

南京大学软件学院

2016 春 软件工程 AnyQuant项目实践

**软件详细设计文档**

**Chicer**

**程翔 陈天歌**

**韩奇 韩梦虞**

**一.引言**

**1.编制目的**

本报告详细描述了对AnyQuant系统的详细设计，达到指导后续软件构造的目的，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。

**2.历史版本**

|  |  |  |  |
| --- | --- | --- | --- |
| 更改人员 | 日期 | 变更内容 | 版本 |
| 程翔 | 2016- |  |  |

**3.参考资料**

1）IEEE标准

2）《需求规格说明文档》

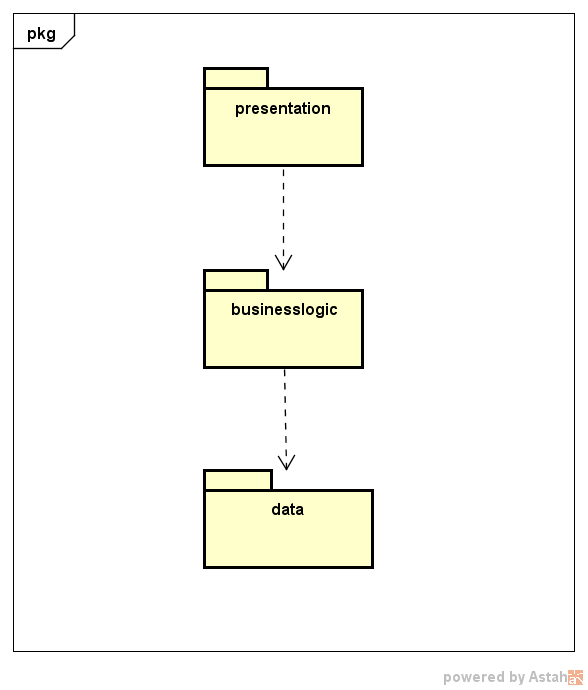
3）《软件工程与计算（卷三）》

**二.产品概述**

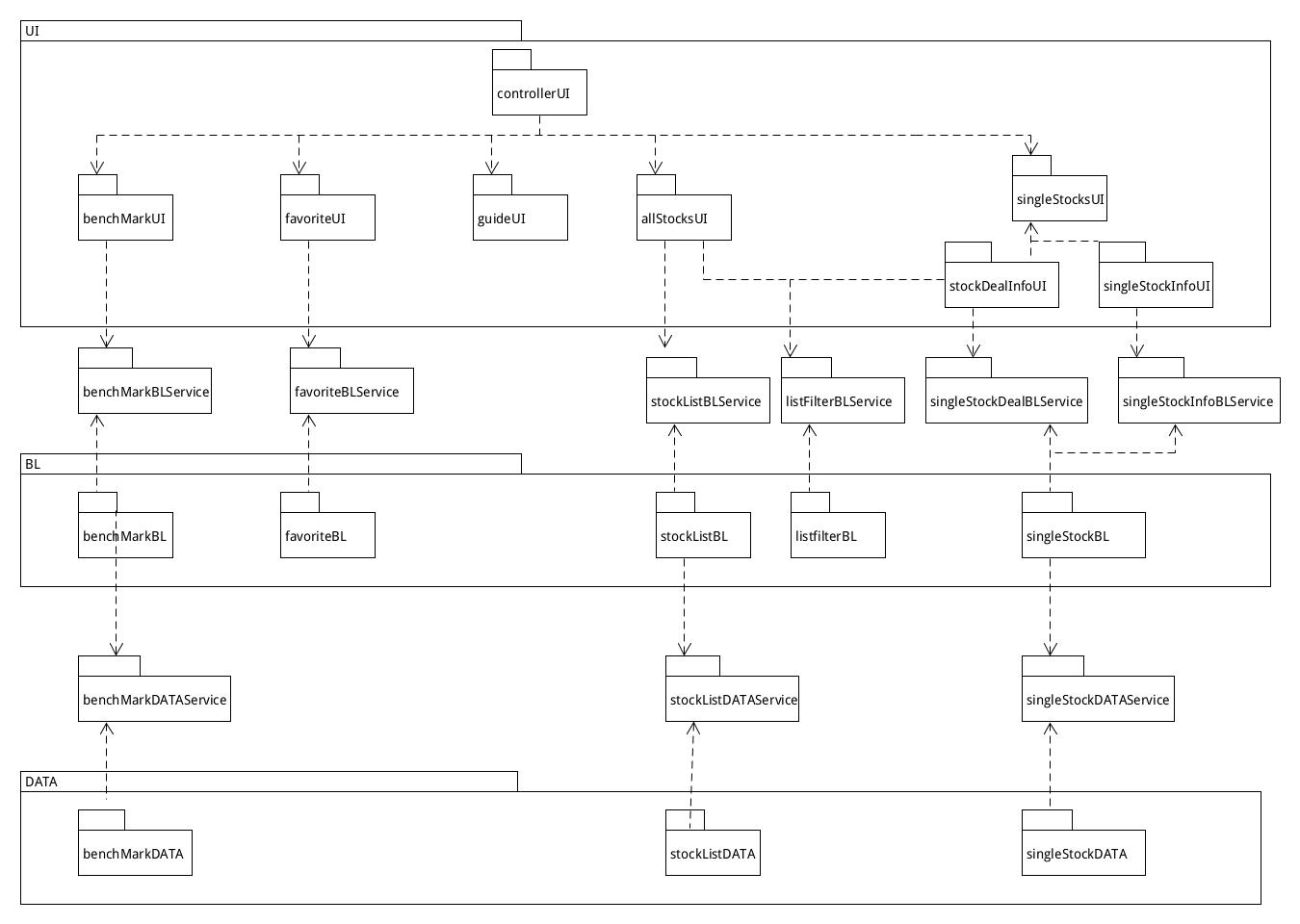
参考AnyQuant需求规格说明文档对产品的概括描述。

**三.架构风格**

AnyQuant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。展示层包含GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如下图所示（即其架构风格）。

****

具体的详细设计图如下：



**四．系统设计描述主体**

1、逻辑层部分：

1、benchMarkBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **getAllBanchMark** | 语法 | List<BenchMark> getAllBanchMark() |
| 前置条件 | 无 |
| 后置条件 | 返回大盘信息列表 |
| **getBenchMarkInfo** | 语法 | List<BenchMark> getBenchMarkInfo(String name, Calendar year) |
| 前置条件 | 传入指定的大盘名称、年份 |
