### 1.2.1说明

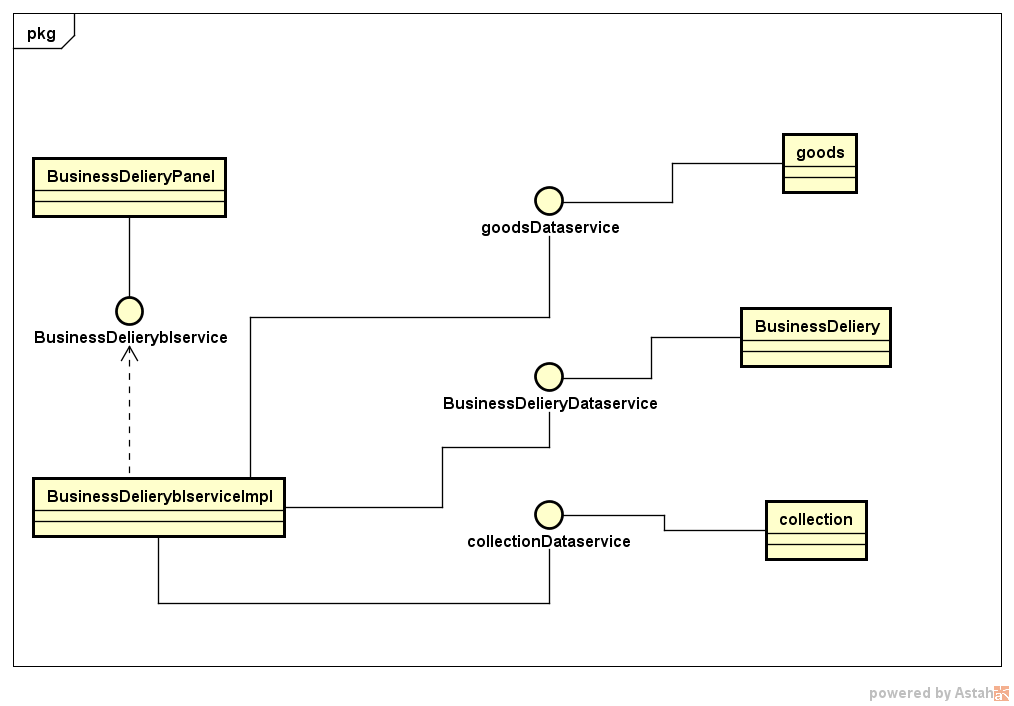
派件单界面BusinessDelieryPanel营业厅业务员界面BusinessPanel的一部分，营业厅业务员通过JtabblePanel分为多个界面（车辆管理界面(BusinessVehiclePanel)，司机管理界面(BusinessDriverPanel), 收款单界面(BusinessColletionPanel)，装车管理界面(BItransportPanel), 派件单界面(BusinessDelieryPanel)），下面详细说明派件单界面

收款单界面BusinessDelieryPanel需要在构造方法里传入收款单业务逻辑接口BusinessDelieryblservice，它定义了派件单的增删改查。其具体实现类BusinessDelieryblserviceimpl需要在构造方法中传入

以获得方法为例，从数据库获得一个BusinessDeliery对象，需要通过BusinessDelieryDataService获得BusinessDeliery对象（只包含其本身的属性，而不包含其他的对象），

再根据其外键的id获得BusinessDeliery里包含的其他对象

若其他对象还包含着其他对象，需要继续封装



### 1.2.2派件单详细设计描述

|  |  |
| --- | --- |
| 模块 | 职责 |
| BusinessDelieryPanel | BusinessDeliery的用户界面 |
| BusinessDelieryblserviceImpl | 负责有关BusinessDeliery的业务逻辑实现 |
| BusinessDelieryblservice | BusinessDeliery业务逻辑接口，定义需要的业务方法 |
| \*Dateservice | 各种实体类的数据接口，定义数据库操作方法 |

