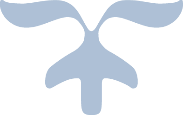


AnyQuant股票分析平台软件设计文档

V1.0



2016-3-4

南京大学软件学院

JLBuffett小组

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 罗金宏 | 2016-03-04 | 建立文档草稿 | V1.0草稿版 |
| 梁家诚 | 2016-03-05 | 添加market和detail模块 | V1.0正式版 |
| 焦润 | 2016-03-19 | 添加迭代二内容 | V2.0试行版 |

目录

[1. 引言 2](#_Toc444972992)

[2. 系统的体系架构 2](#_Toc444972993)

[2.1 逻辑视角 2](#_Toc444972994)

[2.2 组合视角 3](#_Toc444972995)

[2.2.1 开发包图 3](#_Toc444972996)

[2.2.2 运行进程 4](#_Toc444972997)

[2.2.3 物理部署 5](#_Toc444972998)

[2.3 接口视角 5](#_Toc444972999)

[2.3.1 模块的职责 5](#_Toc444973000)

[2.3.2 用户界面层模块的接口规范 6](#_Toc444973001)

[2.3.3 业务逻辑层模块的接口规范 7](#_Toc444973002)

[2.3.4 数据层接口规范 9](#_Toc444973003)

1. 引言

|  |  |
| --- | --- |
| **内容和说明** |  |
| 1. 编写目的 | 本文档提供AnyQuant股票分析系统平台的软件架构概览，采用若干架构视图描述股票分析系统的各个方面，以便构造系统所需的重要架构决策。 |
| 1. 对象与范围 | 本文档的读者是JLBuffett工作组内部的全体人员，用于指导该阶段及下一循环的代码开发和测试工作。 |
| 1. 参考文献 | 《软件工程与计算 卷三》骆斌，刘嘉，张瑾玉，黄蕾  《AnyQuant股票分析系统需求规格说明文档》，JLBuffett小组  The Object Mannagemen Group(OMG),The Unified Modeling Language Specification V1.4,2003 |
| 1. 名词与术语 | 参见《AnyQuant股票分析系统需求规格说明文档》，JLBuffett小组 |

1. 系统的体系架构
   1. 逻辑视角

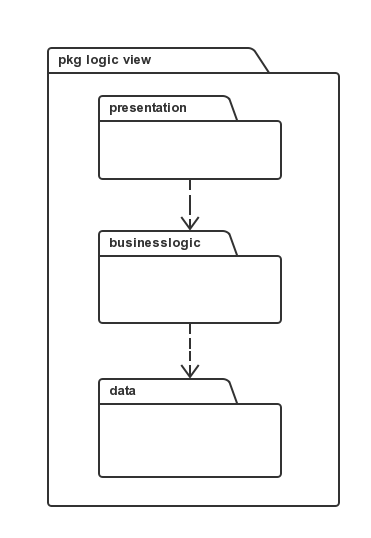
本系统选择了分层体系结构风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好地示意整个高层抽象。展示层包含GUI界面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角见图2.1-1。