| 后置条件 | 返回信息列表 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **benchMarkDATAService.getAllBenchMark** | 获得所有大盘名称 | |
| **benchMarkDATAService.getBenchMarkAtDate** | 获得指定日期、名称的大盘数据 | |
| **benchMarkDATAService.getBenchMarkAmongDate** | 获得日期区间内的大盘数据 | |

2、favoriteBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **getMyFavor()** | 语法 | List<Stock> getMyFavor() |
| 前置条件 | 无 |
| 后置条件 | 返回喜爱的股票列表 |
| **favorStock** | 语法 | OperationResult favorStock（String name） |
| 前置条件 | 传入关注的股票 |
| 后置条件 | 返回操作结果 |
| **unFavorStock** | 语法 | OperationResultunFavorStock（String name） |
| 前置条件 | 取消关注的股票名称 |
| 后置条件 | 返回操作结果 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **singleStockDATAService.getStockAtDate** | 获得指定日期的喜爱股票信息 | |

3、stockListBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **getAllStocks()** | 语法 | List<Stock> getAllStocks() |
| 前置条件 | 无 |
| 后置条件 | 返回所有股票的列表 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **stockListDATAService.getAllStocks** | 获得所有股票名称列表 | |
| **singleStockDATAService.getStockAtDate** | 获得指定日期的喜爱股票信息 | |

