目录

[更新历史 2](#_Toc477098401)

[一、 引言 3](#_Toc477098402)

[(一) 目的 3](#_Toc477098403)

[(二) 参考文献 3](#_Toc477098404)

[二、 产品概述 3](#_Toc477098405)

[三、 逻辑视角 3](#_Toc477098406)

[四、 组合视角 4](#_Toc477098407)

[(一) 开发包图 4](#_Toc477098408)

[(二) 物理部署 5](#_Toc477098409)

[五、 接口视角 5](#_Toc477098410)

[(一) 模块的职责 5](#_Toc477098411)

[1. 用户界面层模块的职责 8](#_Toc477098412)

[2. 用户界面层模块设计原理 8](#_Toc477098413)

[(二) 业务逻辑层的分解 8](#_Toc477098414)

[1. 业务逻辑层模块职责 9](#_Toc477098415)

[2. 业务逻辑层模块的接口规范 9](#_Toc477098416)

[(三) 数据层分解 11](#_Toc477098417)

[1. 数据层模块职责 12](#_Toc477098418)

[2. 数据层模块的接口规范 12](#_Toc477098419)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 冯俊杰 | 2017/3/5 | 市场温度计需求规格添加 |  |
| 龚尘淼 | 2017/3/5 | 股票比较需求规格添加 |  |
| 董金玉 | 2017/3/5 | 股票数据管理和用户信息管理需求规格添加 |  |
| 董金玉 | 2017/3/10 | 整合需求规格说明文档，实现需求规格说明文档初版 | V1.0 |

# 引言

## 目的

本报告详细完成对电影评论和用户数据分析展示系统的概要设计，达到指导详细设计和开发目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

## 参考文献

1. IEEE标准
2. 《需求规格说明文档》

# 产品概述

参考《需求规格说明文档》中对产品的概括描述。

# 逻辑视角

电影评论和用户数据分析展示系统中，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。展示层包括UI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责从API上拉取数据。分层体系结构的逻辑视角和逻辑设计方案如图3-1和图3-2所示。