图2.1-1参照体系结构风格的包图表达逻辑视角

* 1. 组合视角

开发包图

本系统的开发包图如图2.2.1-1所示。

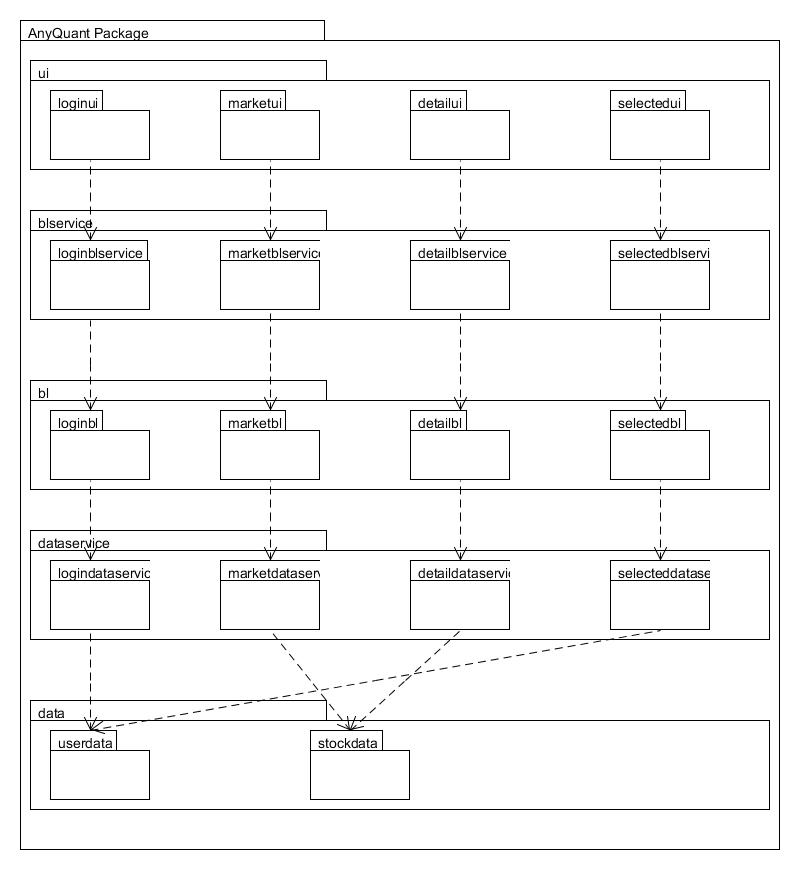


图2.2.1-1 软件系统开发包图

运行进程

在AnyQuant股票分析系统中,会有多个软件系统用户进程和一个网络数据系统,其进程图如图2.2.2所示。结合部署图,软件系统进程是在本地的机器上运行,本地软件系统通过网络连接从网络数据系统获得数据。

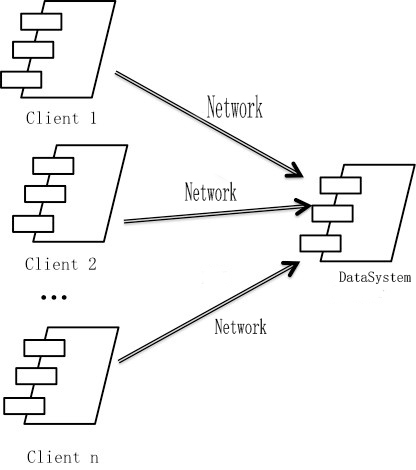
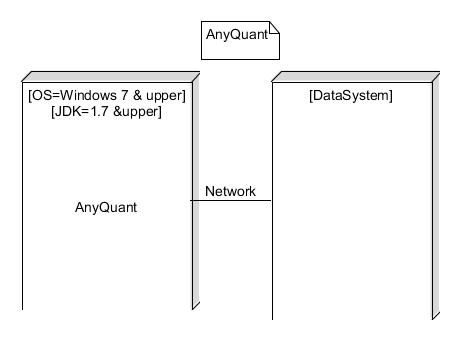


图2.2.2 进程图

物理部署



部署图如图2.2.3所示。

图2.2.3 部署图

* 1. 接口视角

模块的职责

客户端模块和服务器端模块试图分别如图2.3.1-1和图2.3.1-2所示。客户端各层和服务器端各层的职责分别如表2.3.1-1和表2.3.1-2所示。

图2.3.1-1 客户端模块视图

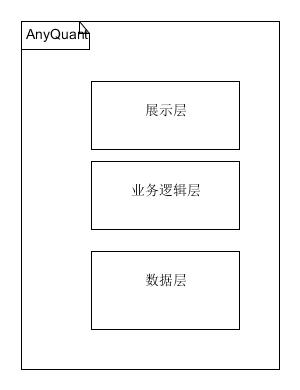


表2.3.1-1 客户端模块职责

|  |  |
| --- | --- |
| **层** | **职责** |
| **展示层** | 负责基于窗口的股票分析系统平台 |
| **业务逻辑层** | 响应界面的操作和处理业务逻辑 |
| **服务器端网络模块** | 负责数据的获取及持久化 |

用户界面层模块的接口规范

**表2.3.2-1loginui模块的接口规范**

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| **服务名** | 服务 |
| login | 判断是否登录信息是否输入正确 |
| register | 处理匿名用户注册操作 |

**表2.3.2-1marketui模块的接口规范**

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| **服务名** | 服务 |
| getMarketData | 获得某一天的某个交易所的大盘数据 |
| getSortedMarketData | 获得按某种属性排好序的大盘数据 |

**表2.3.2-3detailtui模块的接口规范**

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| **服务名** | 服务 |
| **getDetailData** | 获得某一只股票的一段时间的数据 |
| **getSortedDetailData** | 获得某一只股票按某一属性排好序的信息 |
|  |  |

**表2.3.2-4selectedui模块的接口规范**

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| **服务名** | 服务 |
| **getSelectedStock** | 获得自选股的信息 |
| **GetSortedSelectedStock** | 获得按某种属性排好序的单只股票信息 |
|  |  |

