A股金融证券分析系统

Quantourist

体系规格文档

迭代二

南京大学ASI

ASI of Nanjing University

2017-04-19

目录

更新历史

修改人员	修改日期	修改原因	版本号
高源	2017/3/4	提交组合视角	
高源	2017/3/4	提交界面层接口规范	
冯俊杰	2017/3/4	提交数据层接口规范	
冯俊杰	2017/3/4	提交信息视角	
龚尘淼	2017/3/4	提交逻辑层接口规范	
冯俊杰	2017/3/5	提交逻辑视角	
冯俊杰	2017/3/16	整合、修改所有相关文档	
冯俊杰	2017/3/17	整合、排版	V1.0
董金玉	2017/4/19	修改逻辑层接口，推进版本	V2.0
冯俊杰	2017/4/19	整合、修改、排版	V2.1

一、 引言

（一）编制目的

本报告详细完成对互联网酒店预订系统的概要设计，达到指导详细设计和开发的目的，同时实现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编写，是了解系统的导航。

（二）词汇表

缩写或单词	解释
Quantourist	ASI 小组开发的 A 股金融证券分析系统
stock	股票
stockSituation	股票市场温度计
chart	图表
stockComparision	股票比较
increaseMargin	涨跌幅
logarithmicYield	对数收益率
logarithmicYieldVariance	对数收益率方差
adj close	复权后的收盘指数
candlestick	K 线
average	均线
traceBack	股票回测
stock pool	股票池

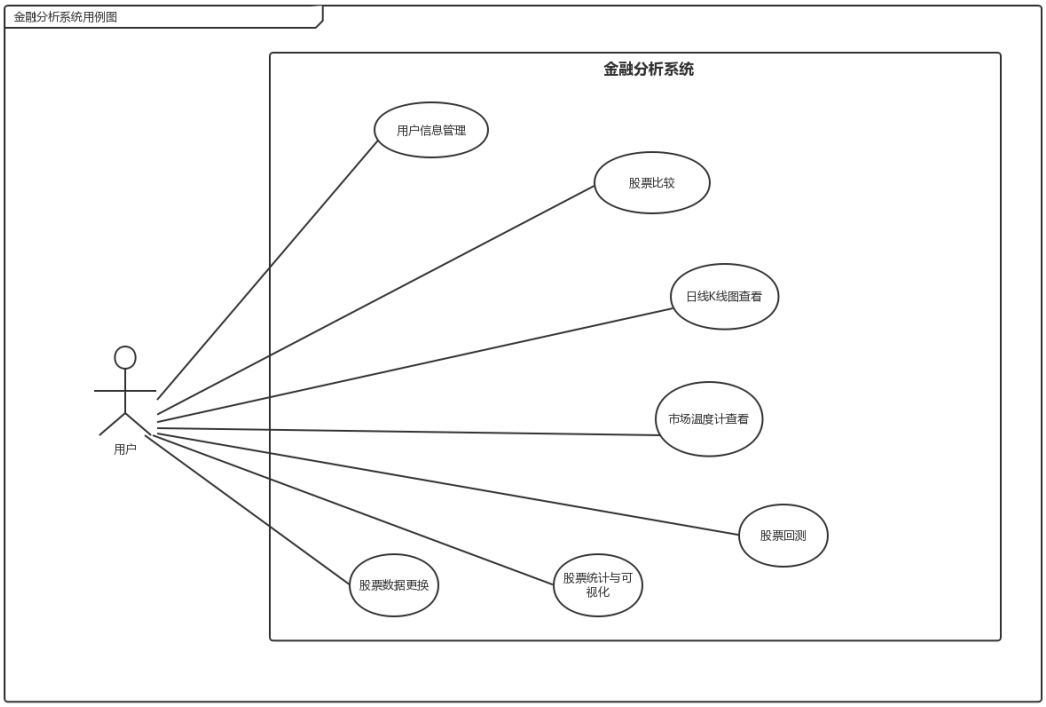
user	客户
logIn	登录
service	逻辑层
dao	数据层
dataHelper	数据助手

（三）参考资料

1. 丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版 2012：134—182
2. IEEE std 1471-2000
3. A 股金融证券分析系统需求规格说明文档 V2.1

二、 产品描述

参考《A 股金融证券分析系统需求规格说明文档》中对产品的概括描述。A 股金融证券分析系统主要功能见用例图如下。



三、 逻辑视角

A 股金融证券分析系统中，选择了分层体系结构的风格，将系统分为 3 层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括 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图 1 和图 2 所示。

