Quantour

项目设计文档

V1.2

组名：Innovator Team

成员：谢振宇、王凡、伍俊、杨凯

2017/4/23

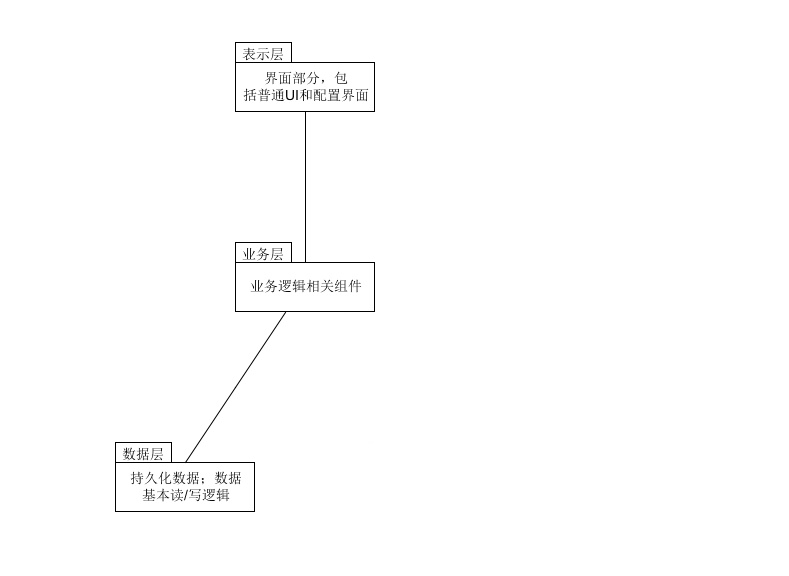
# 文档更新记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 杨凯 | 基本架构 | 2017/3/4 |
| V1.1 | 杨凯 | 逻辑层分解和分层细化 | 2017/3/5 |
| V1.2 | 王凡 | 数据层完善 | 2017/3/6 |
| V1.3 | 杨凯、王凡 | 新增迭代二部分 | 2017/4/23 |

# 引言：

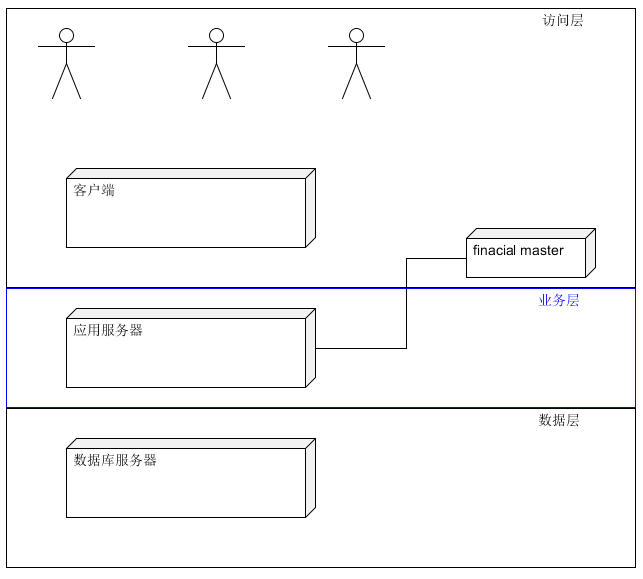
|  |  |
| --- | --- |
| 内容说明 |  |
| 1 编写目的  （说明整个文档所有达到的目标） | 本文档提供师生交流系统的软件架构概览，采用若干架构试图描述系统的不同方面，以便表示构造系统所需要的重要架构决策 |
| 2 对象与范围  （说明整个文档的内容范围和针对的读者对象） | 本文档的读者是Innovator Team团队内部的开发和管理人员，参考了RUP的《软件架构文档模板》，用于指导下一循环的代码开法和测试工作 |
| 3 参考文献  （说明文档中主要的所需所用） | 《软件需求规格说明书》，Innovator Team  《软件架构文档模板》，Rational 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） |  |