图3-1 参照体系结构风格的包图表达逻辑视角

# 组合视角

## 开发包图

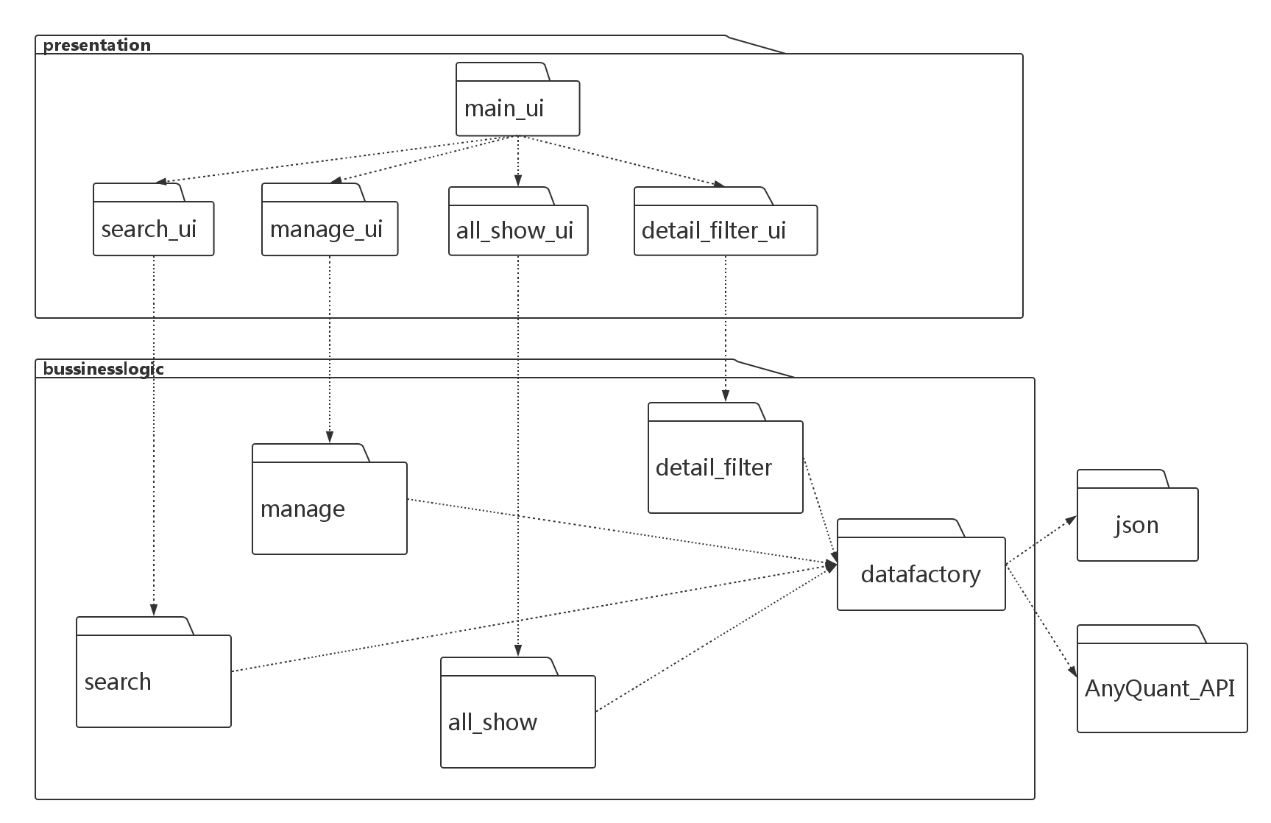


图4.1-1开发包图

## 物理部署

# 接口视角

## 模块的职责

|  |  |
| --- | --- |
| 层 | 职 责 |
| 用户界面层 | 展示股票信息（数据列表，走势图表） |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| 数据层 | 从API获取数据并保存，进行简单的数据处理 |

表5.1-1 各层的职责

层之间调用的接口如表5.1-3所示

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| public Stock getDetailStock(String code) throws IOException; | UI | BL |
| public List<KData> getKData(String code, UnitType unitType); | UI | BL |
| public List<KData> getKData(String code, UnitType unitType, Calendar start, Calendar end); | UI | BL |
| public List<KData> getAllKData(String code, UnitType unitType); | UI | BL |
| public List<MACDData> getMACDData(String code, UnitType type) throws IOException; | UI | BL |
| public List<MACDData> getMACDData(String code, UnitType type, Calendar start, Calendar end) throws IOException; | UI | BL |
| public List<MACDData> getAllMACDData(String code, UnitType type) throws IOException; | UI | BL |
| public List<RData> getRData(String code); | UI | BL |
| public List<RData> getRData(String code, Calendar start, Calendar end); | UI | BL |
| public List<RData> getAllRData(String code); | UI | BL |
| public List<VData> getVData(String code, UnitType unitType); | UI | BL |
| public List<VData> getVData(String code, UnitType unitType, Calendar start, Calendar end); | UI | BL |
| public List<VData> getAllVData(String code, UnitType unitType); | UI | BL |
| public List<String> getAllStocks() throws IOException | BL | DATA |
| public Stock getStockByCode(String code) throws IOException | BL | DATA |
| public Stock getStockByCode(String code, String start, String end) throws InvalidDate, IOException | BL | DATA |

**表5.1-3 层之间调用的接口**

### 用户界面层模块的职责

如表5.1.1-1所示为用户界面层模块的职责。

|  |  |
| --- | --- |
| 模块 | 职责 |
| sharexanalysepane | 负责单支股票的K线、成交量、MACD图表实现 |
| sharexindustrycompare | 负责行业板块换手率和涨幅对比图表实现 |
| sharexstockshow | 负责股票列表实现 |

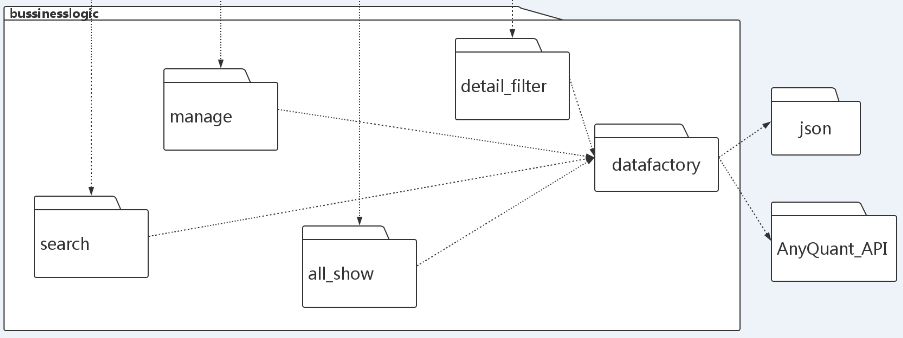
表5.1.1-1 用户界面层模块的职责

### 用户界面层模块设计原理

用户界面利用JavaSwing实现

## 业务逻辑层的分解

业务逻辑层包括多个针对界面的业务逻辑处理对象。业务逻辑层的设计如图：



### 业务逻辑层模块职责

业务逻辑层模块的职责如下表所示：

|  |  |
| --- | --- |
| 模块 | 职责 |
| KDataShowImpl | **实现K线图的数据拉取和处理的功能** |
| MACDDataShowImpl | **实现MACD相关数据处理的功能** |
| RDataShowImpl | **实现涨幅数据处理的功能** |
| VDataShowImpl | **实现成交量中换手率数据处理功能** |