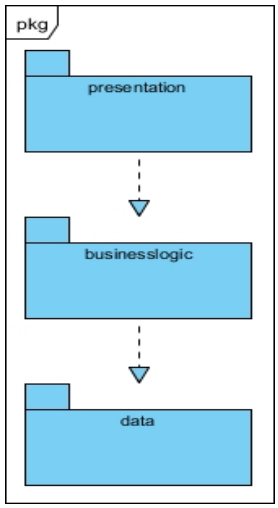


图 1：分层结构

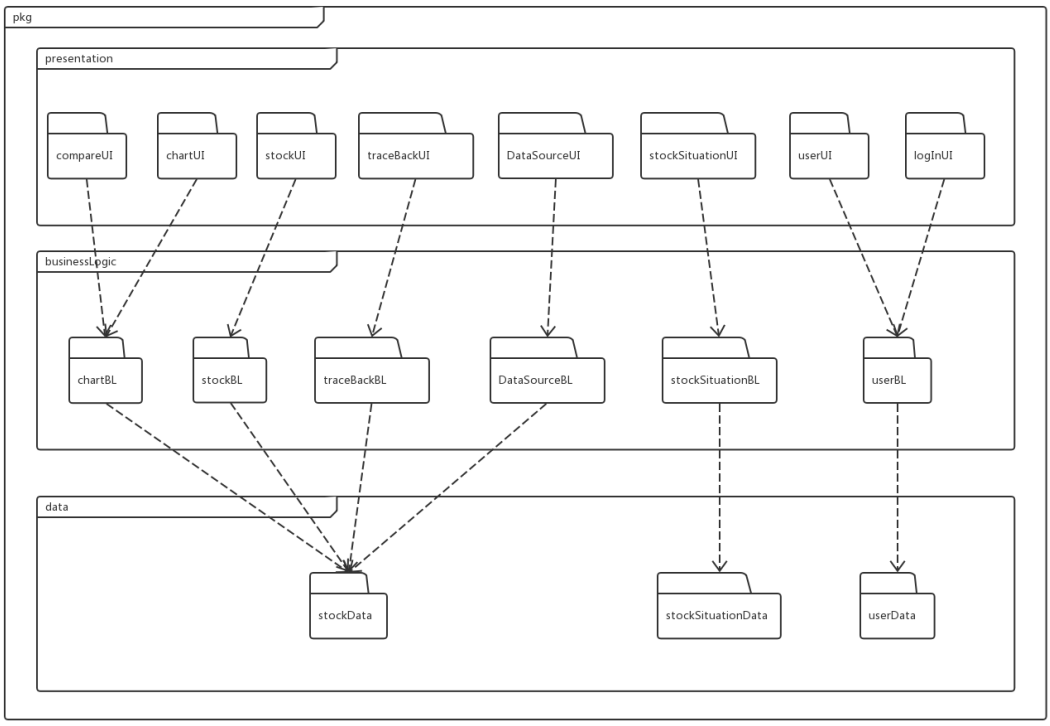


图 2：逻辑设计方案

四、 组合视角

（一）开发包图

与抽象的逻辑设计相比，实现物理设计要考虑更多的实现细节，这些细节有：

1）所有的 presentation 层开发包都需要使用图形类型建立界面，都要依赖于图形界面类库包。

2）在 presentation 层实现时，由 mainUI 包负责不同参与者在整个页面之间的跳转逻辑。其他各包负责各自页面自身的功能，并由各自的 controller 负责自身内部页面之间的跳转逻辑。

3）所有的 data 层开发包都需要进行数据持久化（例如读写数据库、读写文件等），所以它们会有一些重复代码，将重复代码独立为新的开发包 dataHelper。

4）在分层风格的典型设计中，不希望高层直接依赖于低层，而是为低层建立接口包，实现依赖倒置原则，所以应该调整为:各 presentation 层开发包(调用)依赖于逻辑层接口包 businessLogicService 包，businessLogic 层开发包也依赖于(实现了) businessLogic 层接口包 businessLogicService 包。

5）在分层风格的典型设计中，presentation 层与 logic 层之间、logic 层与 data 层之间可能会传递复杂数据对象，那么相邻两层都需要使用数据对象声明，所以需要将数据对象声明独立为开发包——vo 包和 po 包。其中 vo 包负责在 presentation 层与 logic 层之间传递，po 包负责在 logic 层与 data 层之间传递。

