**网上图书购买系统**

**详细设计文档**

V2.0

2012-12-09

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 陈逸凡、赵玉峰、章许帆、周梦婷 | 2012-10-20 | 最初草稿 | V1.0 初稿 |
| 章许帆、周梦婷 | 2012-12-09 | 类细化 | V2.0 |
|  |  |  |  |

**1引言**

**1.1编制目的**

本报告详细完成对网上图书购买系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

**1.2词汇表**

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
|  |  |  |
|  |  |  |
|  |  |  |

**1.3参考资料**

1.网上图书购买系统用例文档V2.0

**2产品概述**

网上图书购买系统是一个自主开发的业务系统，开发的目标是为了给顾客提供在网上购买图书的服务，重点业务包括图书销售、会员发展、库存管理和商品促销。

**3体系结构设计概述**

网上图书购买系统采用分层式结构进行开发，包括界面层，业务逻辑层，数据层。

**4结构视角**

**4.1 业务逻辑层的分解**

业务逻辑层的开发包图参见软件体系结构文档图3。

**4.1.1 促销模块**

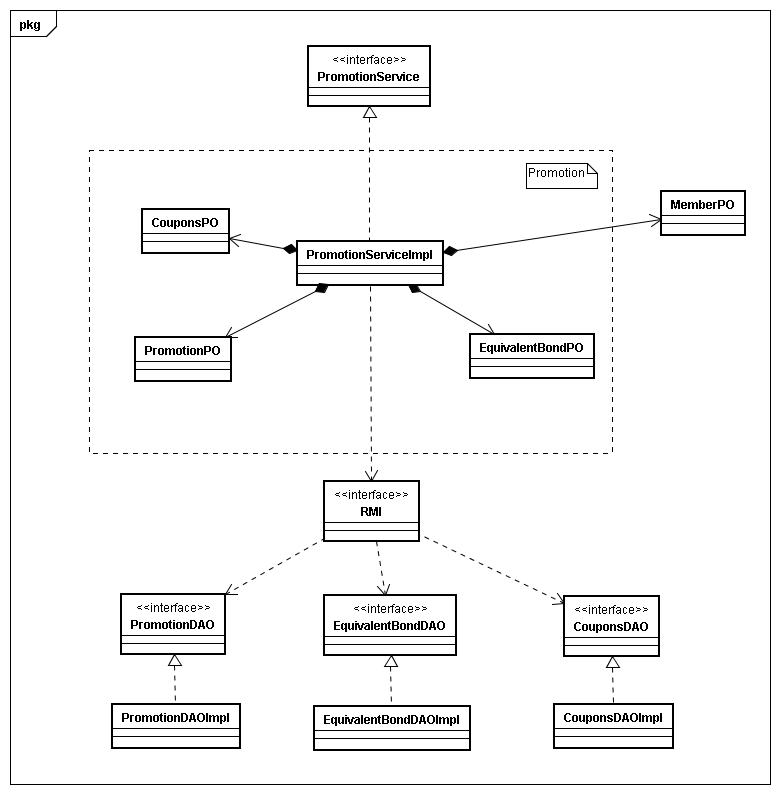
**一、 模块概述**

promotion模块负责处理总经理设置促销策略和销售经理赠送优惠券等需求。

promotion模块的职责及接口参照网上图书购买系统体系结构设计文档。

**二、 整体结构**

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加接口promotionService。业务逻辑层和数据层之间添加promotionDAO,promotionHistoryDAO等接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了promotionServiceImpl，这样promotionServiceImpl会将对促销的业务逻辑处理委托给promotion对象。PromotionPO是作为促销策略的持久化对象被添加到设计模型中去的。同样为了隔离数据操作职责和调用数据职责，我们增加了相应领域对象的DAO接口。

