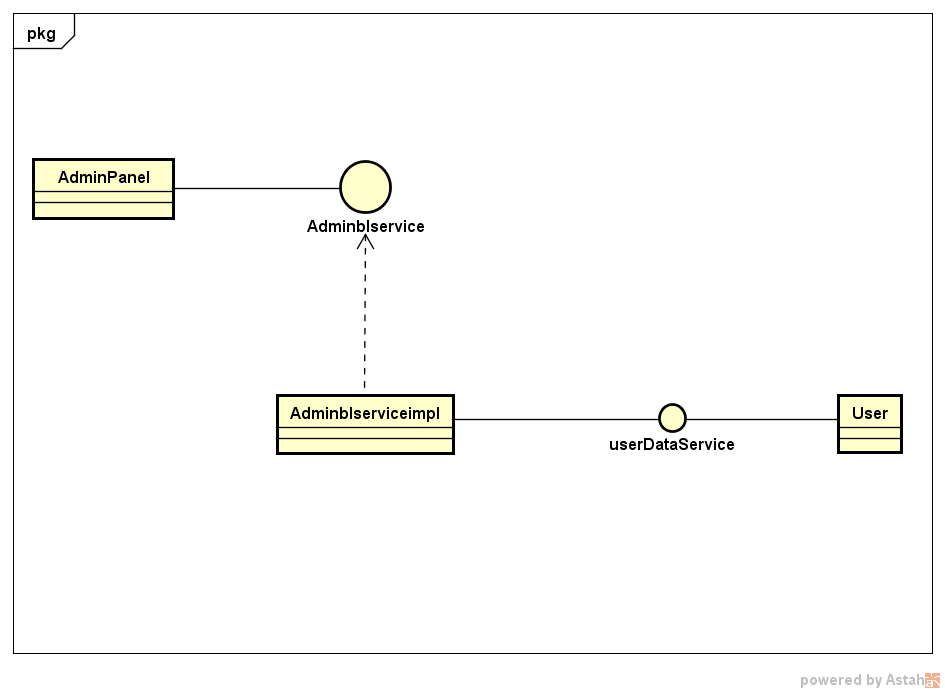
**1.2管理员模块设计文档**

**1.2.1说明**

管理员的用户权限管理界面AdminPanel需要在构造方法里面传入管理员业务逻辑接口Userblservice .Userblservice定义了UserData以及User的权限

其具体实现类Userblserviceimpl需要在构造方法中传入多个数据接口

**1.2.2用户权限管理详细设计描述**



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

**postPanel的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| AdminPanel.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 |
| AdminPanel.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UserData显示在表格中 |
| AdminPanel.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UserData显示在表格中 |
| **提供的服务（供接口）** | | |
| PostPanel.clear | 语法 | private void clear() |
| 前置条件 | 用户点击清空按钮，或者post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 |
| AdminPanel.save | 语法 | private void save() |
| 前置条件 | 用户点击保存按钮 |
| 后置条件 | 如果UserId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 |
| AdminPanel.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Userblservice的add方法，然后clear清空 |
| AdminPanel.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post的外键值，调用Adminblservice的update方法，然后clear清空 |
| AdminPanel.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post的id，调用Adminblservice的delete方法，然后clear清空 |
| AdminPanel.getUser | 语法 | private Post getPost() |
| 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User对象 |

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

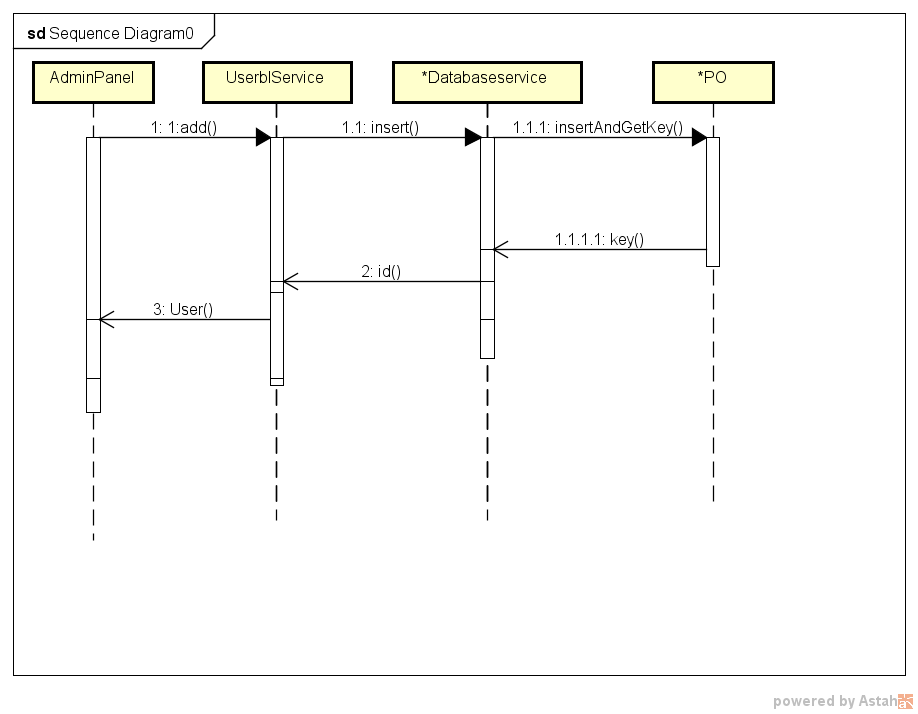
Admin**blserviceImpl的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| AdminblserviceImpl.getAll | 语法 | public Vector<Post> getAll() |
| 前置条件 | 需要查询所有的User时 |
| 后置条件 | 调用UserDataService的getList方法，对获得的每一个User调用UserDataDataservice的getById方法 |
| AdminblserviceImpl.getById | 语法 | public Post getById(int id) |
| 前置条件 | 需要根据id查询User时 |
| 后置条件 | 调用UserDataService的getById方法 |
| AdminblserviceImpl.add | 语法 | public User add(Post post) |
| 前置条件 | 需要新增一个post时 |
| 后置条件 | 调用UserDataService的insert方法，  获得新增的其他实体类的主键id，设置User对应的外键的属性的值，调用UserDataService的insert方法，获得新增的User的主键，根据该主键id查询User并返回这个新增的User |
| AdminblserviceImpl.update | 语法 | public Post update(Post post) |
| 前置条件 | 需要更新User时 |
| 后置条件 | 调用senderDataService，receiverDataservice，goodsDataservice，collectionDatas的update方法，  调用postDataService的update方法，获得返回的post的主键，根据该主键id查询post并返回这个post |
| AdminblserviceImpl.delete | 语法 | public boolean delete(int id) |
| 前置条件 | 需要删除一个User时 |
| 后置条件 | 调用UserDataservice的delete方法 |

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

**1.2.3 动态模型**

新增一个User，调用add方法，调用UserDataService的insert方法，通过数据库insert每个PO对象，返回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。