# 组合视角

## 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. presentation层被置于用户端，data层与businessLogic层被置于服务器端，那么presentation层的开发包不可能依赖于businessLogic层的开发包。
4. 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，可以将重复代码独立为新的开发包。
5. 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
6. 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。
7. 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

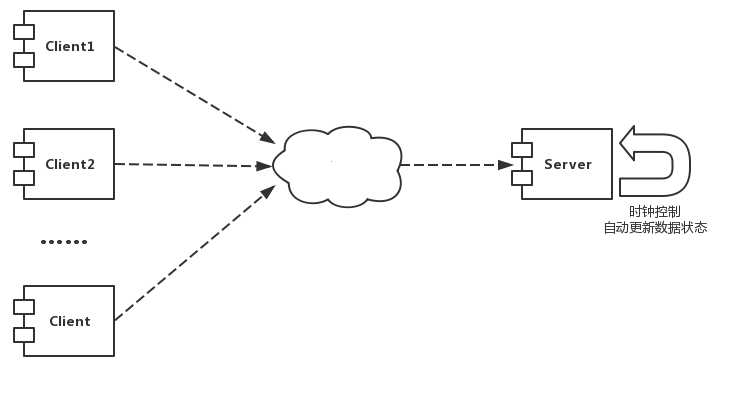
8）在logic层中，初始化和业务逻辑层上下文的工作被分配到utilities包中。

界面层依据功能需求分为登录注册界面和4个功能主界面；逻辑层主要依据逻辑分包；数据层进行整合合并，依照时间或者公司名称等实际情况进行分类。

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他包 |
| MainUI | 界面类库包 |
| LoginUI | loginPanelListener，界面类库包 |
| RegisterUI | registerPanelListener, 界面类库包 |
| KStringUI | StockService，jfreeChart, 界面类库包, VO |
| CompareUI | StockService, jfreeChart, 界面类库包,VO |
| ThermometerUI | StockSituationService, jfreeChart, 界面类库包,VO |
| FavoritesPanel | StockService, jfreeChart, 界面类库包,VO |
| ChartService | VO |
| StockService | VO |
| StockSituationService | VO |
| UserService | VO |
| ChartServiceImpl | StockDao, ChartService, PO，VO |
| StockServiceImpl | StockDao，StockService, PO,VO |
| StockSituationServiceImpl | StockSituationService，StockSituationDao, PO,VO |
| UserServiceImpl | StockDao, UserDao，UserService，PO,VO |
| StockDao | PO |
| StockSituationDao | PO |
| UserDao | PO |
| StockDaoImpl | StockDataHelper, PrivateStockDataHelper，StockDao, PO |
| StockSituationDaoImpl | StockSituationDataHelper, StockSituationDao, PO |
| UserDaoImpl | UserDao, UserDataHelper, LogDataHelper, PO |
| VO |  |
| PO |  |
| 界面类库包 |  |
| java.javafx |  |
| java.swing,java.awt,java.2D |  |

## 运行时进程

在Quantourist系统中，会有多个用户端进程和一个服务器端进程，其进程图如下图所示。结合部署图，用户端进程是在用户端机器上运行，服务器端进程是在服务器端机器上运行。数据库的自动更新会在服务器端自动进行，作为另一个进程。



## 物理部署

Quantourist系统中用户端构件是放在用户端机上，服务器端构件是放在服务器端机器上。在系统JDK环境已经设置好的情况下，不需要独立部署。部署图如下图所示。



# 接口视角

## 模块的职责

### 用户端模块视图



### 服务器端模块视图



### 用户端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化通信机制，启动用户界面 |
| 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 |

### 服务器端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化通信机制，启动用户界面 |
| 数据层 | 负责数据的持久化及数据访问接口 |

### 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** |
| ChartService  StockService  StockSituationService  UserService | 用户端  展示层  （presentation） | 服务器端  业务逻辑层  （businessLogic） |
| StockDao  StockSituationDao  UserDao | 服务器端  业务逻辑层  （businessLogic） | 服务器端  数据层  （data） |

## 用户界面层的分解

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，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 |
| MainPanel | 界面frame，负责界面的显示和界面的跳转 |

### 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 | 显示是否登陆成功 |
| 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 | 返回注册的结果 |
| kStringPanel. getSingleStockRecords | 语法 | Iterator<StockVO> getSingleStockRecords (ChartShowCriteriaVO chartShowCriteriaVO) |
| 前置条件 | 填写日期和股票号码 |
| 后置条件 | 获取单支股票的一段日期内的信息 |
| kStringPanel. getSingleStockRecords | 语法 | Iterator<StockVO> getSingleStockRecords(String code) |
| 前置条件 | 填写股票号码 |
| 后置条件 | 获取单支股票的所有数据 |
| kStringPanel.getAveData | 语法 | Map<Integer, Iterator<MovingAverageVO>> getAveData(ChartShowCriteriaVO chartShowCriteriaVO, List<Integer> days) |
| 前置条件 | 填写相应的日期股票号，查看的均线类型 |
| 后置条件 | 获取单支股票一段日期内，用户所选天数的均线图的平均值. |
| ComparePanel  .getComparision | 语法 | StockComparisionVO getComparision(StockComparsionCriteriaVO stockComparsionCriteriaVO) |
| 前置条件 | 用户输入日期和股票编号 |
| 后置条件 | 获取两只股票的比较信息 |
| ThermometerPanel. getStockStituationData | 语法 | Iterator<PriceRiseOrFallVO> getStockStituationData(LocalDate date); |
| 前置条件 | 用户查找到想要的酒店 |
| 后置条件 | 显示对应酒店的信息 |
| FavoritePanel. getAllStocks | 语法 | Iterator<StockVO> getAllStocks(LocalDate date) |
| 前置条件 | 用户填写了日期 |
| 后置条件 | 显示所有股票信息的列表 |
| FavoritePanel. getPrivateStocks | 语法 | Iterator<StockVO> getPrivateStocks(String userName, LocalDate date) |
| 前置条件 | 用户尚未注册会员 |
| 后置条件 | 显示用户的自选股信息列表 |
| FavoritePanel. deletePrivateStock | 语法 | public boolean deletePrivateStock(String userName, String stockCode); |
| 前置条件 | 用户登陆系统 |
| 后置条件 | 用户删除自选股 |

1. **用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| 服务名 | 服务 |
| ChartService | 有关统计图的业务逻辑接口 |
| StockService | 获取股票的相关的接口 |
| StockSituationService | 市场温度计的相关的接口 |
| UserService | 用户登陆注册相关的接口 |

### 用户界面模块设计原理

用户用户端界面拟用java的swing库来实现，服务器端用java的Swing、AWT实现。