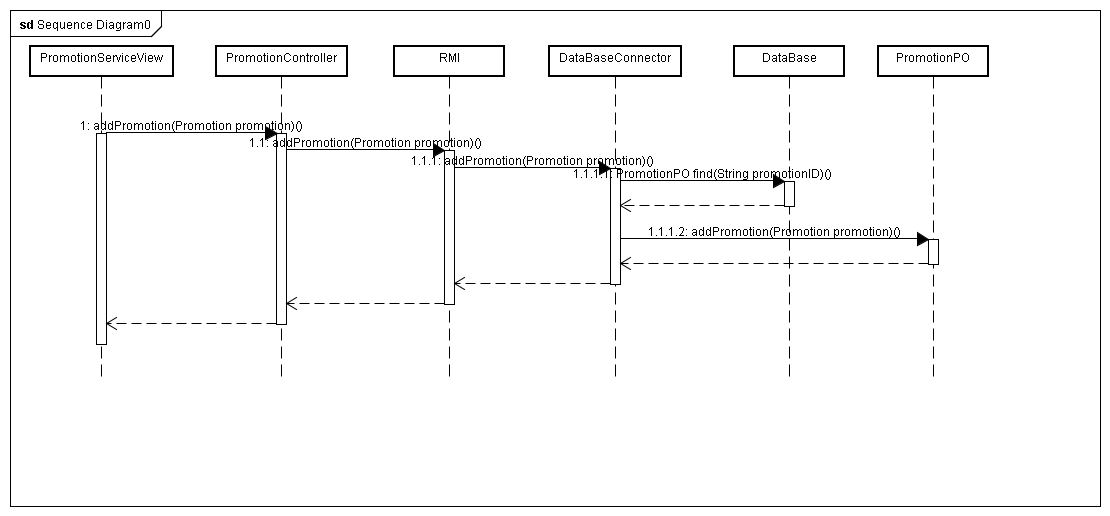


Promotion模块设计

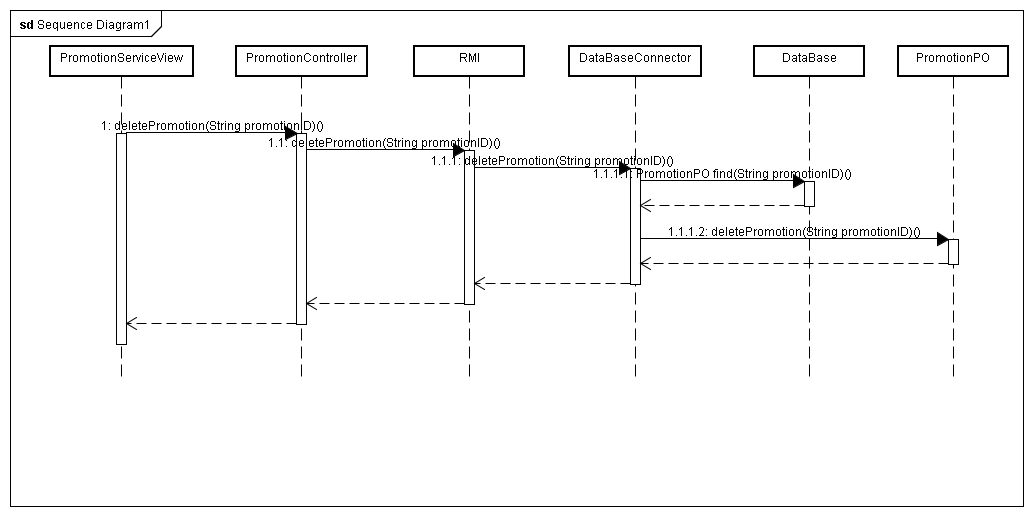
**三、模块内部类的接口规范**

|  |  |  |
| --- | --- | --- |
| 1. 提供的服务（供接口） | | |
| PromotionService.addPromotion | 语法 | public ResultMessage addPromotion(PromotionPO promotionPO  ) throws RemoteException |
| 前置条件 | 总经理制定了新的促销方案 |
| 后置条件 | 根据促销方案查找是否存在相应的促销方案，验证不存在后将新增加的促销方案进行持久化保存 |
| PromotionService.deletePromotion | 语法 | public ResultMessage deletePromotion(int promotionID) throws RemoteException |
| 前置条件 | 总经理输入了想要删除的促销方案 |
| 后置条件 | 根据促销方案ID查找是否存在相应的促销方案，验证存在后从促销方案数据库中删除该促销方案 |
| PromotionService.getPromotionList | 语法 | Public ResultMessage getPromotionList() throws RemoteException |
| 前置条件 | 销售经理请求查看促销策略列表 |
| 后置条件 | 系统返回促销策略 |
| PromotionService.triggerPromotion | 语法 | public ResultMessage triggerPromotion(int memberID, int promotionID) throws RemoteException |
| 前置条件 | 销售经理根据总经理制定的促销方案实施促销策略 |
| 后置条件 | 根据顾客ID向顾客发送打折券和等价券 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| PromotionDAO.addPromotion(PromotionPO promotionPO) | 添加新促销策略 | |
| PromotionDAO.deletePromotion(int promotionID) | 根据ID删除该促销策略 | |
| PromotionDAO.queryPromotion(int promotionID) | 根据ID查询该促销策略详细信息 | |
| PromotionDAO.updatePromotion(PromotionPO promotionPO) | 更新单一持久化对象 | |
| PromotionDAO.getPromotionList() | 查找所有促销策略 | |

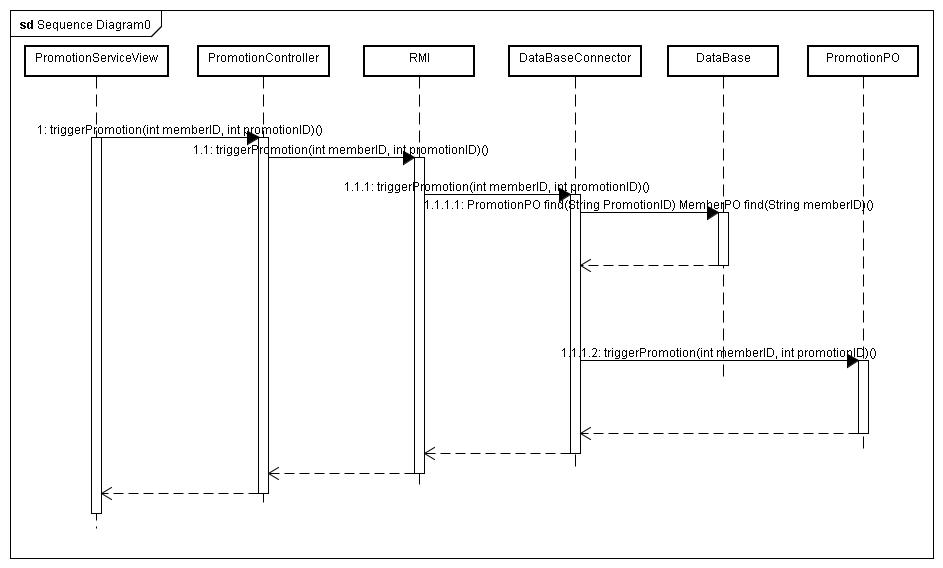
**四、业务逻辑层的动态模型**

****

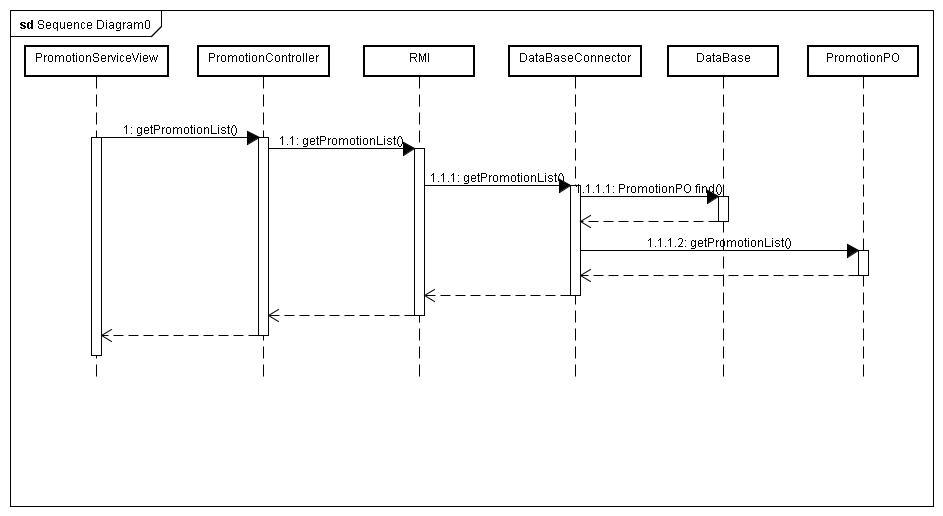
添加促销策略的顺序图

****

删除促销策略的顺序图



赠送优惠券的顺序图



获取促销策略列表的顺序图

**五、业务逻辑层的设计原理**

网上图书购买系统采用委托式的控制风格，每个界面需要访问的业务逻辑有各自的控制器委托给不同的领域对象。例如Promotion模块的界面需要的业务逻辑通过promotionservice委托给promotion、coupons、equivalentbond、message对象，通过将业务逻辑分解为对不同领域对象的操作及协作，并且委托给各自领域对象的持久化操作的数据操作接口DAO层，实现了委托式的控制。

