****

ProQuant

软件架构设计文档

V1.0

**组名：Innovator**

**成员：谢振宇、伍俊、王凡、杨凯**

**2017-6-10**

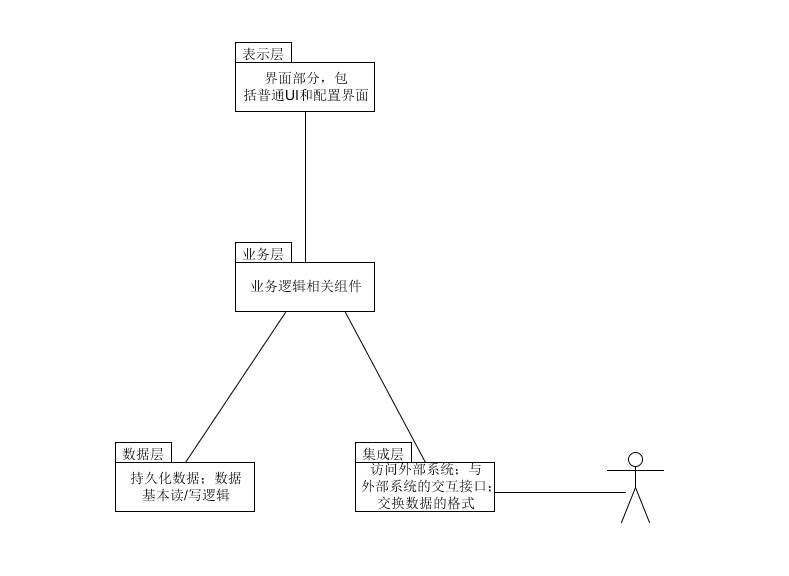
# 文档更新记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 版本描述（应该说明做了哪些修改及修改的标示等） | 日期 |
| V1.0 | 王凡 | 迭代一基本架构 | 2016/3/4 |
| V1.1 | 杨凯 | 迭代一逻辑层分解和分层细化 | 2016/3/4 |
| V1.2 | 王凡 | 迭代一数据层完善 | 2016/3/6 |
| V2.1 | 王凡 | 迭代二数据层 | 2016/3/24 |
| V2.2 | 王凡 | 迭代二功能补充 | 2016/4/10 |
| V2.3 | 杨凯 | 迭代二逻辑层 | 2016/4/15 |
| V3.1 | 王凡 | 迭代三数据层及部分逻辑层补充 | 2016/6/10 |
| V3.2 | 杨凯 | 迭代三逻辑层，web层补充 | 2016/6/10 |

# 引言：

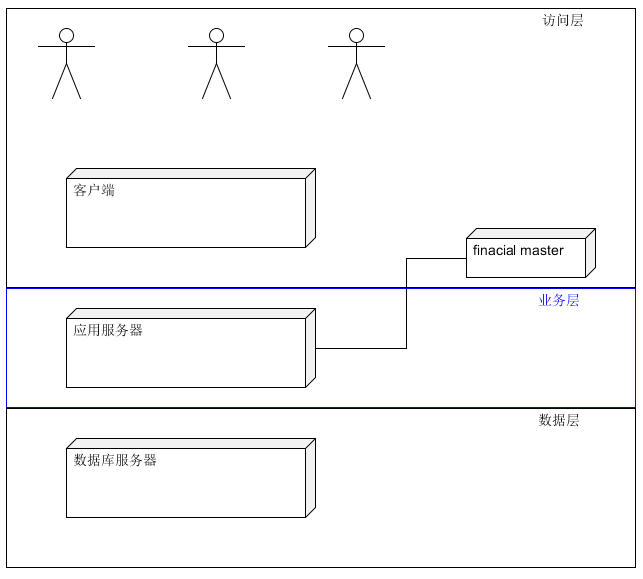
|  |  |
| --- | --- |
| 内容说明 |  |
| 1 编写目的  （说明整个文档所有达到的目标） | 本文档提供师生交流系统的软件架构概览，采用若干架构试图描述系统的不同方面，以便表示构造系统所需要的重要架构决策 |
| 2 对象与范围  （说明整个文档的内容范围和针对的读者对象） | 本文档的读者是干票大的团队内部的开发和管理人员以及大三测试人员，参考了RUP的《软件架构文档模板》，用于指导下一循环的代码开法和测试工作 |
| 3 参考文献  （说明文档中主要的所需所用） | 《软件需求规格说明书》，  《软件架构文档模板》，Rational Software Corporation |
| 4 名词与术语  （说明文档汇总常用的技术缩略和相关词条） | 无 |