# 系统的分层架构：



系统划分为以下3个逻辑层次。

1. 展示层：用于前台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。



# 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

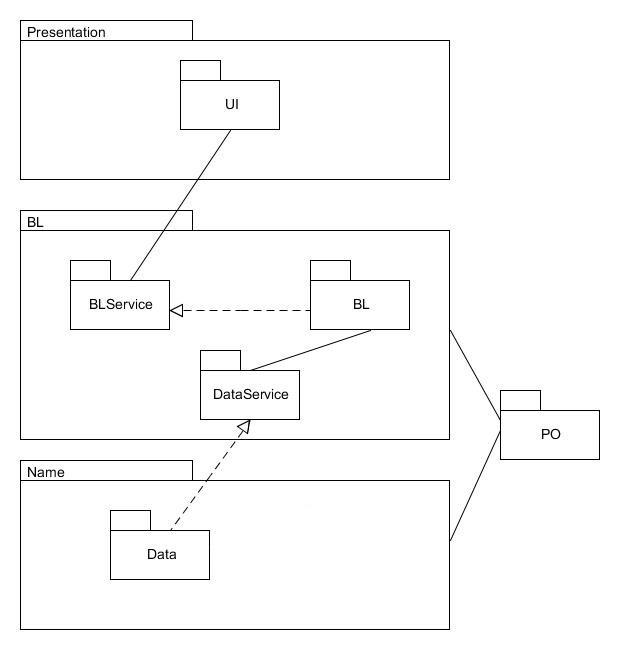
## 系统的架构设计如下。

系统架构中的对象分为7类

1. UI对象，负责处理系统数据的展示和用户的交互
2. BLService对象，负责向界面层提供抽象接口
3. BL对象，负责对BLService抽象接口的实现
4. DataService对象，负责向逻辑层提供抽象接口
5. Data对象，负责对DataService抽象接口的实现

6）PO对象，用于将数据对象打包成PO进行数据传递

## 系统中的组件和组件接口：



## 4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** |
| StockInfoBL | 负责股票信息展示和大盘信息展示 |
| BlockServiceImpl | 负责得到板块信息 |
| ColumnDiagramServiceImpl | 负责策略和基准比较的直方图 |
| StrategyEESBL | 负责超额收益率和策略胜率的计算 |
| StrategyVSBaseBL | 负责策略和基准比较的曲线图 |

##### 4.2业务逻辑层模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockInfoBL. getStockForKLine | 语法 | public image getStockForKLine(Date startDate,Date endDate,String info) |
| 前置条件 | 启动查询 |
| 后置条件 | 返回相应的k线图 |
| StockInfoBL.compareTwoStocks | 语法 | public Map<Integer,StockVO>compareTwoStocks (Date startDate,Date endDate,String info1,String info2 ) |
| 前置条件 | 查看两只股票的状况 |
| 后置条件 | 返回两只股票的集合，为k线图做准备 |
| StockInfoBL.getMarketerInfo | 语法 | Public MarketerInfoVO getMarketerInfo (Date date) |
| 前置条件 | 用户查看市场信息 |
| 后置条件 | 返回市场的VO |
| StockInfoBL.getLogRateVariance | 语法 | public double getLogRateVariance (Date startDate,Date endDate,String info) |
| 前置条件 | 查看对数收益率方差 |
| 后置条件 | 返回对数收益率方差的对数乘以100 |
| StockInfoBL.getHistoryCloseValue | 语法 | public Map<Date,StockCloseValueVO>getHistoryCloseValue (String nameOrCode) |
| 前置条件 | 用户需要查看历史价格 |
| 后置条件 | 返回历史最低价的vo集合 |
| StockInfoBL.searchStock | 语法 | public Map<String,Integer> searchStock (String nameOrCode) |
|  | 前置条件 | 股票的残缺信息查询 |
|  | 后置条件 | 返回股票名称和code的集合 |
| StockInfoBL.getHotStock | 语法 | public List<HotStockVO> getHotStock (Date date) |
|  | 前置条件 | 用户查看热门股票 |
|  | 后置条件 | 返回十个股票 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StockDataService. getSpecificDateStock() | 通过时间段和id查询股票 | |
| StockDataService. getSpecificDateStock() | 通过时间段和那么查询股票 | |
| StockDataService.getStockByDate() | 查询某一天的股票 | |
| StockDataService.searchStock() | 模糊搜索股票 | |