6）使用依赖倒置原则消除包的循环依赖现象，将循环依赖变为单向依赖。

7）在项目中，初始化和业务逻辑层上下文的工作被分配到 utilities 包中。

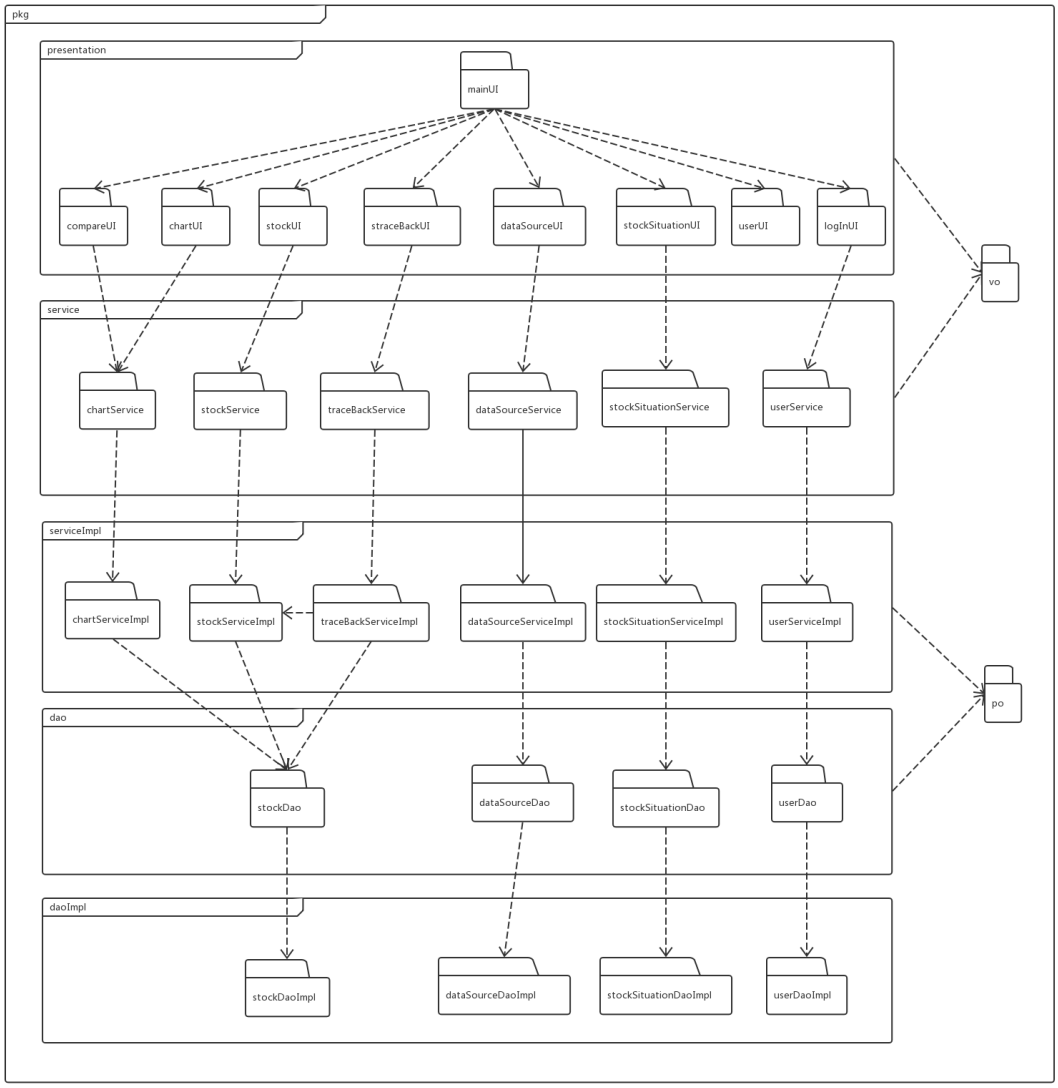
注意：在实际项目文件代码中，逻辑视角中逻辑分包的 businessLogic 实际为 service 和 serviceImpl；data 实际为 dao 和 daoImpl。

