1. 业务逻辑层接口规范

|  |  |  |
| --- | --- | --- |
| **提供的业务（供接口）** | | |
| User.login | 语法 | public ResultMessage login(String id,String password) |
| 前置条件 | password符合输入规则 |
| 后置条件 | 查找是否存在相应的User，根据输入的password返回登录验证的结果 |
| User.addUser | 语法 | public ResultMessage addUser(String id,Sting name,String identity，String password) |
| 前置条件 | 启动一个用户添加回合 |
| 后置条件 | 检验是否已经存在此id的用户，若否，创建该用户；持久化更新涉及的领域对象的数据 |
| User.deleteUser | 语法 | public ResultMessage deleteUser(String id) |
| 前置条件 | 启动一个用户删除回合 |
| 后置条件 | 检验是否存在此id的用户，若是，删除此用户信息，否则拒绝删除；持久化更新涉及的领域对象的数据 |
| User.findChangingUser | 语法 | public ResultMessage findChangingUser(String id) |
| 前置条件 | 启动一个用户修改回合 |
| 后置条件 | 查找是否存在相应的User，返回查找结果 |
| User.changeUser | 语法 | public ResultMessage changeUser(String id, Sting name,String identity,String password) |
| 前置条件 | 已查找到存在此id的用户 |
| 后置条件 | 根据输入，修改此用户信息；持久化更新涉及的领域对象的数据 |
| Book.findByKey | 语法 | public ArrayList<Book> find(String name) |
| 前置条件 | 已经正确输入了关键字 |
| 后置条件 | 返回相应的图书列表 |
| Book.findByType | 语法 | public ArrayList<Book> findByType(String type) |
| 前置条件 | 已经正确选择了图书类别 |
| 后置条件 | 返回相应的图书列表 |
| Book.addBook | 语法 | public ResultMessage addBook(Book b) |
| 前置条件 | 已经正确输入了新增图书的信息 |
| 后置条件 | 添加新增图书，持久化更新涉及的领域对象的数据 |
| Book.deleteBook | 语法 | public ResultMessage deleteBook(String id) |
| 前置条件 | 已经正确输入了待删除图书的信息 |
| 后置条件 | 将图书删除 |
| Book.modifyBook | 语法 | public Book modifyBook(String id) |
| 前置条件 | 已经正确输入了待修改图书的信息 |
| 后置条件 | 返回该图书的所有信息，持久化更新涉及的领域对象的数据 |
| Member.find | 语法 | public Member find(String id) |
| 前置条件 | 输入的会员ID符合规则 |
| 后置条件 | 返回相应的会员信息 |
| Member.addEquivalent | 语法 | public ResultMessage addEquivalent(Equivalent e) |
| 前置条件 | 已经制定了等价券赠送规则，并且输入的等价券信息符合规则 |
| 后置条件 | 增加会员的等价券信息，持久化更新涉及的领域对象的数据 |
| Member.addCoupon | 语法 | public ResultMessage addCoupon(Coupon c) |
| 前置条件 | 已经制定了折价券赠送规则，并且输入的打折券信息符合规则 |
| 后置条件 | 增加会员的折价券信息，持久化更新涉及的领域对象的数据 |
| Member.addFavorities | 语法 | public ResultMessage addFavorities(Book book) |
| 前置条件 | 已经找到想收藏的图书 |
| 后置条件 | 将图书加入收藏夹，持久化更新涉及的领域对象的数据 |
| Member.getOrderList | 语法 | public ArrayList<Order> getOrderList() |
| 前置条件 | 销售经理已经登录 |
| 后置条件 | 返回所有未签收的订单列表 |
| Member.changeOrder | 语法 | public ResultMessage changeOrder(State state) |
| 前置条件 | 启动一个修改订单状态回合，且输入的订单状态符合规则 |
| 后置条件 | 修改订单状态，持久化更新涉及的领域对象的数据 |
| Member.register | 语法 | public ResultMessage register(String name,String password,String telephone) |
| 前置条件 | 启动一个会员注册回合 |
| 后置条件 | 为注册者分配唯一的会员编号；持久化更新涉及的领域对象的数据 |
| Member. cancellation | 语法 | public ResultMessage cancellation(password) |
| 前置条件 | 此会员已经登录，且password符合输入规则 |
| 后置条件 | 检验password的正误，若正确，删除此会员信息，否则拒绝注销；持久化更新涉及的领域对象的数据 |
| Sale.calculate | 语法 | public Order calculate() |
| 前置条件 | 已经进入购物车 |
| 后置条件 | 返回相应的订单 |
| Sale.putInCart | 语法 | public ResultMessage putInCart(String id) |