**4.1.2 User模块**

**一、 模块概述**

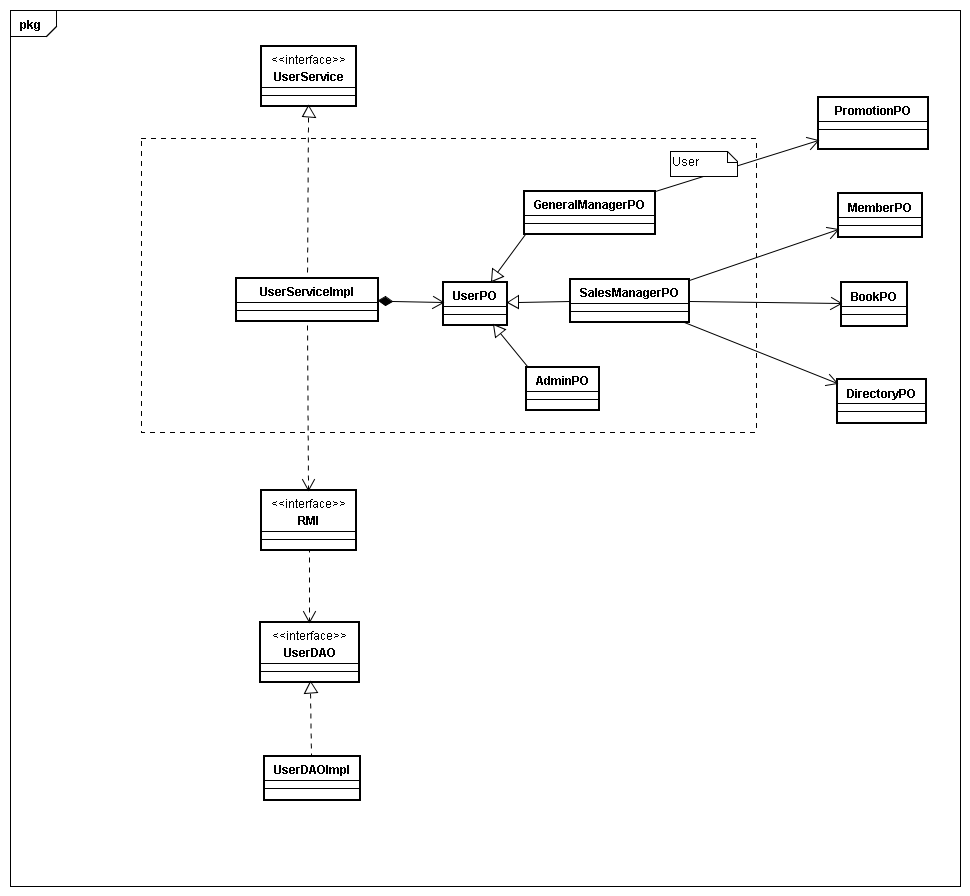
user模块负责处理系统管理员增删改用户，销售经理查看顾客信息等任务。

user模块的职责及接口参照网上图书购买系统体系结构设计文档。

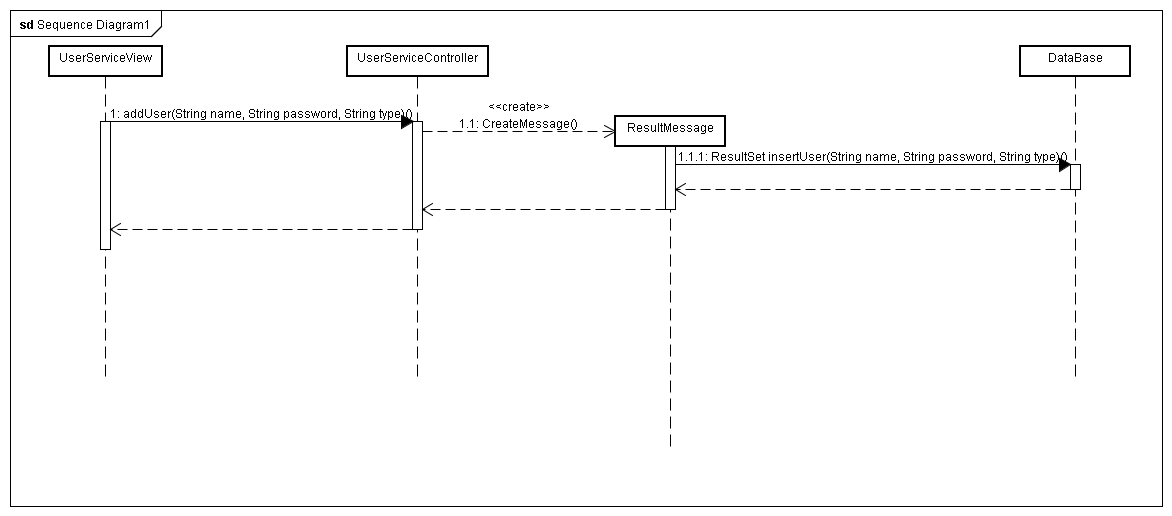
**二、 整体结构**

user展示层和user业务逻辑层之间，我们添加接口UserService。业务逻辑层和数据层之间添加UserDAO等接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了UserServiceImpl，这样UserServiceImpl会将对用户的业务逻辑处理委托给user对象。UserPO、AdminPO、SalesmanagerPO、GeneralmanagerPO是作为用户、系统管理员、销售经理、总经理的持久化对象被添加到设计模型中去的。同样为了隔离数据操作职责和调用数据职责，我们增加了相应领域对象的DAO接口。