界面层氛围界面、控制器、监听、均线 / K 线等 4 个包；逻辑层主要依据逻辑分包；数据层进行整合合并，依照股票、市场温度计等实际情况进行分类。

开发（物理）包	依赖的其他包
presentation.view	service, controller, listener, line, jfreeChart, 界面类库包, utilities, vo
presentation.controller	service, view, jfreeChart, 界面类库包, utilities, vo
presentation.listener	controller, view, 界面类库包
presentation.line	service , jfreeChart, 界面类库包, utilities, vo

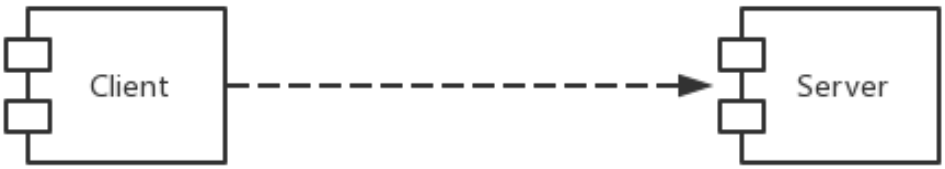
service	utilities, vo
service.serviceImpl	dao, service, po, vo, utilities
dao	utilities, po
dao.daoImpl	dataHelper, dao, po, utilities
dataHelper	po
dataHelper.dataHelperImpl	dataHelper, po, utilities
utilities	
vo	
po	
界面类库包	
java.javafx	
java.swing,java.awt,java.2D	

下图为 A 股金融证券分析系统开发包图:



（二）运行时进程

在 Quantourist 系统中，因为系统不联网，所有数据内置于系统中，服务器客户端为一体。故其进程图如下图所示。



（三）物理部署

Quantourist 系统中用户端、服务器端构一体为一个程序。在系统 JDK 环境已经设置好的情况下，不需要独立部署。

五、 接口视角

（一）模块的职责

1. 项目系统模块视图

2. 项目系统各层的职责

层	职责
用户界面层	基于窗口的互联网预订系统用户端用户界面
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	负责数据的持久化及数据访问接口

3. 各层联系

每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如下图所示。

接口	服务调用方	服务提供方
service.ChartService service.StockService service.StockSituationService service.UserService service.DataSourceService service.TraceBackService	用户端 展示层 （presentation）	服务器端 业务逻辑层 （service）
dao.StockDao dao.StockSituationDao dao.UserDao dao.DataSourceDao	服务器端 业务逻辑层 （service）	服务器端 数据层 （dao）

（二）用户界面层的分解

1. 用户界面层模块的职责

根据需求，系统存在 6 个用户界面（加粗为主界面）： **登录界面**、**注册界面**、日线 K 线均线显示界面，股票比较界面，股票市场温度计界面，股票总体界面。

用户界面层模块的职责：

模块	职责
Main	界面 frame，负责界面的显示和界面的跳转

2. 用户界面模块的接口规范

1） 用户界面层模块的接口规范

mainUI	语法	init(args:String[])
	前置条件	无
	后置条件	显示 Frame
logInUI.logIn	语法	boolean logIn(String userName,String password)
	前置条件	得到界面的用户名密码
	后置条件	成功登陆则记录该用户，否则提示原因
logInUI.register	语法	public boolean registerUser(UserVO userVO, String password2)
	前置条件	从界面得到用户信息
	后置条件	成功则注册成功，否则提示原因
kStringPanel.getSingleStockRecords	语法	List<StockVO> getSingleStockRecords (ChartShowCriteriaVO chartShowCriteriaVO)
	前置条件	有效的日期和股票号码
	后置条件	无
kStringPanel.getSingleStockRecords	语法	List <StockVO> getSingleStockRecords(String code)
	前置条件	有效的股票号码

	后置条件	无
kStringPanel.getAveData	语法	Map<Integer, List <MovingAverageVO>> getAveData(ChartShowCriteriaVO chartShowCriteriaVO, List<Integer> days)
	前置条件	有效的的日期股票号，查看的均线类型
	后置条件	无
ComparePanel.getComparision	语法	StockComparisionVO getComparision(StockComparsionCriteriaVO stockComparsionCriteriaVO)
	前置条件	有效的日期和股票编号
	后置条件	无
ThermometerPanel.getStockStituationData	语法	List <PriceRiseOrFallVO> getStockStituationData(LocalDate date);
	前置条件	有效的日期
	后置条件	无
TraceBackPanel.traceBack	语法	TraceBackVO traceBack(TraceBackCriteriaVO traceBackCriteriaVO, List<String> stockPool)
	前置条件	有效的回测条件
	后置条件	无

DataSourcePanel.upload	语法	boolean upload(String filePath)
	前置条件	有效的 csv 文件
	后置条件	无

2） 用户界面层模块需要的服务接口

服务名	服务
service.ChartService	负责处理与股票统计、比较相关的接口
service.StockService	负责处理与股票总体相关的接口
service.StockSituationService	负责处理与股票市场温度计相关的接口
service.UserService	负责处理与用户登陆注册相关的接口
service.DataSourceService	负责用户上传数据源的接口
service.TraceBackService	负责回测的接口

3. 用户界面模块设计原理

项目界面拟用 java 的 swing 库、awt 库、javafx 库来实现。

（三）业务逻辑层的分解

业务逻辑层包括多个针对界面及业务逻辑处理对象。例如对界面显示图表数据的获取和计算操作，对股票信息的获取操作，对市场温度计情况信息的获取操作和对用户信息的操作。

1. 业务逻辑层模块的职责

模块	职责
chartService	负责处理界面显示图表信息的获取和计算
stockService	负责获取股票的具体信息
stockSituationService	负责获取和计算市场温度计的情况

userService	负责提供所有用户的注册和登录的服务
DataSourceService	负责数据自动化拆分与更换处理的功能
TaceBackService	负责股票回测功能

2. 业务逻辑模块的接口模范

1） chart 模块的接口规范

提供的服务（供接口）		
chartService.getSingleStockRecords	语法	List<StockVO> getSingleStockRecords(String code)
	前置条件	无
	后置条件	系统返回指定股票的所有交易信息
chartService.getSingleStockRecords	语法	List <StockVO> getSingleStockRecords(ChartShowCriteriaVO chartShowCriteriaVO)
	前置条件	无
	后置条件	系统返回指定股票的所有交易信息
chartService.getAveData	语法	Map<MovingAverageType, List<MovingAverageVO>> getAveData(ChartShowCriteriaVO chartShowCriteriaVO, List<MovingAverageType> MATypes)
	前置条件	无

	后置条件	系统返回用户所选天数的均线图的移动平均值
chartService.getAveData	语法	Map<MovingAverageType, List<MovingAverageVO>> getAveData(String code, List<MovingAverageType> MATypes)
	前置条件	无
	后置条件	系统返回用户所选天数的均线图的移动平均值
chartService.getComparison	语法	List<StockComparisionVO> getComparision(StockComparsionCriteriaVO stockComparsionCriteriaVO)
	前置条件	无
	后置条件	系统返回界面所需的两只股票的比较信息
ChartService.getDateWithoutData	语法	List<LocalDate> getDateWithoutData(String stockCode)
	前置条件	无
	后置条件	系统返回界面所需的单只股票被剔除的日期
ChartService.getDateWithoutData	语法	List<LocalDate> getDateWithoutData(ChartShowCriteriaVO chartShowCriteriaVO)
	前置条件	无
	后置条件	系统返回界面所需的单只股票指定时间段被剔除的日期

ChartService.geFirstDayAndLastDay	语法	FirstLastDayVO getFirstAndLastDay(String stockCode)
	前置条件	无
	后置条件	系统返回界面所需的股票的起讫时间
需要的接口（需接口）		
服务名	**服务内容**	
StockDao.getAllStocksCode()	返回所有股票的股票代号	
stockDao.getStockData(String stockCode)	返回指定代码股票的所有数据	
stockDao.getStockData (String stockCode, LocalDate start, LocalDate end)	返回指定时间段内的指定股票所有数据	
stockDao.getFirstAndLastDay(String stockCode)	返回指定股票所有的起讫时间	
stockDao.getDateWithoutData(String code)	返回单只股票被剔除的日期	
StockDao.getDateWithoutData(String code,LocalDate start, LocalDate end)	返回单只股票指定时间段被剔除的日期	

