QuantourExploration

软件架构设计文档

V2.0 正式版

南京大学

Team No.666

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 崔伯暘 | 2017-3-16 | 最初草稿 | V1.0 草稿 |
| 宋少威 | 2017-3-17 | 评审后的正式版 | V2.0 正式版 |
|  |  |  |  |

# 团队名称： QuantourExploration

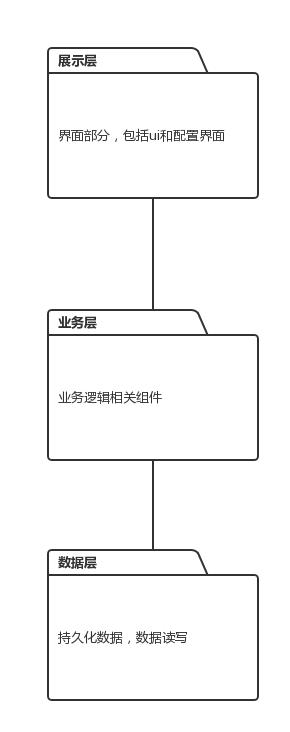
# 文档更新记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 宋少威 | 基本架构 | 2017/3/4 |
| V1.1 | 崔伯暘 | 逻辑层分解和分层细化 | 2017/3/8 |
| V1.2 | 王宇 | 数据层完善 | 2017/3/13 |

# 引言：

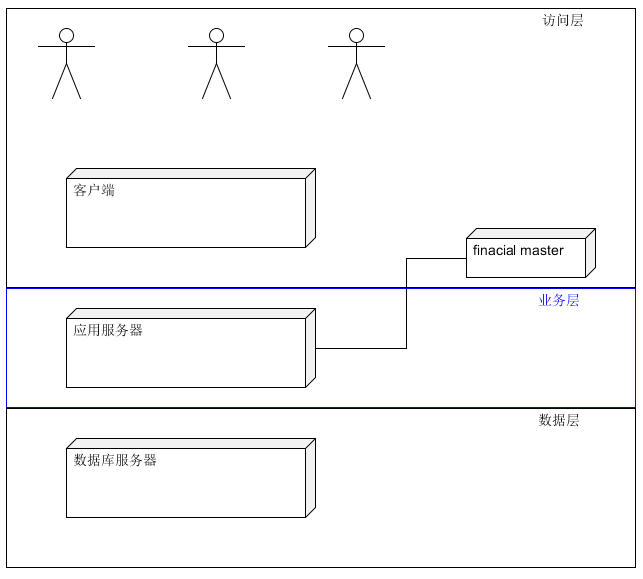
|  |  |
| --- | --- |
| 内容说明 |  |
| 1 编写目的  （说明整个文档所有达到的目标） | 本文档提供股票查询系统的软件架构概览，采用若干架构试图描述系统的不同方面，以便表示构造系统所需要的重要架构决策 |
| 2 对象与范围  （说明整个文档的内容范围和针对的读者对象） | 本文档的读者是QuantourExploration团队内部的开发和管理人员，参考了IHRS的《软件架构文档模板》，用于指导下一循环的代码开法和测试工作 |
| 3 参考文献  （说明文档中主要的所需所用） | 《软件需求规格说明书》，QuantourExploration |
| 4 名词与术语  （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4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
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