4、listFilterBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **OrderStocks** | 语法 | List<Stock> OrderStocks(List<Stock> srcStocks,String columnName, boolean orderType) |
| 前置条件 | 源数据、比较对象，排序类型 |
| 后置条件 | 返回排序后股票的列表 |
| **filterStrocksByFieldAmong** | 语法 | List<Stock> filterStrocksByFieldAmong(List<Stock> srcStocks ,String columnName, double min,double max) |
| 前置条件 | 源数据、比较对象，最大、最小 |
| 后置条件 | 过滤后的股票列表 |
| **filterStrocksByFieldGreater** | 语法 | List<Stock> filterStrocksByFieldGreater （List<Stock> srcStocks ,String columnName, double min) |
| 前置条件 | 源数据、比较对象、最小值 |
| 后置条件 | 过滤后的股票列表 |
| **filterStrocksByFieldLess** | 语法 | List<Stock> filterStrocksByFieldLess（List<Stock> srcStocks ,String columnName, double max) |
| 前置条件 | 源数据、比较对象，最大值 |
| 后置条件 | 过滤后的股票列表 |
| **filterStrocksByFieldEqual** | 语法 | List<Stock> filterStrocksByFieldEqual（List<Stock> srcStocks ,String columnName, double target) |
| 前置条件 | 源数据、比较对象，目标值 |
| 后置条件 | 过滤后的股票列表 |
| **filterAllStrocksByDateAmong** | 语法 | List<Stock> filterAllStrocksByDateAmong ( List<Stock> srcStocks ,String columnName, Calender min, Calender max) |
| 前置条件 | 源数据、日期区间 |
| 后置条件 | 过滤后的股票列表 |
| **filterAllStrocksByDateGreater** | 语法 | List<Stock> filterAllStrocksByDateGreater ( List<Stock> srcStocks ,String columnName, Calender min) |
| 前置条件 | 源数据、最早日期 |
| 后置条件 | 过滤后的股票列表 |
| **filterAllStrocksByDateLess** | 语法 | List<Stock> filterAllStrocksByDateLess ( List<Stock> srcStocks ,String columnName, Calender max) |
| 前置条件 | 源数据、最晚日期 |
| 后置条件 | 过滤后的股票列表 |
| **filterAllStrocksByDateEqual** | 语法 | List<Stock> filterAllStrocksByDateEqual ( List<Stock> srcStocks ,String columnName, Calender target) |
| 前置条件 | 源数据、目标日期 |
| 后置条件 | 过滤后的股票列表 |

5、singleStockInfoBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **getSingleStockInfo** | 语法 | Stock getSingleStockInfo（String name） |
| 前置条件 | 股票名称 |
| 后置条件 | 返回股票的信息 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **singleStockDATAService.getStockAmongDate** | 获得日期内的股票信息 | |

6、singleStockDealBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **getSingleStockDeal** | 语法 | List<Stock>getSingleStockDeal(Stringname，Calendar year) |
| 前置条件 | 传入该支股票的名称、年份 |
| 后置条件 | 返回单支股票的交易信息 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **singleStockDATAService.getStockAmongDate** | 获得日期内的股票信息 | |

2、数据层部分：

1、benchMarkDATAService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **benchMarkDATAService.getAllBenchMark** | 语法 | public list<BenchMark> getAllBenchMark() |
| 前置条件 | 无 |
| 后置条件 | 返回日期的大盘列表 |
| **benchMarkDATAService.getBenchMarkAtDate** | 语法 | PublicBenchMark getOperation(String name，Calendar date) |
| 前置条件 | 传入大盘名称、指定日期 |
| 后置条件 | 返回该大盘指定日期的数据 |
| **benchMarkDATAService.getBenchMarkAmongDate** | 语法 | public list<BenchMark> getBenchMarkAmongDate(String name，Calendar start，Calendar end) |
| 前置条件 | 传入大盘名称、日期区间 |
| 后置条件 | 返回该大盘指定日期内的信息 |

2、stockListDATAService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **stockListDATAService.getAllStocks** | 语法 | public list<String> getAllStocks(Calendar date， Exchange exchange) |
| 前置条件 | 传入指定年份、指定交易所的信息 |
| 后置条件 | 返回股票名称列表 |

3、singleStockDATAService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **singleStockDATAService.getStockAtDate** | 语法 | Public **Stock** getOperation(String name，Calendar date) |
| 前置条件 | 传入股票名称、指定日期 |
| 后置条件 | 返回该股票指定日期的数据 |
| **singleStockDATAService.getStockAmongDate** | 语法 | public list<Stock> getStockAmongDate(String name，Calendar start，Calendar end) |
| 前置条件 | 传入股票名称、日期区间 |
| 后置条件 | 返回该股票指定日期内的信息 |

3、自定义数据结构：

AbstractStock: String name

boolean isFavor 是否被关注

Calender date 数据日期

double open 开盘价

double high 最高价

double low 最低价

double close 收盘价

int volume 成交量

double adj\_price 后复权价

long marketvalue 市值

long flow 流通

Stock(股票): double turnover 换手率

double pe 市盈率

double pb 市净率

Benchmark(大盘): long transaction 成交额

User(用户)： String username 用户名

String password 密码

List<String> preferredStocks 关注的股票

4、文件格式：

所有文件均以.chicer结尾