2） stock 模块的接口规范

提供的服务（供接口）		
stockService.getAllStocks	语法	List<StockVO> getAllStocks(LocalDate date)
	前置条件	无
	后置条件	系统返回所有股票信息列表
stockService.getPrivateStocks	语法	List<StockVO> getPrivateStocks(String userName, LocalDate date)
	前置条件	用户已经登录
	后置条件	返回用户的自选股信息列表
stockService.getPrivateStockCodes	语法	List<String> getPrivateStockCodes(String userName)
	前置条件	无
	后置条件	返回符合查询条件的用户自选股代码列表
stockService.addPrivateStock	语法	boolean addPrivateStock(String userName, String stockCode)
	前置条件	用户已经登录
	后置条件	添加一条自选股
stockService.deletePrivateStock	语法	boolean deletePrivateStock(String userName, String stockCode)
	前置条件	用户已经登录
	后置条件	删除一条自选股

stockService.searchStock	语法	List<StockSearchVO> searchStock(String searchString)
	前置条件	无
	后置条件	返回符合查询条件的股票信息列表
stockService.getOneStockDateAndData	语法	Map<LocalDate, StockVO> getOneStockDateAndData(String stockCode, LocalDate start, LocalDate end)
	前置条件	无
	后置条件	返回指定股票信息和对应日期的键值对
stockService.getOneStockData	语法	List<StockVO> getOneStockData(String stockCode, LocalDate start, LocalDate end)
	前置条件	无
	后置条件	返回指定股票指定时间的所有股票信息
stockService.getBaseStockData	语法	List<StockVO> getBaseStockData(String stockName, LocalDate start, LocalDate end)
	前置条件	无
	后置条件	返回基准股票指定时间的所有股票信息
stockService.getStockPool	语法	List<String> getStockPool(StockPoolCriteriaVO stockPoolVO)
	前置条件	无

	后置条件	符合标准的股票池中所有股票的股票代码
需要的接口（需接口）		
服务名	**服务内容**	
stockDao.getStockData(LocalDate date)	返回指定日期保存的所有股票信息	
stockDao.getStockData(String stockCode, LocalDate start, LocalDate end)	返回指定日期保存的所有股票信息	
stockDao.getPrivateStockData(String username, LocalDate date)	根据用户指定的日期，返回用户自选股的数据	
stockDao.getPrivateStockCodes(String username)	根据用户指定的日期，返回用户自选股的数据	
stockDao.addPrivateStock(String username, String stockCode)	根据股票代码，添加一条自选股	
stockDao.deletePrivateStock(String username, String stockCode)	根据股票代码，删除一条自选股	
stockDao.getAllStocksC	返回所有股票的代码-名称键值对	

ode()	
stockDao.getAllStocksFirstLetters	返回所有股票的首字母-名称键值对
stockDao.getAllStockName	返回所有股票名称-代码键值对
stockDao.getAllStockPool	返回所有股票池
StockDao.getPrivateStockCodes(String stockName)	返回指定用户的所有股票代号

3） stockSituation 模块的接口规范

提供的服务（供接口）		
stockSituationService.getStockSituationData	语法	List<PriceRiseOrFallVO> getStockStituationData(LocalDate date)
	前置条件	无
	后置条件	返回指定日期股票涨跌情况的列表
需要的接口（需接口）		
服务名	**服务内容**	
stockSituationDao.getStockSituation(LocalDate date)	返回指定日期市场温度计数据	

4） user 模块的接口规范

提供的服务（供接口）		
userService.registerUser	语法	boolean registerUser(UserVO userVO, String password2)
	前置条件	无
	后置条件	返回用户是否注册成功
userSerivce.modifyUser	语法	boolean modifyUser(UserVO userVO)
	前置条件	用户已登录
	后置条件	系统更新用户信息
userService.logIn	语法	logIn(String userName,String password)
	前置条件	无
	后置条件	无
需要的接口（需接口）		
服务名	**服务内容**	
userDao.getAllUserNames()	获取已成功注册的所有用户的名称	
userDao.add(UserPO userPO)	添加一条用户的持久化数据	
userDao.modify(UserPO userPO)	修改一条用户的持久化数据	
userDao.get(String username)	获得一条用户的持久化数据	

5） dataSource 模块的接口规范

提供的服务（供接口）		
dataSourceService.upload	语法	boolean upload(String filePath)
	前置条件	用户已被验证登录
	后置条件	返回用户上传数据源的是否成功
dataSourceService.getMyDataSource	语法	DataSourceInfoVO getMyDataSource()
	前置条件	用户已被验证登录
	后置条件	返回用户自己上传的数据源格式信息
dataSourceService.setDataSourceState	语法	boolean setDataSourceState (DataSourceState newState)
	前置条件	用户已被验证登录
	后置条件	返回用户选择修改数据源来源是否成功
dataSourceService.getDataSourceState	语法	DataSourceState getDataSourceState()
	前置条件	用户已被验证登录
	后置条件	返回当前数据源类型
需要的接口（需接口）		
服务名	**服务内容**	
dataSourceDao.upload(String filePath)	返回用户上传数据源的是否成功	
dataSourceDao.getMyDataSource()	返回用户自己上传的数据源格式信息	