### 业务逻辑层模块的接口规范

业务逻辑层模块的职责如下表所示：

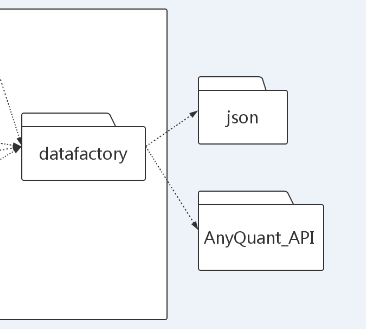
|  |  |
| --- | --- |
| 模块 | 职责 |
| KDataShowImpl | **实现K线图的数据拉取和处理的功能** |
| MACDDataShowImpl | **实现MACD相关数据处理的功能** |
| RDataShowImpl | **实现涨幅数据处理的功能** |
| VDataShowImpl | **实现成交量中换手率数据处理功能** |

1. KDataShowImpl模块
2. 表5.2.2-1 KDataShowImpl模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| List<KData> getKData | 语法 | public List<KData> getKData(String code, UnitType type) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统默认获得90个单位的K线图数据 |
| List<KData> getKData | 语法 | public List<KData> getKData(String code, UnitType unitType, Calendar start, Calendar end) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统根据指定时间获得K线图数据 |
| List<KData> getAllKData | 语法 | public List<KData> getAllKData(String code, UnitType unitType) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统获得全部K线图数据 |
| double getMAs | 语法 | private double getMAs(Stock newStock, String MAType, String calendar) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统根据股票代码、MA形式、日期计算均价 |
| 需要的服务（需接口） | | |
| public List<String> getAllStocks() throws IOException; | | 获得所有年份和交易所的所有股票代码 |
| public Stock getStockByCode(String code) throws IOException; | | 根据股票代码获得股票的信息，返回过去一个月的全部数据 |
| public Stock getStockByCode(String code, String start, String end) throws InvalidDate, IOException; | | 根据股票代码获得股票信息，返回的数据可以设置 |
| public Stock getStockByCode(String code, String start, String end) throws InvalidDate, IOException; | | 根据股票代码获得股票信息，返回的数据可以设置 |

## 数据层分解

数据层主要给业务逻辑层提供数据访问服务，包括对数据的增、删、改、查。数据层模块的描述具体如图5.3-1所示。



### 数据层模块职责

业务逻辑层模块的职责如下表所示：

|  |  |
| --- | --- |
| 模块 | 职责 |
| KDataShowImpl | **实现K线图的数据拉取和处理的功能** |
| MACDDataShowImpl | **实现MACD相关数据处理的功能** |
| RDataShowImpl | **实现涨幅数据处理的功能** |
| VDataShowImpl | **实现成交量中换手率数据处理功能** |

### 数据层模块的接口规范

业务逻辑层模块的职责如下表所示：

|  |  |
| --- | --- |
| 模块 | 职责 |
| KDataShowImpl | **实现K线图的数据拉取和处理的功能** |
| MACDDataShowImpl | **实现MACD相关数据处理的功能** |
| RDataShowImpl | **实现涨幅数据处理的功能** |
| VDataShowImpl | **实现成交量中换手率数据处理功能** |

1. KDataShowImpl模块
2. 表5.2.2-1 KDataShowImpl模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| List<KData> getKData | 语法 | public List<KData> getKData(String code, UnitType type) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统默认获得90个单位的K线图数据 |
| List<KData> getKData | 语法 | public List<KData> getKData(String code, UnitType unitType, Calendar start, Calendar end) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统根据指定时间获得K线图数据 |
| List<KData> getAllKData | 语法 | public List<KData> getAllKData(String code, UnitType unitType) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统获得全部K线图数据 |
| double getMAs | 语法 | private double getMAs(Stock newStock, String MAType, String calendar) |
| 前置条件 | Data层成功拿到数据 |
| 后置条件 | 系统根据股票代码、MA形式、日期计算均价 |
| 需要的服务（需接口） | | |
| public List<String> getAllStocks() throws IOException; | | 获得所有年份和交易所的所有股票代码 |
| public Stock getStockByCode(String code) throws IOException; | | 根据股票代码获得股票的信息，返回过去一个月的全部数据 |
| public Stock getStockByCode(String code, String start, String end) throws InvalidDate, IOException; | | 根据股票代码获得股票信息，返回的数据可以设置 |
| public Stock getStockByCode(String code, String start, String end) throws InvalidDate, IOException; | | 根据股票代码获得股票信息，返回的数据可以设置 |