业务逻辑层模块的接口规范

**表2.3.3-1selectedbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **SelectedStock.remove** | 语法 | Public bool remove(String id) |
| 前置条件 | 股票的id符合要求 |
| 后置条件 | 返回是否移除成功的信息 |
| **SelectedStock.selectSort** | 语法 | Public bool selectedSort（List<StockVO> stockVOs,SortHelper sortHelper,Sorting sorting） |
| 前置条件 | SortHelper,Sorting(升序，降序)已被完整定义 |
| 后置条件 | 返回排序后的数据 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **logindataservice.remove** | 从账户中删除某只自选股 | |
| Stockdataservice.getStockData | 提供需要的股票数据 | |

**表2.3.3-2loginbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Login.login** | 语法 | Public bool login(String name,String password) |
| 前置条件 | 空 |
| 后置条件 | 返回是否登录成功的信息 |
| **Register.regist** | 语法 | Public bool regist（Stirng name,Sting pwd,Stirng confrimpwd） |
| 前置条件 | 无 |
| 后置条件 | 返回是否注册成功的信息 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **logindataservice.remove** | 从账户中删除某只自选股 | |
| stockdataservice.getStockData | 提供需要的股票数据 | |

**表2.3.3-3marketbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| Show.getData | 语法 | list<stockPO> getData(Date) |
| 前置条件 |  |
| 后置条件 | 大盘显示所有股票 |
| Sort.sort | 语法 | list<stockPO> sort(list<stockPO>, Order, SortHelper) |
| 前置条件 |  |
| 后置条件 | 大盘股票排序 |
|  |  |  |
| Sift.sift | 语法 | list<stockPO> sift(list<stockPO>, Range, SiftHelper) |
| 前置条件 |  |
| 后置条件 | 大盘根据属性筛选 |
| Search.searchByName | 语法 | list<stockPO> searchByName(string name) |
| 前置条件 |  |
| 后置条件 | 显示符合名字的股票 |
| Search.searchByID | 语法 | list<stockPO> searchByID(string ID) |
| 前置条件 |  |
| 后置条件 | 显示符合编号的股票 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| stockdataservice.getDataByDate | 显示最近一天的大盘数据 | |

**表2.3.3-3marketblservice模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| getData | 语法 | list<stockPO> getData(Date) |
| 前置条件 |  |
| 后置条件 | 大盘显示所有股票 |
| sort | 语法 | list<stockPO> sort(list<stockPO>, Order, SortHelper) |
| 前置条件 |  |
| 后置条件 | 大盘股票排序 |
| sift | 语法 | list<stockPO> sift(list<stockPO>, Range, SiftHelper) |
| 前置条件 |  |
| 后置条件 | 大盘根据属性筛选 |
| sortHelper | 语法 | list<stockPO> sort(list<stockPO>, Order) |
| 前置条件 |  |
| 后置条件 |  |
| siftHelper | 语法 | list<stockPO> sift(list<stockPO>, Range) |
| 前置条件 |  |
| 后置条件 |  |

**表2.3.3-4detailblservice模块接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| Show.getData | 语法 | list<stockPO> getData(string ID) |
| 前置条件 |  |
| 后置条件 | 显示该股票的详细信息 |
| Sort.sort | 语法 | list<stockPO> sort(list<stockPO>, Order, SortHelper) |
| 前置条件 |  |
| 后置条件 | 大盘股票排序 |
| Select.select | 语法 | boolean select(string ID) |
| 前置条件 | 用户已经登录 |
| 后置条件 | 将该股票选为自选股 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| stockdataservice.getDataByID | 显示某只股票的所有数据 | |

**detailblservice模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| getData | 语法 | list<stockPO> getData(string) |
| 前置条件 |  |
| 后置条件 | 显示该股票的详细信息 |
| sort | 语法 | list<stockPO> sort(list<stockPO>, Order, SortHelper) |
| 前置条件 |  |
| 后置条件 | 股票排序 |
| sortHelper | 语法 | list<stockPO> sort(list<stockPO>, Order) |
| 前置条件 |  |
| 后置条件 |  |
| ATRhelper.getATR | 语法 | ArrayList<Double> getATR(String id,RangeDate range) |
|  | 前置条件 | 无 |
|  | 后置条件 | 无 |
| Mahelper.getMA | 语法 | ArrayList<Double> getMA(String id,RangeDate range,enum days) |
|  | 前置条件 | 无 |
|  | 后置条件 | 无 |