6） traceBack 模块的接口规范

提供的服务（供接口）		
traceBackService.traceBack	语法	TraceBackVO traceBack (TraceBackCriteriaVO traceBackCriteriaVO, List<String> stockPool)
	前置条件	用户已被验证登录
	后置条件	返回用户进行回测后的数据
traceBackService.getCustomizedCumulativeReturn	语法	List<CumulativeReturnVO> getCustomizedCumulativeReturn(LocalDate start, LocalDate end)
	前置条件	用户已被验证登录
	后置条件	返回基准累计收益率，自选股票池
需要的接口（需接口）		
服务名	**服务内容**	
stockService.getStockPool(StockPoolCriteriaVO stockPoolVO)	返回所有符合标准的股票池中所有股票的股票代码	
stockService.getBaseStockData(String stockName, LocalDate start, LocalDate end)	返回此基准股票在指定时间区间中的股票数据	
stockDao.getDateWithD	返回所有的交易日期	

ata()	
stockDao.getStockData(**String code)**	返回此股票的所有数据

（四）数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的查找。由于对持久化数据的保存底层可能存在多种形式：TXT 文件、序列化文件、数据库等，所以抽象了数据服务。（本系统底层使用 TXT 存取）

1. 数据层模块的职责

模块	职责
stockData	负责保存股票信息，进行查找的操作
stockSituationData	负责保存股票市场温度计信息，进行查找的操作
userData	负责保存用户信息，进行增、删、改、查的操作
dataSourceData	负责自动拆分处理及更换数据源

2. 数据层模块的接口规范

1） stockData 模块的接口规范

stockDao.getStockData	语法	StockPO getStockData(String stockCode, LocalDate date)
	前置条件	无
	后置条件	无
stockDao.getStockData	语法	List<StockPO> getStockData(String stockCode, LocalDate start, LocalDate

		end)
	前置条件	无
	后置条件	无
stockDao.getStockData	语法	List<StockPO> getStockData(String stockCode)
	前置条件	无
	后置条件	无
stockDao.getStockData	语法	List<StockPO> getStockData (LocalDate date)
	前置条件	无
	后置条件	无
stockDao.getDateWithoutData	语法	List<LocalDate> getDateWithoutData(String stockCode)
	前置条件	无
	后置条件	无
stockDao.getDateWithoutData	语法	List<LocalDate> getDateWithoutData(String stockCode, LocalDate start, LocalDate end)
	前置条件	无
	后置条件	无
stockDao.getDateWithData	语法	List<LocalDate> getDateWithData()
	前置条件	无

	后置条件	无
stockDao.getPrivateStockData	语法	List<StockPO> getPrivateStockData (String userName, LocalDate date)
	前置条件	无
	后置条件	无
stockDao.getPrivateStocks	语法	PrivateStockPO getPrivateStocks (String userName)
	前置条件	无
	后置条件	无
stockDao.getPrivateStockCodes	语法	List<String> getPrivateStockCodes (String userName)
	前置条件	无
	后置条件	无
stockDao.addPrivateStock	语法	boolean addPrivateStock(String userName, String stockCode)
	前置条件	无
	后置条件	无
stockDao.deletePrivateStock	语法	boolean deletePrivateStock(String userName, String stockCode)
	前置条件	无
	后置条件	无
stockDao.getFirstAndLast	语法	List<LocalDate> getFirstAndLastDay

Day		(String stockCode)
	前置条件	无
	后置条件	无
stockDao.getAllStocksCode	语法	Map<String, String> getAllStocksCode()
	前置条件	无
	后置条件	无
stockDao.getAllStocksFirstLetters	语法	Map<String, String> getAllStocksFirstLetters()
	前置条件	无
	后置条件	无
stockDao.getAllStocksName	语法	Map<String,String> getAllStocksName()
	前置条件	无
	后置条件	无
stockDao.getAllStockPool	语法	List<StockPoolVO> getAllStockPool()
	前置条件	无
	后置条件	无