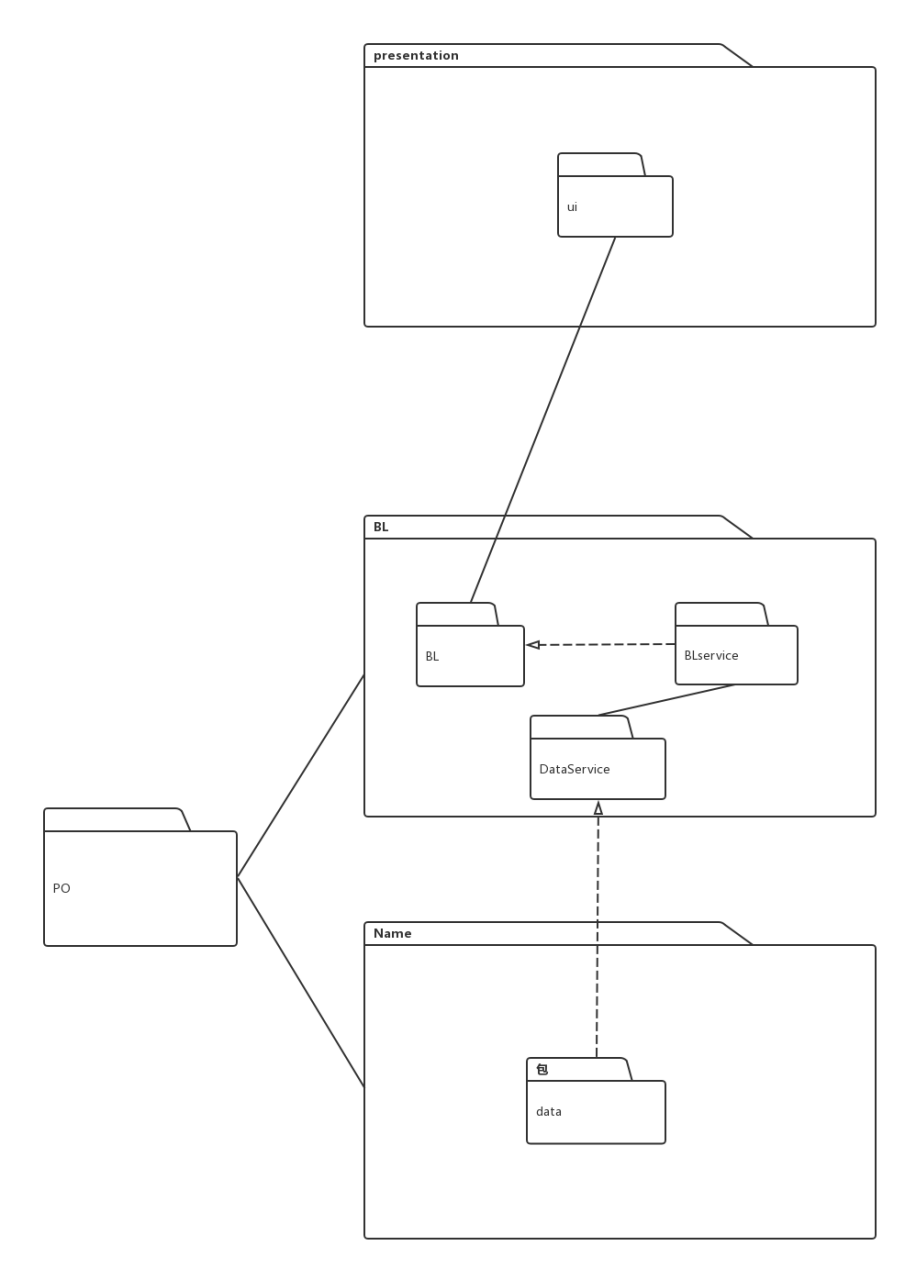
## 系统的架构设计如下。

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：



##### 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** |
| CompareBL | 比较两只股票的信息 |
| KMapBL | 画出k线图和均线图 |
| LineChartDemoBL | 折线图数据处理 |
| MarketBL | 市场温度计逻辑部分 |
| MarketThermometerBL | 市场温度计画图部分 |
| StockBL | 获取股票vo |
| TransferBL | 股票po转换为股票vo |

##### 4.2业务逻辑层模块的接口规范

##### 4.2.1个股日线 K 线+均线图展示模块

##### 4.2.1.1画k线图和均线图模块

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| KmapBL.kTest | 语法 | public JFreeChart kTest(ShareLineVO shareLineVO) | |
| 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k线图和均线图 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| StockBL.getShareLine(Date beginTime, Date endTime, String code) | | | 从数据层得到股票信息 |

##### 4.2.2股票比较模块

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| CompareBL.createChart | 语法 | | private JFreeChart createChart(CategoryDataset categorydataset, String title) |
| 前置条件 | | 启动两个股票比较回合 |
| 后置条件 | | 根据关键字和时间返回比较信息 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| StockBL.getShareLine(Date beginTime, Date endTime, String code) | | 得到指定证券指定时间的大盘信息 | |

##### 4.2.3市场情况温度计模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 提供的服务（供接口） | | | | |
| MarketBL.calculateThermo | 语法 | | public ThermometerVO calculateThermo(Date date, ArrayList<DailyStockVO> today) |
| 前置条件 | | 启动一个市场查询回合 |
| 后置条件 | | 根据关键字和时间返回当前最新的大盘信息 |
| MarketThermometerBL.creatChart | 语法 | | public JFreeChart creatChart(CategoryDataset dataset) |
| 前置条件 | | 启动一个市场查询回合 |
| 后置条件 | | 根据关键字和时间返回当前最新的大盘信息图像 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| StockBL.getShareLine(Date beginTime, Date endTime, String code) | | 得到指定证券指数指定时间的大盘信息 | |

**4.2.4数据的读取和转换**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| StockBL.getPossibleCode | 语法 | public ArrayList<String> getPossibleCodes(String head) |
| 前置条件 | 启动一个股票查询回合 |
| 后置条件 | 根据表头返回股票的代码 |
| Stock.getShareLine | 语法 | Public ShareLineVO getShareLine(Date beginTime, Date endTime, String code) |
| 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股票信息 |
| Stock.getShareLineByName | 语法 | public ShareLineVO getShareLineByName(Date beginTime, Date endTime, String name) |
| 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股票信息 |
| TransferBL.toDailyStockVO | 语法 | public DailyStockVO toDailyStockVO(StockUpDownPO po)) |
| 前置条件 | 启动一个股票查询回合 |
| 后置条件 | 根据PO返回股票的VO |
| TransferBL.toStockShareVO | 语法 | public StockShareVO toStockShareVO(StockUpDownPO po) |
| 前置条件 | 启动一个股票查询回合 |
| 后置条件 | 根据PO返回股票的VO |
| TransferBL.toShareLineVO | 语法 | public ShareLineVO toShareLineVO(ArrayList<StockShareVO> shareVOs, Date beginDate, Date endDate, String code, String name, StockShareVO yesterdayShare) |
| 前置条件 | 启动一个股票查询回合 |
| 后置条件 | 根据PO返回股票的VO |

##### 数据层的分解

##### 股票查询数据层的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockDao.getSearchStocks | 语法 | public ArrayList<StockUpDownPO> getSearchStocks(Date start,Date end,String code) |
| 前置条件 | API中有返回查询股票的接口 |
| 后置条件 | 返回一个ArrayList<StockUpDownPO> |
| StockDao.getDailyStock | 语法 | public ArrayList<StockUpDownPO> getDailyStock(Date date) |
| 前置条件 | API中有返回股票数据的接口 |
| 后置条件 | 返回一个ArrayList<stockStatisticPO> |
| StockNameDao.getStockName | 语法 | public HashMap<String, String> getStockName() |
| 前置条件 | API中有返回所有股票名称的接口 |
| 后置条件 | 返回一个HashMap<String, String> |
| StockNameDao.getStockCode | 语法 | public ArrayList<String> getStockCode() |
| 前置条件 | API中有返回股票代码的接口 |
| 后置条件 | ArrayList<String> |