BusinessDelieryPanel**的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| BusinessDelieryPanel.view | 语法 | private void view() |
| 前置条件 | 用户点击表格的某一行 |
| 后置条件 | 根据选择的id调用查询方法，然后显示在下部的界面详细显示 |
| BusinessDelieryPanel.query | 语法 | private void query() |
| 前置条件 | 用户点击查询按钮 |
| 后置条件 | 根据查询文本框调用查询方法，然后将符合查询条件的BusinessDeliery显示在表格中 |
| BusinessDelieryPanel.setViewData | 语法 | public void setViewData() |
| 前置条件 | 界面初始化 |
| 后置条件 | 调用查询方法，然后将所有的BusinessDeliery显示在表格中 |
| **提供的服务（供接口）** | | |
| BusinessDelieryPanel.clear | 语法 | private void clear() |
| 前置条件 | 用户点击清空按钮，或者BusinessDeliery添加，修改，删除操作完成后 |
| 后置条件 | 清空下面的文本框和按钮，重新获得数据，重新获得提示信息 |
| BusinessDelieryPanel.save | 语法 | private void save() |
| 前置条件 | 用户点击保存按钮 |
| 后置条件 | 如果BusinessDelieryId的文本框(隐藏)的值为空, 则是新增,调用add方法，否则为修改，调用update方法 |
| BusinessDelieryPanel.add | 语法 | private void add() |
| 前置条件 | 用户点击保存按钮后是新增操作，检查文本框内容是否存在以及合法 |
| 后置条件 | 调用BusinessDelieryblservice的add方法，然后clear清空 |
| BusinessDelieryPanel.update | 语法 | private void update() |
| 前置条件 | 用户点击保存按钮后是更新操作，检查文本框内容是否存在以及合法 |
| 后置条件 | 封装其他实体类的主键值和BusinessDeliery的外键值，调用BusinessDelieryblservice的update方法，然后clear清空 |
| BusinessDelieryPanel.delete | 语法 | private void delete() |
| 前置条件 | 用户点击删除按钮 |
| 后置条件 | 获得选择删除的BusinessDeliery的id，调用BusinessDelieryblservice的delete方法，然后clear清空 |
| BusinessDelieryPanel.getBusinessDeliery | 语法 | private BusinessDeliery getBusinessDeliery() |
| 前置条件 | add或update方法调用时 |
| 后置条件 | 根据文本框以及其他按钮封装BusinessDeliery对象 |
| BusinessDelieryPanel.getGoods | 语法 | private Goods getGoods() |
| 前置条件 | add或update方法调用时 |
| 后置条件 | 根据文本框以及其他按钮封装goods对象 |
| BusinessDelieryPanel.getCollection | 语法 | private Collection getCollection() |
| 前置条件 | add或update方法调用时 |
| 后置条件 | 根据文本框以及其他按钮封装collection对象 |

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| 服务名 | 服务 |
| BusinessDelieryblservice | BusinessDeliery业务逻辑接口 |
|  |  |

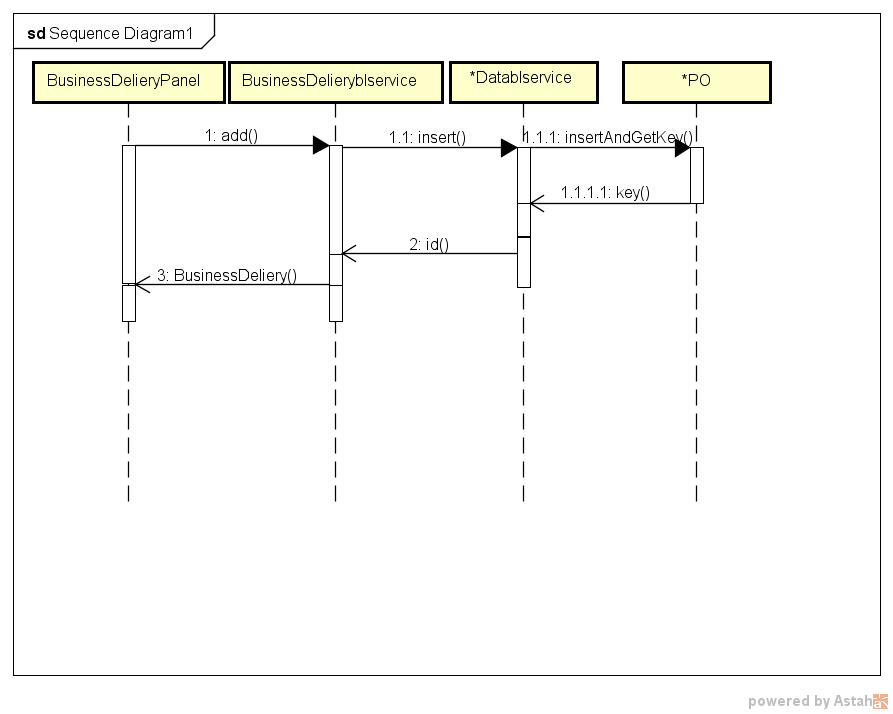
**BusinessDelieryblserviceImpl的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| BusinessDelieryblserviceImpl.getAll | 语法 | public Vector<BusinessDeliery> getAll() |
| 前置条件 | 需要查询所有的BusinessDeliery时 |
| 后置条件 | 调用BusinessDelieryDataService的getList方法， collectionDatas的getById方法，对获得的collection调用userDataservice的getById方法 |
| BusinessDelieryblserviceImpl.getById | 语法 | public BusinessDeliery getById(int id) |
| 前置条件 | 需要根据id查询BusinessDeliery时 |
| 后置条件 | 调用BusinessDelieryDataService的getById方法 goodsDataservice，collectionDatas的getById方法， |
| BusinessDelieryblserviceImpl.add | 语法 | public BusinessDeliery add(BusinessDeliery BusinessDeliery) |
| 前置条件 | 需要新增一个BusinessDeliery时 |
| 后置条件 | 调用 goodsDataservice，collectionDatas的insert方法，  获得新增的其他实体类的主键id，设置BusinessDeliery对应的外键的属性的值，调用BusinessDelieryDataService的insert方法，获得新增的BusinessDeliery的主键，根据该主键id查询BusinessDeliery并返回这个新增的BusinessDeliery |
| BusinessDelieryblserviceImpl.update | 语法 | public BusinessDeliery update(BusinessDeliery BusinessDeliery) |
| 前置条件 | 需要更新BusinessDeliery时 |
| 后置条件 | 调用 goodsDataservice，collectionDatas的update方法，  调用BusinessDelieryDataService的update方法，获得返回的BusinessDeliery的主键，根据该主键id查询BusinessDeliery并返回这个BusinessDeliery |
| BusinessDelieryblserviceImpl.getCourier | 语法 | public Vector<User> getCourier() |
| 前置条件 | 需要获得BusinessDeliery中collection的快递员user时 |
| 后置条件 | 调用userDataservice的getUserOfJob方法，获得返回的user集合，说明：该方法本应该是collectionblservice的方法，但BusinessDelieryblservice也需要这个方法，写在这里可以节省代码 |
| BusinessDelieryblserviceImpl.delete | 语法 | public boolean delete(int id) |
| 前置条件 | 需要删除一个BusinessDeliery时 |
| 后置条件 | 调用BusinessDelieryDataservice的delete方法 |