# 系统的分层架构：



系统划分为以下4个逻辑层次。

1. 展示层：用于浏览器界面展示和配置的层次。
2. 逻辑层：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。
4. 集成层：定义和集成与外部系统交互接口的层次。



# 体系结构设计

## 系统的部署在以下3个物理层次

访问层：用于用户访问系统的层次。

业务层：部署业务控制和逻辑的层次

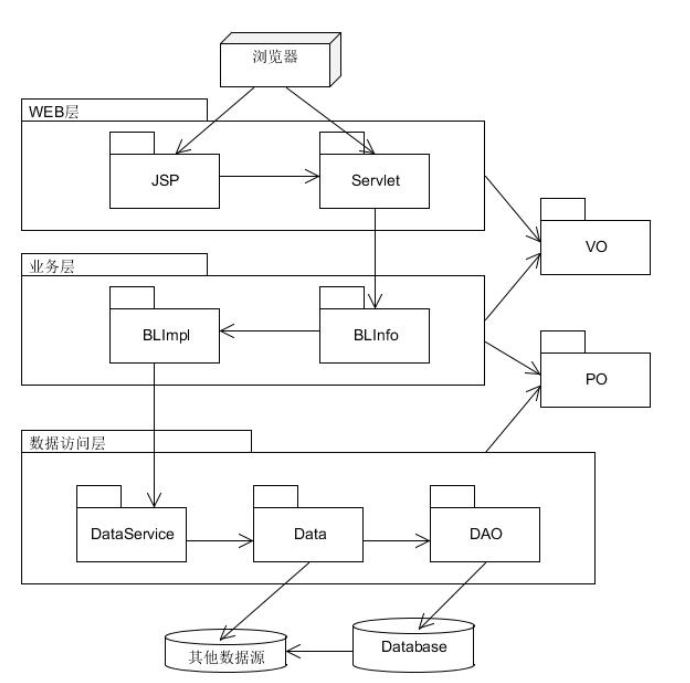
数据层：部署和存储系统中相关数据的层次

## 系统的架构设计如下。

系统架构中的对象分为12类

1. Js对象，负责局部组件显示
2. Css对象，负责组件属性设置
3. JQuery对象，负责异步与servlet交互
4. Jsp对象，负责总体页面显示以及部分与servlet交互
5. BLInfo对象，负责向servlet提供抽象接口
6. BLImpl对象，负责对BLInfo抽象接口的实现
7. DataService对象，负责向逻辑层提供抽象接口
8. Data对象，负责对DataService抽象接口的实现
9. Entity对象，该对象将用来从数据库中获取的数据封装成数据实体
10. PO对象、POJO对象，用于将数据对象打包成PO进行数据传递
11. Dao对象，负责对data业务对象提供数据库查询和更新的底层实现

## 系统中的组件和组件接口：



## 业务逻辑层的分解

### 4.1业务逻辑层模块的职责

**业务逻辑层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| BenchImpl | 负责处理大盘的数据 |
| businessImpl | 负责返回行业与行业详情所需数据 |
| serchInfo | 负责处理并返回搜索框所需数据 |
| stockImpl | 负责处理个股模块所需的数据 |
| userImpl | 负责处理用户登录注册与个人信息的数据 |
| StrategyHandle | 负责处理计算模拟策略时用到的数据 |

### 4.2业务逻辑层模块的接口规范

#### 4.2.1股票查询模块

##### 4.2.1.1 StockListBL模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockListBL.getStockList | 语法 | public String[][] getStockList() |
| 前置条件 | 启动一个股票查询回合 |
| 后置条件 | 返回预选股票的列表，用二维数组展示 |
| StockListBL.updateStockList | 语法 | public String[][] updateStockList(String key) |
| 前置条件 | 用户在搜索框输入用来筛选的关键字 |
| 后置条件 | 返回筛选过后的列表 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StockListData getStockList() | 从数据层得到股票的列表 | |