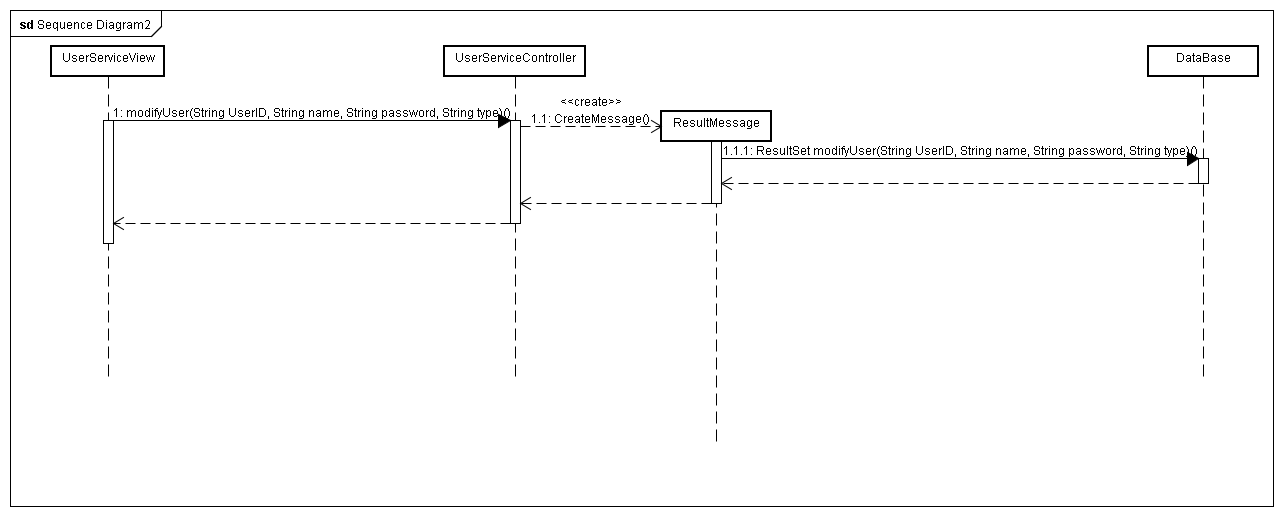
**三、模块内部类的接口规范**

****

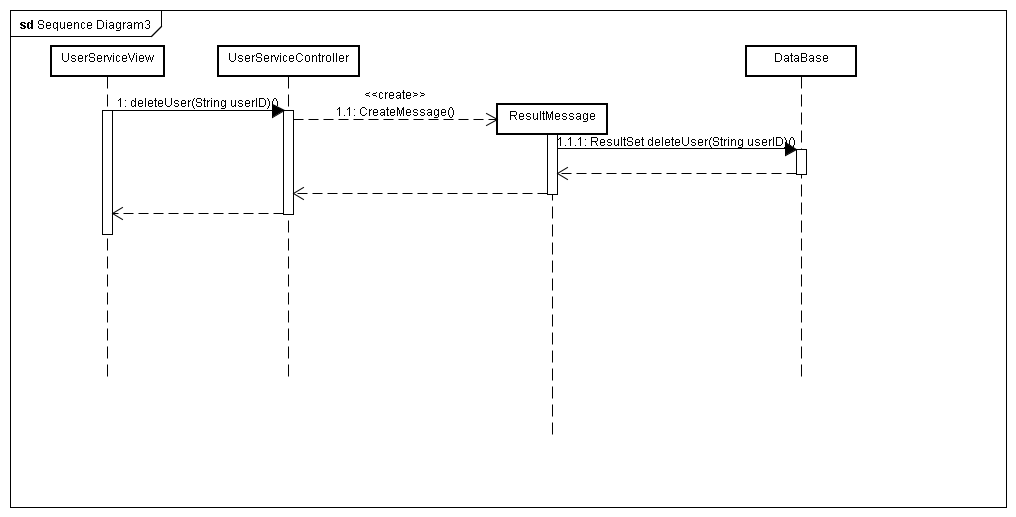
**四、业务逻辑层的动态模型**

****

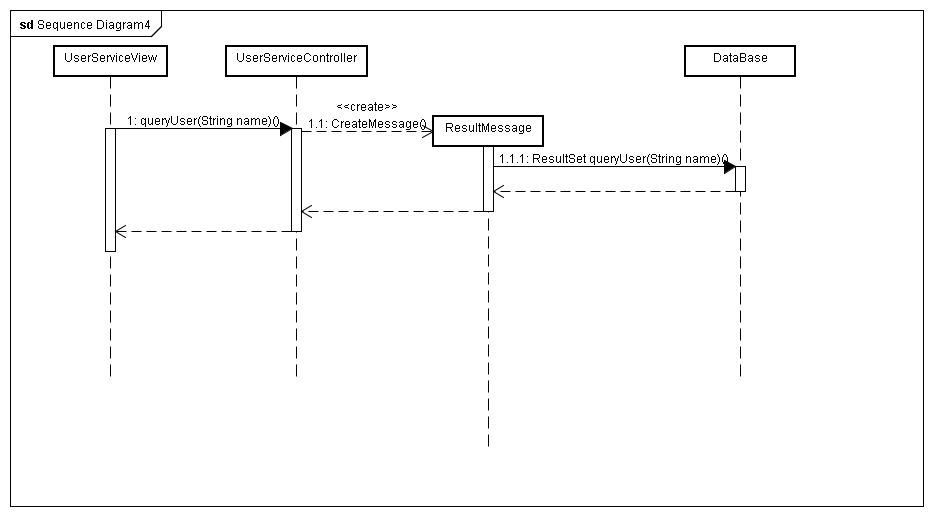
添加用户的顺序图

****

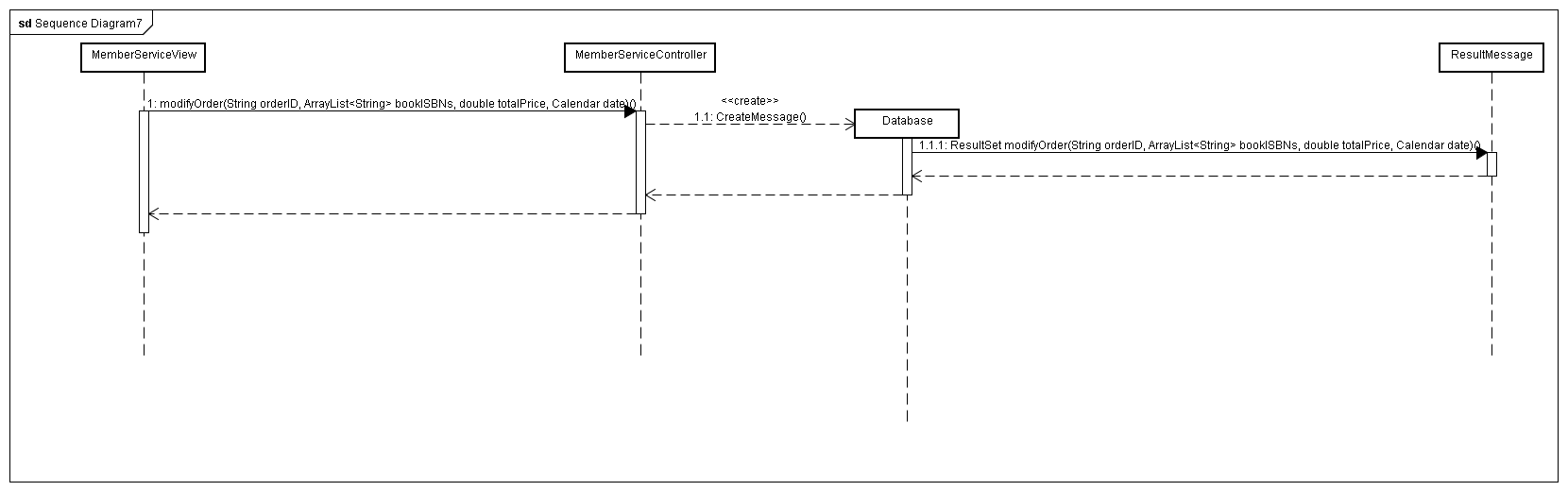
修改用户的顺序图

****

删除用户的顺序图

****

查找用户的顺序图

****

修改会员订单的顺序图

**五、业务逻辑层的设计原理**

User模块的界面需要的业务逻辑通过userservice委托给admin、salesmanager、generalmanager对象，通过将业务逻辑分解为对不同领域对象的操作及协作，并且委托给各自领域对象的持久化操作的数据操作接口DAO层，实现了委托式的控制。

**4.1.3 Member模块**

**一、 模块概述**

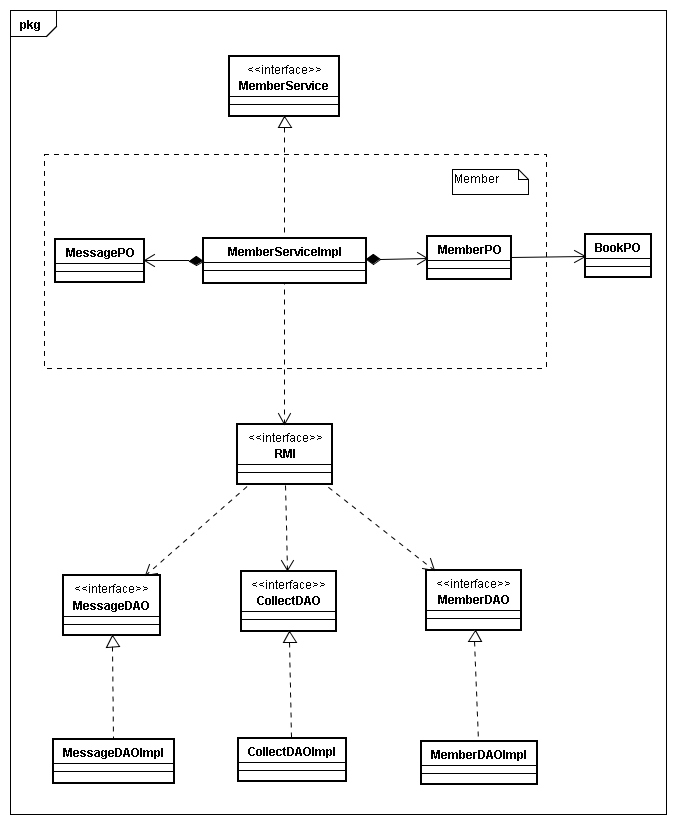
member模块负责处理用户登录，注册，查看修改个人信息，查看优惠券信息等任务。

member模块的职责及接口参照网上图书购买系统体系结构设计文档。

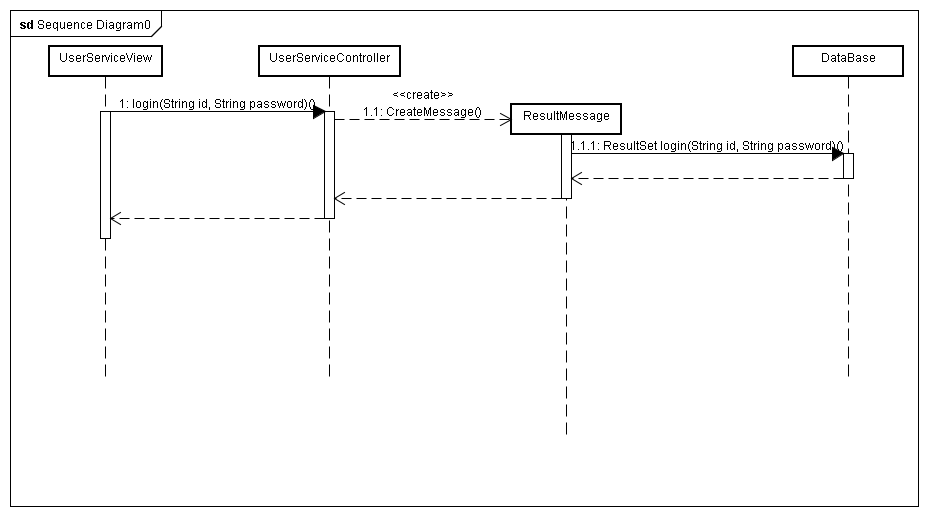
**二、 整体结构**

member展示层和member的业务逻辑层之间，我们添加接口MemberService。业务逻辑层和数据层之间添加MemberDAO等接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了MemberServiceImpl，这样MemberServiceImpl会将对会员的业务逻辑处理委托给member对象。MemberPO是作为会员的持久化对象被添加到设计模型中去的。同样为了隔离数据操作职责和调用数据职责，我们增加了相应领域对象的DAO接口。

