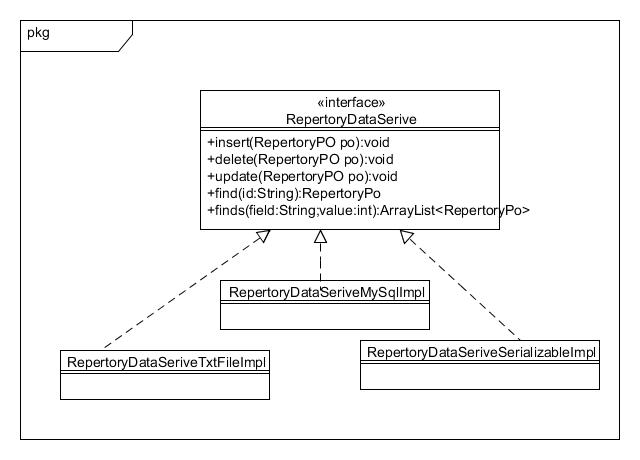
5.4 数据层的分解

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增删改查。User业务逻辑需要的服务由UserDataService接口提供。数据层的具体描述如图所示



5.4.1数据层模块的职责

如表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| UserDataService | 持久化数据库的接口，提供集体载入、集体保存、增删改查服务 |
| UserDataServiceTxtFileImpl | 基于Txt文件的持久化数据的接口，提供集体载入、集体保存、增删改查服务 |
| UserDataServiceSerializableFileImpl | 基于序列化文件的持久化数据的接口，提供集体载入、集体保存、增删改查服务 |
| UserDataServiceMySqlImpl | 基于Txt文件的持久化数据的接口，提供集体载入、集体保存、增删改查服务 |

5.4.2数据层模块的接口规范

数据层模块的接口规范如下表所示

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DatabaseFactory.getUserDatabase | 语法 | public UserPO getUserDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 得到User数据库的服务的引用 |
| UserDataService.insert | 语法 | public void insert(UserPO po) throws RemoteException |
| 前置条件 | 同样ID的po在原本的数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| RepertoryDataService.find | 语法 | public RepertoryPO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回响应的RepertoryPO结果 |
| RepertoryDataService.insert | 语法 | public void insert(RepertoryPO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| RepertoryDataService.delete | 语法 | public void delete(RepertoryPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 删除一个po |
| RepertoryDataService.update | 语法 | public void update(RepertoryPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 更新一个po |
| RepertoryDataService.init | 语法 | public void init() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| RepertoryDataService.finish | 语法 | public void finish() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |
| DatabaseFactory.getRepertoryDatabase | 语法 | public RepertoryPO getRepertoryDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 得到User数据库的服务的引用 |
| FinanceDataService.find | 语法 | public FinancePO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回响应的FinancePO结果 |
| FinanceDataService.insert | 语法 | public void insert(FinancePO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| FinanceDataService.delete | 语法 | public void delete(FinancePO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 删除一个po |
| FinanceDataService.update | 语法 | public void update(FinancePO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 更新一个po |
| FinanceDataService.init | 语法 | public void init() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| FinanceDataService.finish | 语法 | public void finish() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |
| DatabaseFactory.getFinanceDatabase | 语法 | public FinancePO getFinanceDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 得到User数据库的服务的引用 |
| ConfigurationDataService.find | 语法 | public ConfigurationPO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回响应的ConfigurationPO结果 |
| ConfigurationDataService.insert | 语法 | public void insert(ConfigurationPO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| ConfigurationDataService.delete | 语法 | public void delete(ConfigurationPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 删除一个po |
| ConfigurationDataService.update | 语法 | public void update(ConfigurationPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 更新一个po |
| ConfigurationDataService.init | 语法 | public void init() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| ConfigurationDataService.finish | 语法 | public void finish() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |
| DatabaseFactory.getConfigurationDatabase | 语法 | public ConfigurationPO getConfigurationDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 得到User数据库的服务的引用 |
| ReceiptDataService.find | 语法 | public ReceiptPO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回响应的ReceiptPO结果 |
| ReceiptDataService.insert | 语法 | public void insert(ReceiptPO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| ReceiptDataService.delete | 语法 | public void delete(ReceiptPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 删除一个po |
| ReceiptDataService.update | 语法 | public void update(ReceiptPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 更新一个po |
| ReceiptDataService.init | 语法 | public void init() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| ReceiptDataService.finish | 语法 | public void finish() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |
| DatabaseFactory.getReceiptDatabase | 语法 | public ReceiptPO getReceiptDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 得到User数据库的服务的引用 |