###### 4.2.2 BlockServiceImpl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| BlockServiceImpl.getStocksInBlock | 语法 | Public Map<Integer, StockVO> getStocksInBlock(Date date,StockPlate name) |
| 前置条件 | 启动股票板块查询 |
| 后置条件 | 返回板块内股票的集合 |
| **需要的服务（需接口）** | | |
| 服务名 | | 服务 |
| BlockDataServiceImpl.getStocksByDateAndPlate(date, name); | | 得到当日板块股票的集合 |

###### 4.2.3 ColumnDiagramServiceImpl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ColumnDiagramServiceImpl. getDataWithinAllStocks | 语法 | public ColumnDiagramVO getDataWithinAllStocks(Date startDate, Date endDate, int becomingDays, int possessingDays,StrategyType type) |
| 前置条件 | 启动所有股票回测 |
| 后置条件 | 得到股票直方图的VO |
| ColumnDiagramServiceImpl. getDataWithinPlates | 语法 | public ColumnDiagramVO getDataWithinPlates(Date startDate, Date endDate, int becomingDays, int possessingDays,StrategyType type,PlateType,plate) |
| 前置条件 | 启动股票池回测 |
| 后置条件 | 返回股票直方图的VO |
| ColumnDiagramServiceImpl. getDataWithinSomeStocks | 语法 | public ColumnDiagramVO getDataWithinSomeStocks(Date startDate, Date endDate, int becomingDays, int possessingDays,StrategyType type, ArrayList<TrolleyVO> trolleyVOS) |
| 前置条件 | 启动自选股票回测查询 |
| 后置条件 | 返回自选股票回测的结果 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StrategyHelper. getGreastSttocksInAll(Date lastDate,int growDays,StrategyType type) | 股票池为所有股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StrategyHelper. getGreastSttocksInPlate(Date lastDate,int growDays,StrategyType type,StockPlate plate) | 股票池为板块股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StrategyHelper. getGreastSttocksInSelect(Date lastDate,int growDays,StrategyType type,ArrayList<String > selectLists) | 股票池为自选股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StockStrategyService.getYieldOfSelectsStocks(ArrayList<String> selectedLists , int Growdays,Date startDate) | 得到对应日期前一段时间的股票的收益率 | |

###### 4.2.4 StrategyEESBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StrategyEESBL. getAllStocksEESVO | 语法 | public StrategyExcessEarningVO getAllStocksEESVO(Date startDate,Date endDate,StrategyType strategyType,TimeType timeType ,int nums,int firstDays,int iterval,int coloms) |
| 前置条件 | 启动所有股票回测查询 |
| 后置条件 | 返回所有股票和基准的超额收益率和胜率 |
| StrategyEESBL. getStockPoolEESVO | 语法 | public StrategyExcessEarningVO getStockPoolEESVO(StockPlate plate,Date startDate,Date endDate,StrategyType strategyType,TimeType timeType ,int nums,int firstDays,int iterval,int coloms) |
| 前置条件 | 启动股票板块回测查询 |
| 后置条件 | 返回板块内股票和基准的超额收益和胜率 |
| StrategyEESBL. getUserStocksEESVO | 语法 | public StrategyExcessEarningVO getUserStocksEESVO(ArrayList<String>selectedStocks,Date startDate,Date endDate,StrategyType strategyType,TimeType timeType ,int nums,int firstDays,int iterval,int coloms) |
| 前置条件 | 启动自选股票回测查询 |
| 后置条件 | 返回自选股票和基准的超额收益率 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StrategyHelper. getGreastSttocksInAll(Date lastDate,int growDays,StrategyType type) | 股票池为所有股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StrategyHelper. getGreastSttocksInPlate(Date lastDate,int growDays,StrategyType type,StockPlate plate) | 股票池为板块股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StrategyHelper. getGreastSttocksInSelect(Date lastDate,int growDays,StrategyType type,ArrayList<String > selectLists) | 股票池为自选股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StockStrategyService.getYieldOfSelectsStocks(ArrayList<String> selectedLists , int Growdays,Date startDate) | 得到对应日期前一段时间的股票的收益率 | |