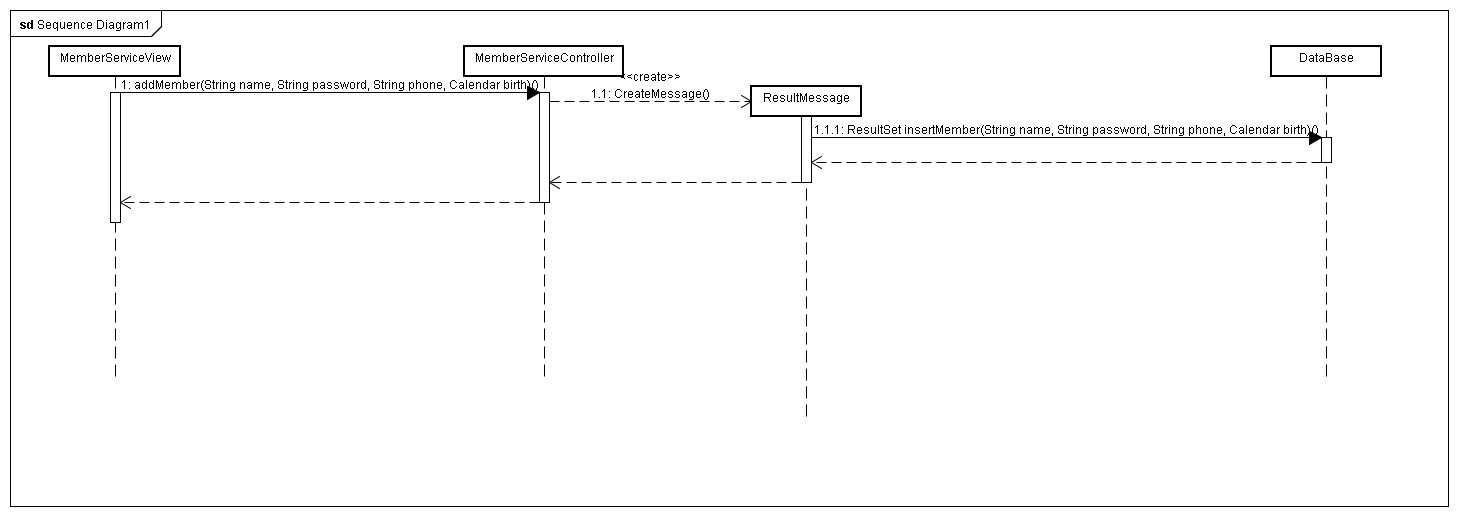
**三、模块内部类的接口规范**

****

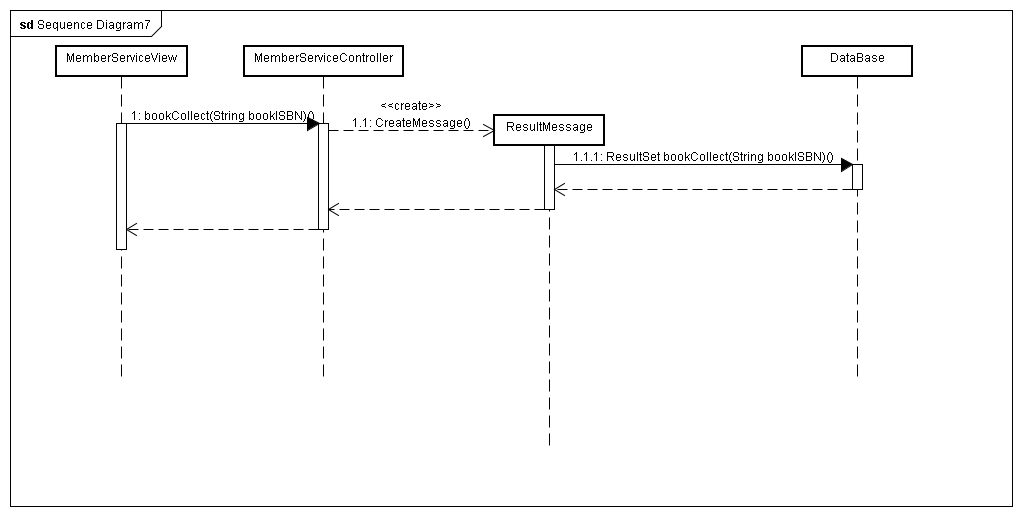
**四、业务逻辑层的动态模型**

****

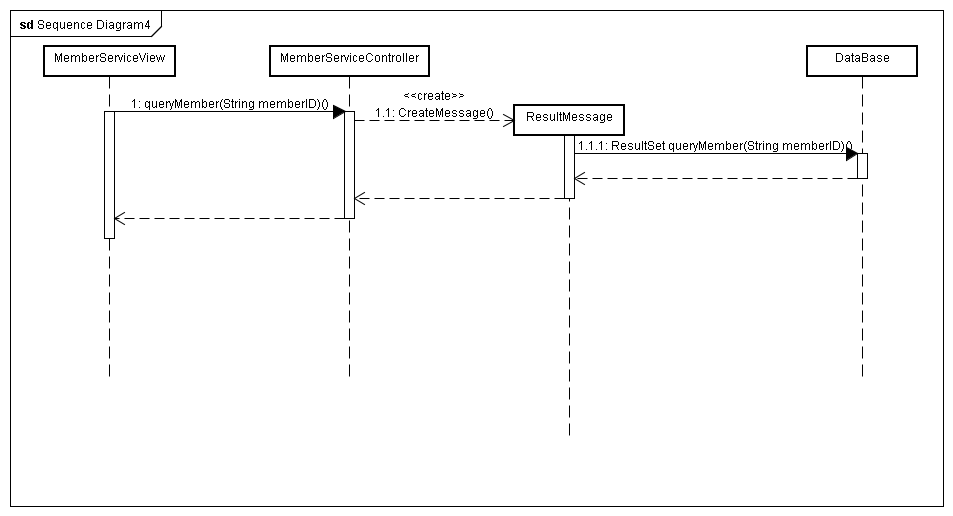
会员登陆的顺序图

****

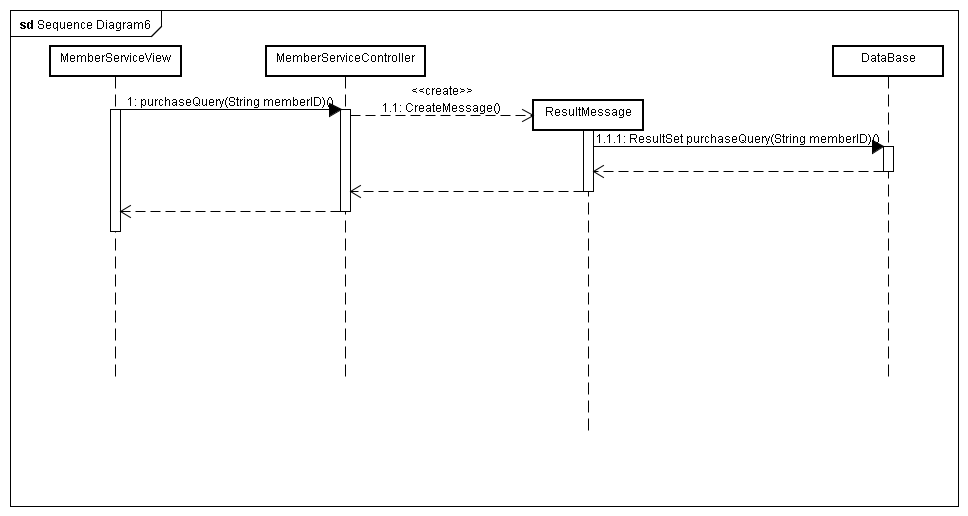
注册的顺序图

****

收藏图书的顺序图

****

查找会员的顺序图

****

查找购买记录的顺序图

**五、业务逻辑层的设计原理**

Member模块的界面需要的业务逻辑通过MemberService委托给member对象，通过将业务逻辑分解为对不同领域对象的操作及协作，并且委托给各自领域对象的持久化操作的数据操作接口DAO层，实现了委托式的控制。

**4.1.4 Sales模块**

**一、 模块概述**

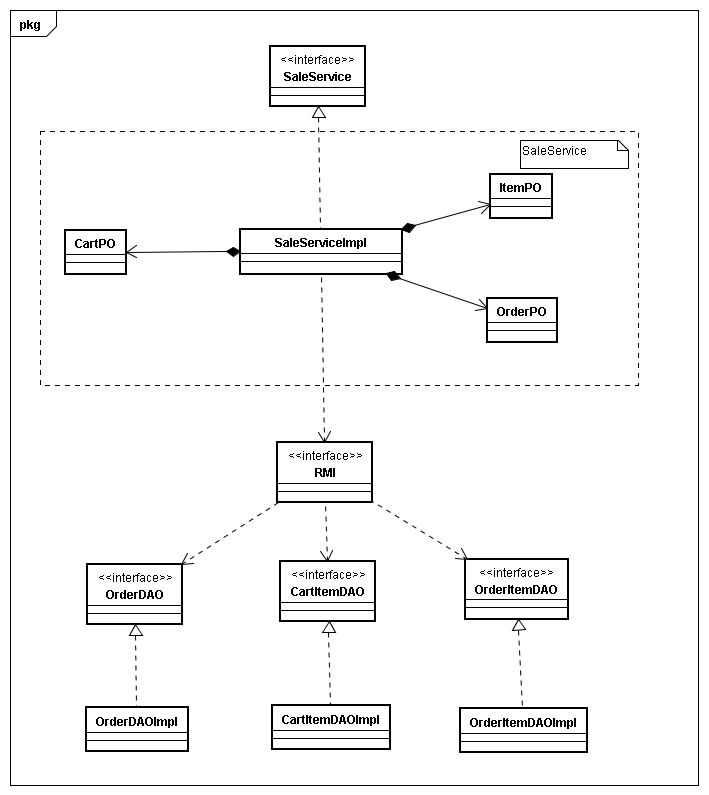
sales模块负责处理会员购买图书的一系列操作。

sales模块的职责及接口参照网上图书购买系统体系结构设计文档。

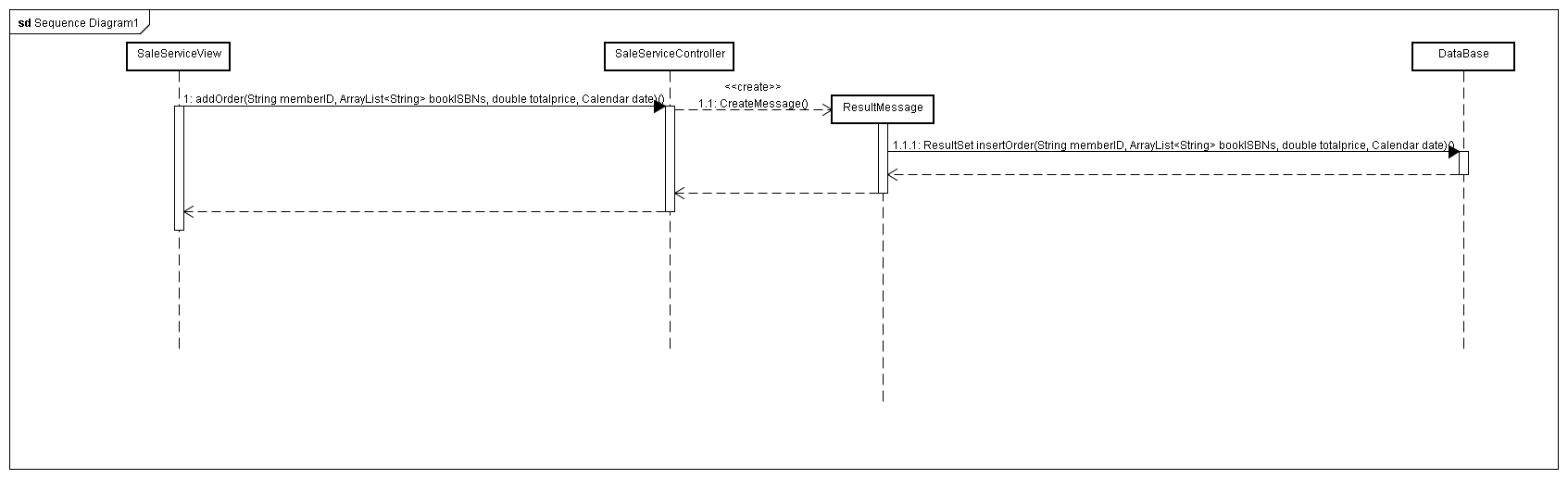
**二、**

sales展示层和sales业务逻辑层之间，我们添加接口SalesService。业务逻辑层和数据层之间添加SalesDAO等接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了SalesServiceImpl，这样SalesServiceImpl会将对购买的一系列操作的业务逻辑处理委托给cart，order等对象。CartPO、ItemPO、OrderPO是作为购物车、图书列表、订单的持久化对象被添加到设计模型中去的。同样为了隔离数据操作职责和调用数据职责，我们增加了相应领域对象的DAO接口。