| 前置条件 | 已经找到所需购买的图书， |
| 后置条件 | 将图书加入购物车，持久化更新涉及的领域对象的数据 |
| Sale.enterCart | 语法 | public void enterCart() |
| 前置条件 | 已经找到所有需要购买的图书，准备结算 |
| 后置条件 | 系统允许顾客进入购物车进行操作 |
| Sale.purchase | 语法 | public ResultMessage purchase(Order order) |
| 前置条件 | 已经进行过结算，并生成订单 |
| 后置条件 | 结束此次购买，持久化更新设计的领域对象的数据 |
| Sale.chooseEquivalent | 语法 | public double chooseEquivalent (String id) |
| 前置条件 | 已经进入购物车，并持有等价券 |
| 后置条件 | 重新计算图书总价 |
| Sale.chooseCupon | 语法 | public double chooseCoupon (String id) |
| 前置条件 | 已经进入购物车，并持有打折券 |
| 后置条件 | 重新计算图书总价 |
| Sale.removeFromCart | 语法 | public void removeFromCart(String id) |
| 前置条件 | 顾客已经进入购物车 |
| 后置条件 | 从购物车删除所选图书，持久化更新涉及的领域对象的数据 |
| Sale.endSale | 语法 | public void endSale () |
| 前置条件 | 已提交订单 |
| 后置条件 | 结束此次购买，持久化更新涉及的领域对象的数据 |
| Promotion.selectPromotionType | 语法 | public ResultMessage selectPromotionType() |
| 前置条件 | 启动一个促销手段设置回合 |
| 后置条件 | 选定一个促销手段，并得到该促销手段的信息 |
| Promotion.setPromotion | 语法 | public ResultMessage setPromotion(double number) |
| 前置条件 | 已经选定了一个促销手段 |
| 后置条件 | 设置具体的促销策略 |
| Promotion.selectLevelOfVIP | 语法 | public ResultMessage selectLevelOfVIP() |
| 前置条件 | 启动大范围促销服务 |
| 后置条件 | 选定一个会员积分等级 |
| Promotion.setPresent | 语法 | public ResultMessage setPresent(String vipLevel,int number) |
| 前置条件 | 已经选定了一个会员积分等级 |
| 后置条件 | 完成对该积分等级的每个会员的礼券赠送信息设置 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| DataService.find(String id) | 根据ID 进行查找单一持久化对象 | |
| DataService.update | 更新单一持久化对象 | |
| DataService.insert | 插入单一持久化对象 | |
| DataSerivce.delete | 删除单一持久化对象 | |
| DatabaseFactory.getBookDatabase | 得到Book数据库的服务的引用 | |
| DatabaseFactory.getPromotionDatabase | 得到Promotion数据库的服务的引用 | |
| DatabaseFactory.getMemberDatabase | 得到Member数据库的服务的引用 | |

1. 数据层接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| DateService.find | 语法 | public PO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 根据ID进行查找返回相应的po结果 |
| DataService.insert | 语法 | public void insert(PO po) throws RemoteException |
| 前置条件 | 同样ID的PO在Mapper中不存在 |
| 后置条件 | 在Mapper增加一个记录 |
| DataService.delete | 语法 | public void delete(PO po) throws RemoteException |
| 前置条件 | 在Mapper中存在同样ID的po |
| 后置条件 | 删除一个po |
| DataService.update | 语法 | public void update(PO po) throws RemoteException |
| 前置条件 | 在Mapper中存在同样ID的po |
| 后置条件 | 更新一个po |
| DataService.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 |
| DataService.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 |
| DataFactory. getMemberDatabase | 语法 | public void getMemberDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |
| DataFctory. getBookDatabase | 语法 | public void getBookDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |
| DataFactory. getPromotionDatabase | 语法 | public void getPromotionDatabase ()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |