlayersBySeason **()** | 语法 | public ArrayList<player> getPlayersBySeason(String season, boolean isPlayoff) | |
| 前置条件 | 已创建一个**PlayerReader**对象 | |
| 后置条件 | 根据赛季信息得到所有球员 | |

|  |  |  |
| --- | --- | --- |
| **PlayerReader.** **getPlayerByNameAndTeam** **()** | 语法 | public player getPlayerByNameAndTeam(String playerName, String teamName) |
| 前置条件 | 已创建一个**PlayerReader**对象 |
| 后置条件 | 根据球员和球队名称得到球员信息 |

|  |  |  |
| --- | --- | --- |
| **PlayerItemReader.** getPlayerItemById **()** | 语法 | public ArrayList<playerItem> getPlayerItemById(String id) |
| 前置条件 | 已创建一个PlayerItemReader对象 |
| 后置条件 | 根据球员id得到球员条目 |
| **PlayerItemReader.** getPlayerItemBySeason **()** | 语法 | public ArrayList<playerItem> getPlayerItemBySeason(String season) |
| 前置条件 | 已创建一个PlayerItemReader对象 |
| **TeamDataService提供的服务（供接口）** | | |

|  |  |  |
| --- | --- | --- |
| **HotZoneData.** getHotZones **()** | 语法 | public ArrayList<HotZone> getHotZones(HotZone hotZone) |
| 前置条件 | 已创建一个**HotZoneData**对象 |
| 后置条件 |  |
| **TeamNameListData.** **getAllPlayer** **()** | 语法 | public ArrayList<TeamListItem> getTeamList(TeamListItem item) |
| 前置条件 | 已创建一个PlayerReader对象 |
| **TeamRecordItemData.** **getTeamRecords** **()** | 语法 | public ArrayList<player> getPlayersBySeason(String season, boolean isPlayoff) | |
| 前置条件 | 已创建一个**PlayerReader**对象 | |
| 后置条件 | 根据赛季信息得到球队所有记录 | |

|  |  |  |
| --- | --- | --- |
| **TeamRecordItemData.** addTeamRecordItem **()** | 语法 | public player getPlayerByNameAndTeam(String playerName, String teamName) |
| 前置条件 | 已创建一个**PlayerReader**对象 |
| 后置条件 | 增加球队记录 |

|  |  |  |
| --- | --- | --- |
| **HotZoneData.** **addHotZone** **()** | 语法 | public boolean addHotZone(HotZone hotZone) |
| 前置条件 | 已创建一个HotZoneData对象 |
| 后置条件 | 增加热区信息 |

|  |  |  |
| --- | --- | --- |
| **TeamNameListData.** addTeamNameListItem **()** | 语法 | public boolean addTeamNameListItem(TeamListItem item) |
| 前置条件 | 已创建一个**TeamNameListData**对象 |
| 后置条件 | 增加球队名称 |