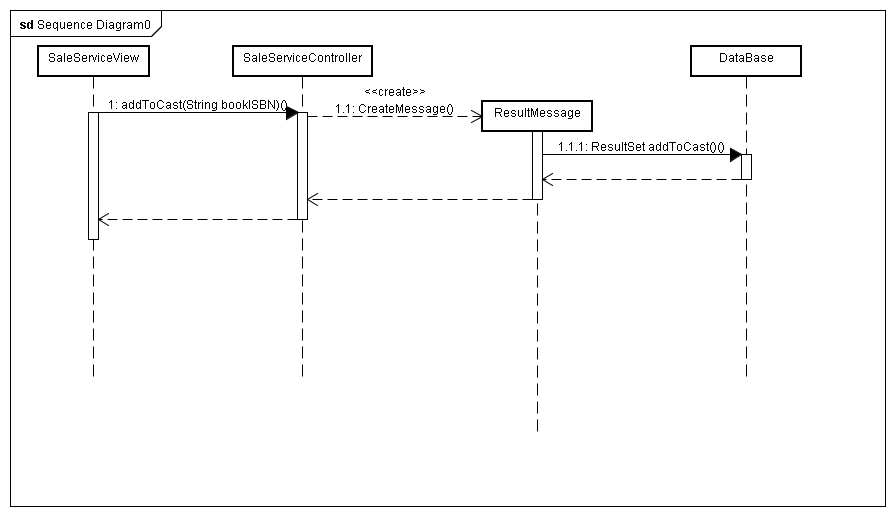
**三、模块内部类的接口规范**

****

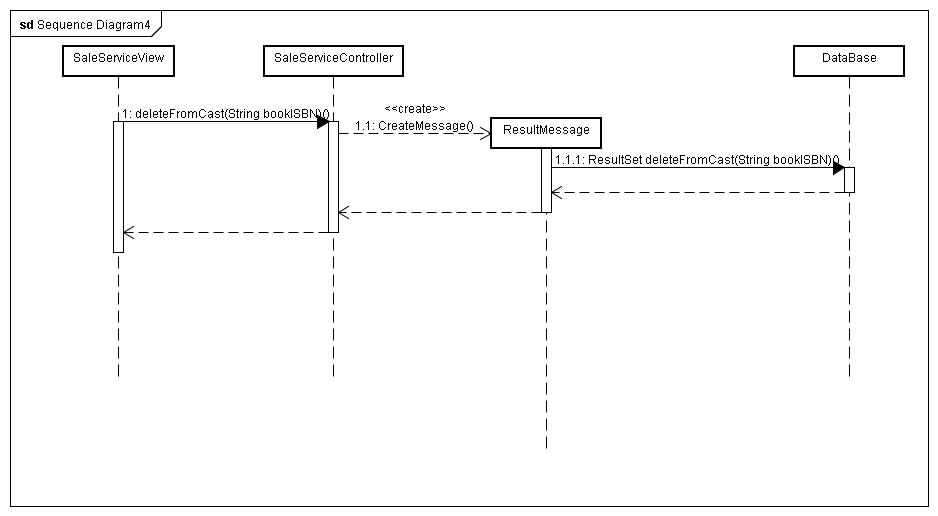
**四、业务逻辑层的动态模型**

****

添加订单的顺序图

****

图书加入购物车的顺序图

****

在购物车删除图书的顺序图

**五、业务逻辑层的设计原理**

Sales模块的界面需要的业务逻辑通过SalesService委托给cart、order等对象，通过将业务逻辑分解为对不同领域对象的操作及协作，并且委托给各自领域对象的持久化操作的数据操作接口DAO层，实现了委托式的控制。

**4.1.5 Book模块**

**一、 模块概述**

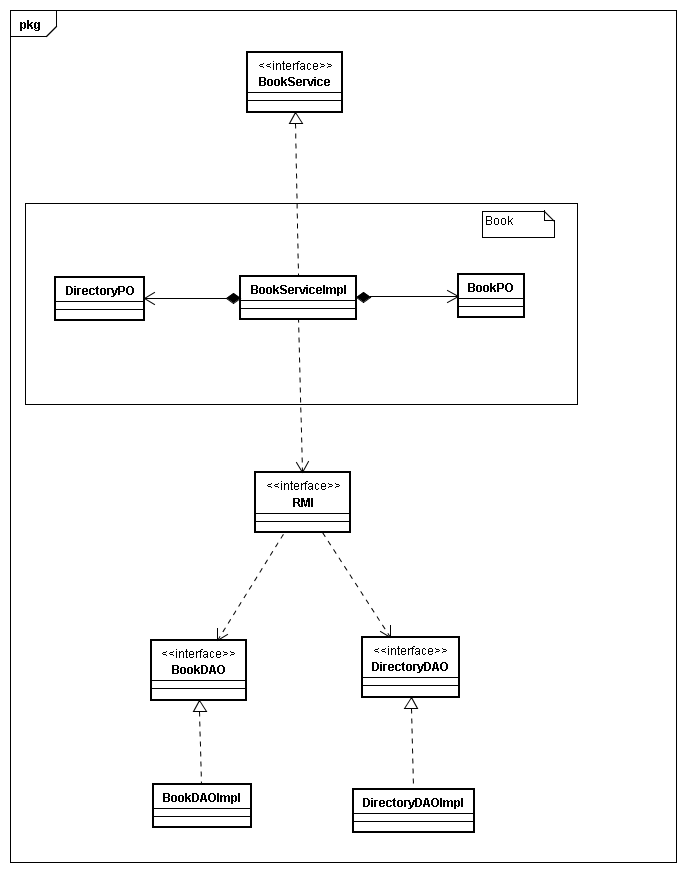
book模块负责处理图书增删改，图书分类增删改，查找图书信息等任务。

book模块的职责及接口参照网上图书购买系统体系结构设计文档。

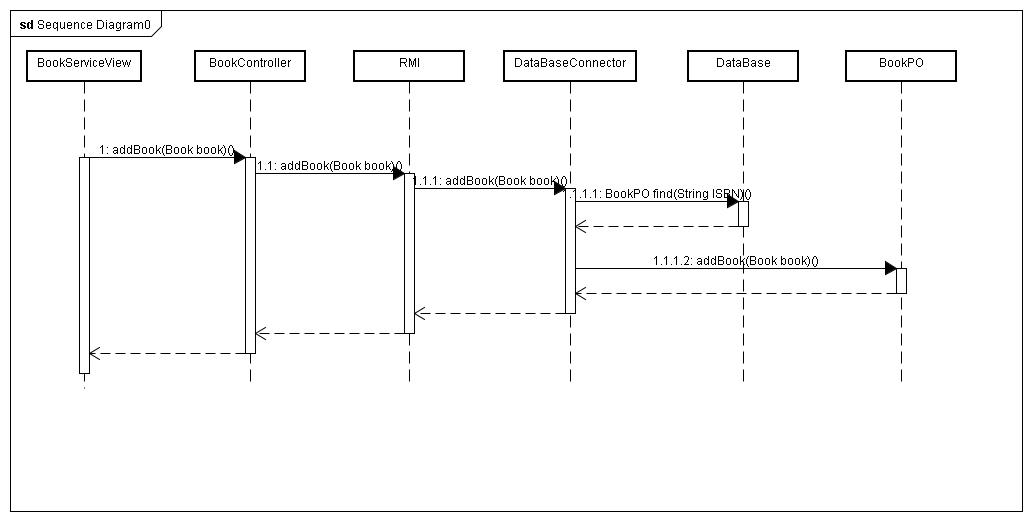
**二、 整体结构**

book展示层和book业务逻辑层之间，我们添加接口BookService。业务逻辑层和数据层之间添加BookDAO等接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了BookServiceImpl，这样BookServiceImpl会将对图书的业务逻辑处理委托给book对象。BookPO、BookClassifyPO是作为图书、图书类别的持久化对象被添加到设计模型中去的。同样为了隔离数据操作职责和调用数据职责，我们增加了相应领域对象的DAO接口。