##### 4.2.1.2 StockMessageBL模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockMessagetBL.getStockMessage | 语法 | public StockVO getStockMessage(String id) |
| 前置条件 | 从股票列表中选择某支股票 |
| 后置条件 | 返回所选股票的详细信息，打包成StockPO |
| StockMessageBL.updateStockMessage | 语法 | public StockVO updateStockList(String id,String startData,String overData) |
| 前置条件 | 用户选择正确的时间段 |
| 后置条件 | 根据用户输入的时间段返回筛选过后的列表 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StockMessageData getStockmessage(String id) | 从数据层得到指定id股票的详细信息（默认为最近三十天的历史数据） | |
| StockMessageData getStockmessage(String id,String startData,String overData) | 从数据层得到指定id，指定时间段股票的详细信息 | |

##### 4.2.1.3 StockItemBL模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockItemBL. getRank(String item) | 语法 | public ArrayList<StockItemVO> getRank(String item) |
| 前置条件 | 完成bl的初始化 |
| 后置条件 | 返回对应项目的排序 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StockData. getStatisitcOfStock（String id,String StartDay,String EndDay) | 从数据层得到指定id和起止日期得到股票的详细信息 | |
| manageStockData. getCodeOfStock(); | 从数据层得到关注的股票列表 | |

##### 4.2.1.4 StockSearchBL模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockSearchBL. getList(String key) | 语法 | public String[][] getList(String key) |
| 前置条件 | 完成bl的初始化 |
| 后置条件 | 根据关键字返回包含关键字的股票列表 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StockData. getCodeName(int year,String exchange); | 根据年份和交易所代号获得当年所有股票 | |

#### 4.2.2大盘模块

##### 4.2.2.1 StockMarketBL模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockMarketBL.getStockMarket | 语法 | public StockMarketVO getStockMarket(String key,Sting data) |
| 前置条件 | 启动一个股票查询回合 |
| 后置条件 | 根据关键字和时间返回当前最新的大盘信息 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StockMessageData getStockMarket(String key,Sting data) | 得到指定证券指数指定时间的大盘信息 | |

##### 4.2.2.2 StockKLineBL模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockKLineBL. update | 语法 | public ManageState update() |
| 前置条件 | 启动软件 |
| 后置条件 | 更新本地周k和月k的缓存数据 |
| StockKLineBL. getData | 语法 | public StockMarketVO getData(marketKline\_enum k) |
| 前置条件 | 完成本地缓存的更新 |
| 后置条件 | 根据k线图类型返回对应的数据 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| BenchKLineData. update() | 更新本地缓存数据 | |
| BenchKLineData.getStatisticData(String kind) | 根据不同k线图类型返回对应本地的缓存数据 | |
| StockMarketBL. getStockMarket(String exchange, date\_enum date); | 得到指定证券指数指定时间的大盘信息 | |
| InitFactory.getStockMarketBL() | 通过初始化工厂得到StockMarketBL | |

#### 4.2.3股票对比模块

##### 4.2.3.1 StockContrastBL模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockContrastBL. getData | 语法 | public double[] getData(String id) |
| 前置条件 | 完成该BL的初始化 |
| 后置条件 | 根据股票编号得到雷达图需要显示的数据 |
| StockContrastBL. getList | 语法 | public String[] getList() |
| 前置条件 | 完成该BL的初始化 |
| 后置条件 | 返回关注的股票名称列表 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StockData. getStatisitcOfStock（String id,String StartDay,String EndDay) | 从数据层得到指定id和起止日期得到股票的详细信息 | |
| manageStockData. getCodeOfStock(); | 从数据层得到关注的股票列表 | |

#### 4.2.4股票管理模块

##### 4.2.4.1 ManageStockBL模块

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| ManageStockBL. addStock | 语法 | | public ManageState addStock(String id) |
| 前置条件 | | 无 |
| 后置条件 | | 更新工厂中初始化过的BL |
| ManageStockBL. deleteStock | 语法 | | public ManageState deleteStock(String id) |
| 前置条件 | | 无 |
| 后置条件 | | 更新工厂中初始化过的BL |
| ManageStockBL. isAttented | 语法 | | public attentionState isAttented(String id) |
| 前置条件 | | 无 |
| 后置条件 | | 返回id是否被关注 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| ManageStockData. addStock(String id) | | 在本地关注列表中加上id | |
| manageStockData. deleteStock (String id); | | 在本地关注列表中删除id | |
| ManageStockData. getCodeOfStock() | | 得到所有关注的股票列表 | |