2） stockSituationData 模块的接口规范

stockSituationDao.getStockSituation	语法	StockSituationPO getStockSituation (LocalDate date)
	前置条件	无

	后置条件	无

3） userData 模块的接口规范

userDao.add	语法	boolean add(UserPO userPO)
	前置条件	无
	后置条件	系统持久化增加该用户的数据
userDao.get	语法	UserPO get(String username)
	前置条件	无
	后置条件	无
userDao.modify	语法	boolean modify(UserPO userPO)
	前置条件	无
	后置条件	系统修改并持久化增加该用户的数据
userDao.getAllUserNames	语法	Set<Object> getAllUserNames
	前置条件	无
	后置条件	无

4） dataSource 模块的接口规范

DataSource.upload	语法	boolean upload(String filePath)
	前置条件	无
	后置条件	系统持久化保存新增股票数据
DataSource.getMyDataSource	语法	DataSourceInfoPO getMyDataSource()
	前置条件	无
	后置条件	无

六、 信息视角

（一）数据持久化对象

系统的 PO 类就是对应的相关的实体类，如下所示。

类名	包含的属性
StockPO	记录编号，日期(月/日/年)，开盘指数，最高指数，最低指数，收盘指数，成交量，复权后的收盘指数，股票代码，股票名称，市场名称，昨日收盘指数，昨日复权后的收盘指数
StockSituationPO	当日总交易量，涨停股票数，跌停股票数，涨幅超过 5%的股票数，跌幅超过 5%的股票数，开盘-收盘大于 5%*上一个交易日收盘价的股票个数、开盘-收盘小于-5%*上一个交易日收盘价的股票个数
PrivateStockPO	用户名，用户的自选股列表
AveragePO	收盘指数，股票代码，股票名称
StockSearchPO	股票代码，股票名称，汉字首字母名称
UserPO	用户姓名，用户密码
DataSourceInfoPO	上传文件大小，上传时间

（二）文件格式

本系统采用 txt 和资源文件保存文件。

所属包与表名	内容	字段设计
stocks/stock_recor	按股票代码	分别为记录编号，日期(月/日/年)，开盘指数，最高

ds_by_code/*	分类的非基准股票信息	指数，最低指数，收盘指数，成交量，复权后的收盘指数，股票代码，股票名称，市场名称，昨日收盘指数，昨日复权后的收盘指数。各变量之间用 tab 分隔。
stocks/stock_records_by_date/*	按股票日期分类的非基准股票信息	分别为记录编号，日期(月/日/年)，开盘指数，最高指数，最低指数，收盘指数，成交量，复权后的收盘指数，股票代码，股票名称，市场名称，昨日收盘指数，昨日复权后的收盘指数。各变量之间用 tab 分隔。
stocks/stock_situation/*	非基准股票的市场温度计	分别为当日总交易量，涨停股票数，跌停股票数，涨幅超过 5%的股票数，跌幅超过 5%的股票数，开盘-收盘大于 5%*上一个交易日收盘价的股票个数、开盘-收盘小于-5%*上一个交易日收盘价的股票个数。各变量之间用 tab 分隔。
stocks/stockName-code/stockName-code.properties	非基准股票的名字与代码对应	股票名称、股票代码的键值对
stocks/stockName-code/shortPinyin.txt	非基准股票的股票拼音与代码对应	股票拼音、代码的键值对
base_stocks/stock_records_by_code/*	按股票代码分类的基准	分别为记录编号，日期(月/日/年)，开盘指数，最高指数，最低指数，收盘指数，成交量，复权后的收盘

	股票信息	指数，股票代码，股票名称，市场名称，昨日收盘指数，昨日复权后的收盘指数。各变量之间用 tab 分隔。
base_stocks/stock_records_by_date/*	按股票日期分类的基准股票信息	分别为记录编号，日期(月/日/年)，开盘指数，最高指数，最低指数，收盘指数，成交量，复权后的收盘指数，股票代码，股票名称，市场名称，昨日收盘指数，昨日复权后的收盘指数。各变量之间用 tab 分隔。
base_stocks/stockName-code/stockName-code.properties	基准股票的名字与代码对应	股票名称、股票代码的键值对
base_stocks/stockName-code/shortPinyin.txt	基准股票的股票拼音与代码对应	股票拼音、代码的键值对
userinfo.properties	用户信息	用户名称、用户密码的键值对
[username]/[username].txt	用户自选股	用户自选股的集合