###### 4.2.5 StrategyVSBaseBL接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StrategyVSBaseBL. useStrategyWithinAllStocks | 语法 | public StrategyResultVO useStrategyWithinAllStocks(Date startDate, Date endDate, int becomingDays, int possessingDays, StrategyType type); |
| 前置条件 | 启动所有股票的回测 |
| 后置条件 | 返回所有股票和基准的比较结果 |
| StrategyVSBaseBL. useStrategyGenerally | 语法 | public StrategyResultVO useStrategyGenerally(Date startDate, Date endDate, StockPlate plate, int becomingDays, int possessingDays ,StrategyType type) |
| 前置条件 | 启动股票板块的回测 |
| 后置条件 | 返回板块内股票和基准的回测结果 |
| StrategyVSBaseBL. useStrategyWithinSomeStocks | 语法 | public StrategyResultVO useStrategyWithinSomeStocks(Date startDate, Date endDate, ArrayList<String> selectedStocks, int becomingDays, int possessingDays, StrategyType type) |
| 前置条件 | 启动自选股票回测查询 |
| 后置条件 | 返回自选股票和基准的比较结果 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StrategyHelper. getGreastSttocksInAll(Date lastDate,int growDays,StrategyType type) | 股票池为所有股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StrategyHelper. getGreastSttocksInPlate(Date lastDate,int growDays,StrategyType type,StockPlate plate) | 股票池为板块股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StrategyHelper. getGreastSttocksInSelect(Date lastDate,int growDays,StrategyType type,ArrayList<String > selectLists) | 股票池为自选股票，对应策略在形成期内选择得到的需要购买的股票列表 | |
| StockStrategyService.getYieldOfSelectsStocks(ArrayList<String> selectedLists , int Growdays,Date startDate) | 得到对应日期前一段时间的股票的收益率 | |

## 5．数据层的分解

##### 5.1数据层的职责

**数据层的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| StockDataService | 负责股票信息的管理 |
| BlockDataSService | 负责股票板块信息的管理 |
| StockStrategyService | 负责股票策略信息的管理 |

##### 5.2数据层接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| StockDataService. getSpecificDateStock() | 语法 | public Map<Date,StockPO> getSpecificDateStock (Date startDate Date endDate,int code) |
| 前置条件 | 需要查询一段时间的股票 |
| 后置条件 | 返回一段时间内的stockPo |
| StockDataService. getSpecificDateStock() | 语法 | public Map<Date,StockPO> getSpecificDateStock(Date startDate Date endDate,String name) |
| 前置条件 | 需要查询一段时间的股票 |
| 后置条件 | 返回一段时间内的stockPo |
| StockDataService.getStockByDate() | 语法 | public Map<Integer,StockPO> getStockByDate (Date date) |
| 前置条件 | 需要所有的股票 |
| 后置条件 | 返回一个所有股票的集合 |
| StockDataService.searchStock() | 语法 | public Map<String,Integer> searchStock (String nameOrCode) |
| 前置条件 | 股票信息不全 |
| 后置条件 | 返回一个相应的名称和id的集合 |

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| BlockDataService. getStocksByDateAndPlate | 语法 | public Map<Integer, StockPO> getStocksByDateAndPlate(Date date, StockPlate plate) { |
| 前置条件 | 需要根据版块查询一段时间的股票 |
| 后置条件 | 返回一段时间内该版块内的stockPo |

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| StockStrategyService. getSomeStocks() | 语法 | public Map<String, StockPO> getSomeStocks(Date date,ArrayList<String> stockLists) |
| 前置条件 | 需要查询一些股票某天的数据 |
| 后置条件 | 返回这些股票某天内的stockPo |
| StockStrategyService. getSomeStocksOfSomeDays() | 语法 | public Map<String, Map<Date, StockPO>> getSomeStocksOfSomeDays(Date date,int days,ArrayList<String> stockLists) |
| 前置条件 | 需要查询一些股票一段时间的股票 |
| 后置条件 | 返回这些股票一段时间内的stockPo |
| StockStrategyService. getYieldOfSelectsStocks() | 语法 | public ArrayList<YieldPO> getYieldOfSelectsStocks(ArrayList<String> selectedLists , int Growdays,Date startDate) |
| 前置条件 | 需要一些股票一段时间的收益率 |
| 后置条件 | 返回这些股票该段时间的收益率集合 |