#### 4.2.5策略模拟模块

##### 4.2.5.1 StrategyHandle模块

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| StrategyHandle. handle | 语法 | | Public ArrayList<profitPO> handle(ArrayList<StrategyPO> arrayList1, ArrayList<StrategyPO> arrayList2) |
| 前置条件 | | 无 |
| 后置条件 | | 返回该策略的收益数据 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| StockDataService.getStockRecord | | 提供相关策略中包含的历史交易记录 | |

#### 4.2.6大盘数据模块

##### 4.2.6.1BenchData

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| BenchData. getBench | 语法 | | public List getBench() throws Exception |
| 前置条件 | | 数据层能提供所需的大盘名称数据 |
| 后置条件 | | 返回所需数据 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| BenchDaoProxyService. getBenchDaoProxy | | 提供大盘数据 | |

##### 4.2.6.2BenchRecord模块

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| BenchRecord. getBenchData | 语法 | | public List getBenchData(String benchId, String starttime, String endtime) throws Exception |
| 前置条件 | | 数据层能提供所需的大盘数据 |
| 后置条件 | | 返回所需数据 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| BenchDaoProxyService. getBenchDaoProxy | | 提供大盘历史交易数据 | |

## 数据层的分解.

* 1. **数据层模块接口规范**

股票查询数据层的接口规范

股票数据StockData接口

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| StockDataService.getCodeNames | 语法 | public codeNamePO getCodeName(int year,String exchange) |
| 前置条件 | API中有返回所有股票代号的接口 |
| 后置条件 | 返回一个codeNamePO |
| StockDataService.getStatisitcOfStock | 语法 | public ArrayList<stockStatisticPO> getStatisitcOfStock(String codeName,String start,String end) |
| 前置条件 | API中有查询所有股票数据的接口 |
| 后置条件 | 返回一个相应的ArrayList<stockStatisticPO> |

大盘数据BenchData接口

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| BenchDataService.getBenchmark | 语法 | public benchmarkPO getBenchmark() |
| 前置条件 | API中有返回所有股票大盘指数的接口 |
| 后置条件 | 返回一个相应的benchmarkPO |
| BenchDataService.getStatisticOfBenchmark | 语法 | public ArrayList<benchmarkStatisticPO> getStatisticOfBenchmark (String benchCode , String start,String end) |
| 前置条件 | API中有返回指定股票大盘数据的接口 |
| 后置条件 | 返回一个相应的ArrayList<benchmarkStatisticPO> |
| BenchDataService. getStatisticOfField | 语法 | public fieldStatisticPO getStatisticOfField () |
| 前置条件 | API中有返回可用的交易数据字段的接口 |
| 后置条件 | 返回相应fieldStatisticPO |

交易用语fieldData接口

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| fieldDataService. getStatisticOfField | 语法 | public fieldStatisticPO getStatisticOfField () |
| 前置条件 | API中有返回所有股票交易用语的接口 |
| 后置条件 | 返回一个包含这些用语数据的fieldStatisticPO |

处理观察数据manageStockService接口

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| manageStockService. getCodeOfStock | 语法 | public ArrayList<String> getCodeOfStock () |
| 前置条件 | API中有返回所有股票交易用语的接口 |
| 后置条件 | 返回一个包含这些用语数据的fieldStatisticPO |
| manageStockService. addStock | 语法 | Public ManageState addStock () |
| 前置条件 | API中有返回所有股票交易用语的接口 |
| 后置条件 | 返回一个包含这些用语数据的fieldStatisticPO |
| manageStockService. deleteStock | 语法 | Public ManageState deleteStock () |
| 前置条件 | API中有返回所有股票交易用语的接口 |
| 后置条件 | 返回一个包含这些用语数据的fieldStatisticPO |

监听网络状况ConnectionChecker接口

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| ConnectionChecker. checkconnection | 语法 | public boolean checkconnection() |
| 前置条件 | 有网络交互 |
| 后置条件 | 根据网络状况返回状态 |