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| 服务名 | 服务 |
| BusinessDelieryDataservice.getList() | 获得所有的BusinessDeliery持久化对象 |
| BusinessDelieryDataservice.getById() | 根据id获得BusinessDeliery持久化对象 |
| BusinessDelieryDataservice.insert() | 增加一个BusinessDeliery持久化对象 |
| BusinessDelieryDataservice.update() | 更新一个BusinessDeliery持久化对象 |
| BusinessDelieryDataservice.delete() | 删除一个BusinessDeliery持久化对象 |
|  |  |
|  |  |
| goodsDataservice.getList() | 获得所有的goods持久化对象 |
| goodsDataservice.getById() | 根据id获得goods持久化对象 |
| goodsDataservice.insert() | 增加一个goods持久化对象 |
| goodsDataservice.update() | 更新一个goods持久化对象 |
| goodsDataservice.delete() | 删除一个goods持久化对象 |
|  |  |
| collectionDataservice.getList() | 获得所有的collection持久化对象 |
| collectionDataservice.getById() | 根据id获得collection持久化对象 |
| collectionDataservice.insert() | 增加一个collection持久化对象 |
| collectionDataservice.update() | 更新一个collection持久化对象 |
| collectionDataservice.delete() | 删除一个collection持久化对象 |
|  |  |

### 1.2.3派件单动态模型

以新增派件单BusinessDeliery为例，当用户保存时BusinessDeliery主键为空，则新增操作，调用调用add方法，调用BusinessDelieryblservice的add方法，调用各个Dataservice的insert方法，通过数据库insert每个PO对象，返回其主键id。



### 1.2.4寄件单模块设计原理

利用分层的结构，和委托式的控制风格，将其业务逻辑独立出来，通过不同的数据管理对象管理数据，实现了高内聚低耦合的特点。

**1.2.5度量计算部分：**

BusinessDeliverypanel 这个类调用的只有BusinessDeliveryblservice的六个方法，和一个BusinessDeliveryblservice对象引用

CBD=6、DAC=1。继承一个commonpanel 无子类，NOC=0，DIT=1

BusinessDeliveryblServic有BusinessDeliveryDataservice,goodsDataservice,collectionDataservice三个对象引用，和这个对象的三个方法。

CBO=9,DAC=1。BusinessDeliveryblseviceImpl实现该接口，NOC=1，DIT=0.