**大盘响应顺序**

|  |  |
| --- | --- |
| 打开软件 | marketbl.Show.getData |
| 点击某属性将大盘排序 | marketbl.Sort.sort |
| 再次点击该属性将大盘逆序 | marketbl.Sort.sort |
| 筛选 | marketbl.Sift.sift |
| 根据股票编号或名字搜索 | marketbl.Search.searchByID/searchByName |
| 点击某只股票名字查看具体信息 | detailbl.Show.getDat |

**详细信息查询顺序**

|  |  |
| --- | --- |
| 点击某属性排序 | detailbl.Sort.sort |
| 再次点击该属性逆序 | detailbl.Sort.sort |
| 点击自选股选择按钮 | detailbl.Select.select |

数据层接口规范

**表2.3.4-1 userdata模块的接口规范**

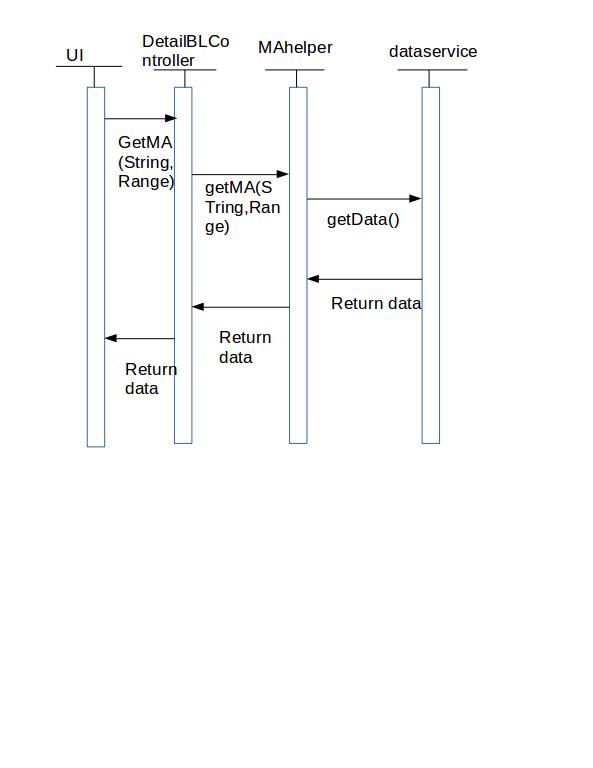
|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **LoginData.getUserByName** | 语法 | Public User getUserByName(String name) |
| 前置条件 | 无 |
| 后置条件 | 返回某个name的User |
| **RegistData.regist** | 语法 | Public bool regist（String name,String password） |
| 前置条件 | 无 |
| 后置条件 | 返回是否注册成功的信息 |
| **UserStock.add** | 语法 | Public bool add(String name,String id) |
| 前置条件 | name已存在，股票代号id符合规范 |
| 后置条件 | 返回是否add成功的信息 |
| **UserStock.remove** | 语法 | Public bool remove（String name，String id） |
| 前置条件 | name已存在，股票代号id符合规范 |
| 后置条件 | 返回是否remove成功的信息 |

**表2.3.4-2 detaildata模块的接口规范**

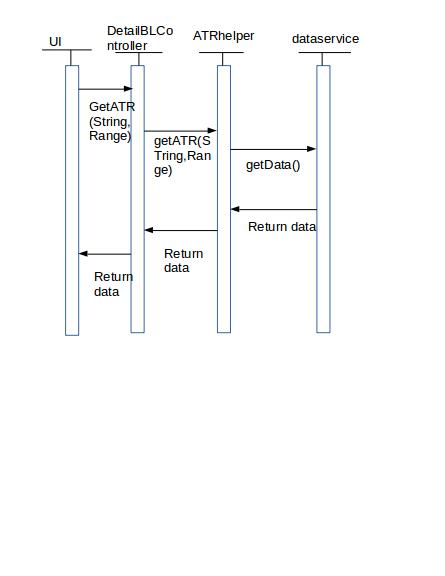
|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **StockData.getDataByID** | 语法 | Public List<StockPO> getDataByID(String id) |
| 前置条件 | Id符合规范 |
| 后置条件 | 返回某只股票的详细信息 |
| **StockData.getMarketData** | 语法 | Public List<StockPO> getMarketData(Date date,Range range,Position position) |
| 前置条件 | Date,Range,Position已定义 |
| 后置条件 | 返回某个交易所的某一天所有股票的信息 |

新增接口的顺序图：

1.getMA



2.getATR



（其他内容详见迭代一设计文档）