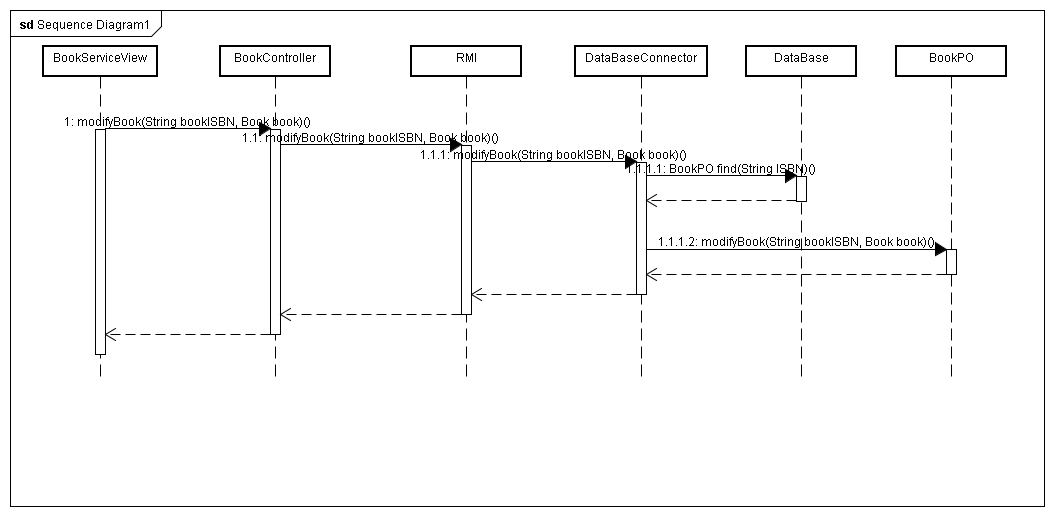
**三、模块内部类的接口规范**

****

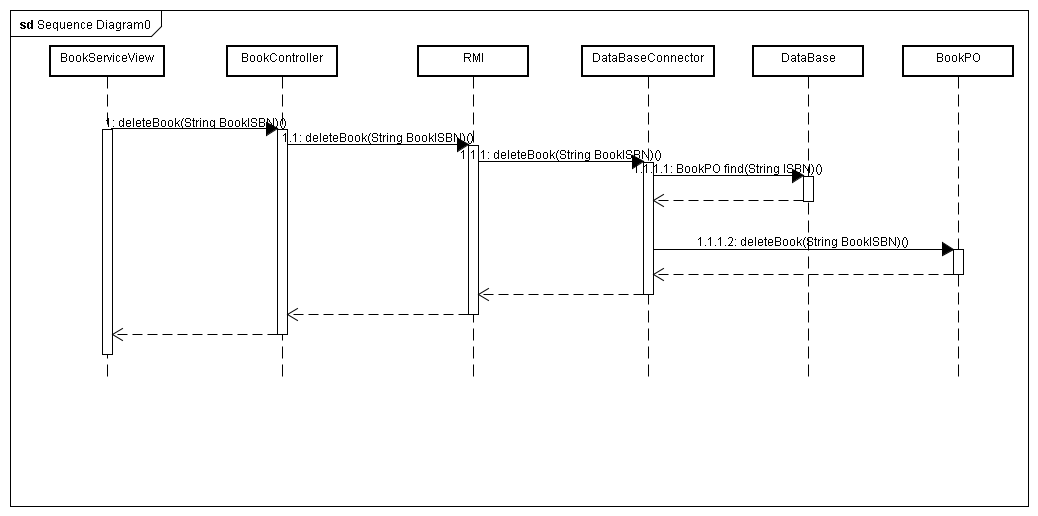
**四、业务逻辑层的动态模型**

****

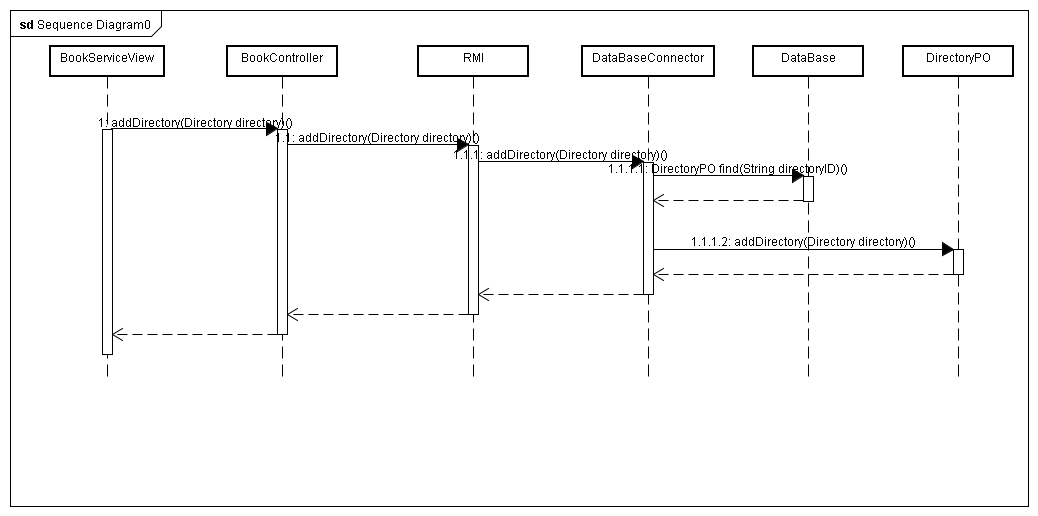
添加图书的顺序图

****

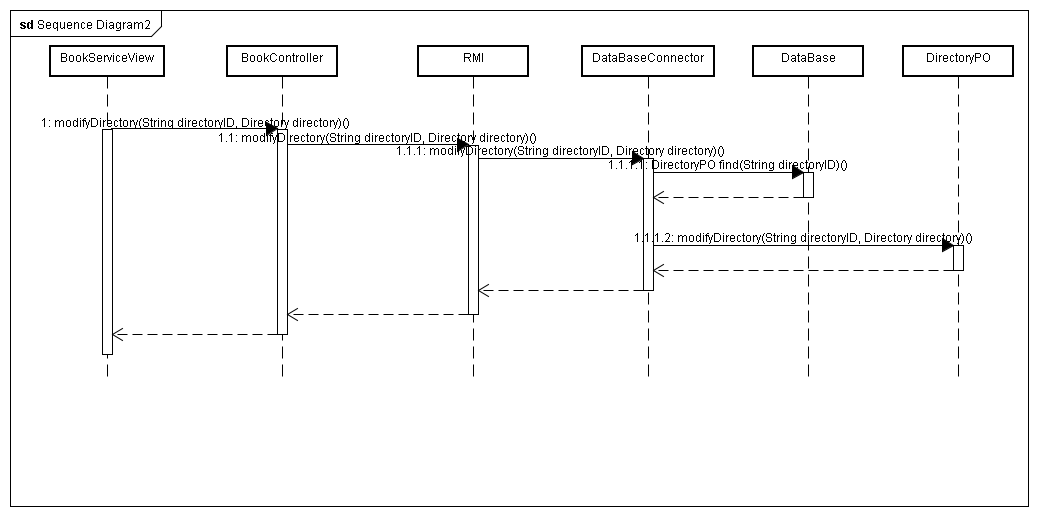
修改图书的顺序图

****

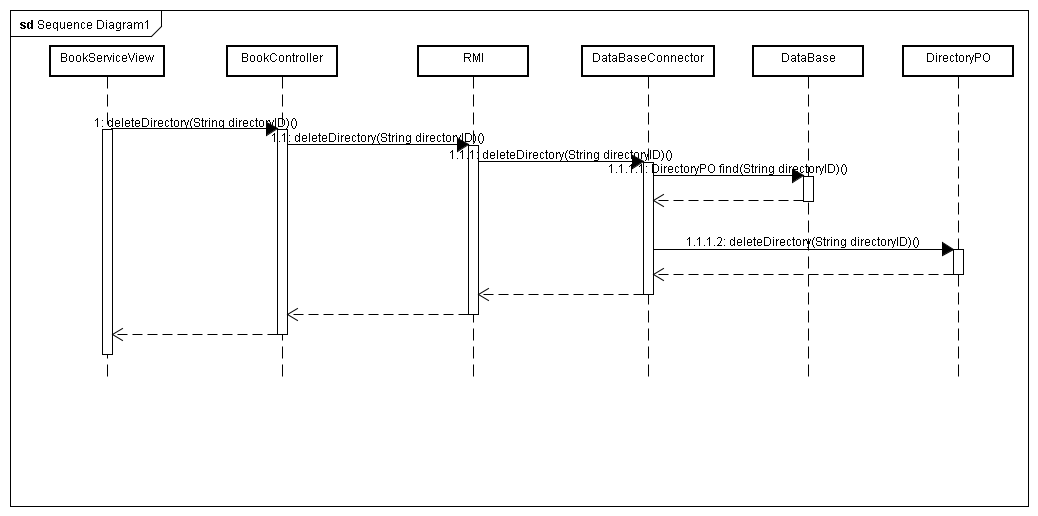
删除图书的顺序图

****

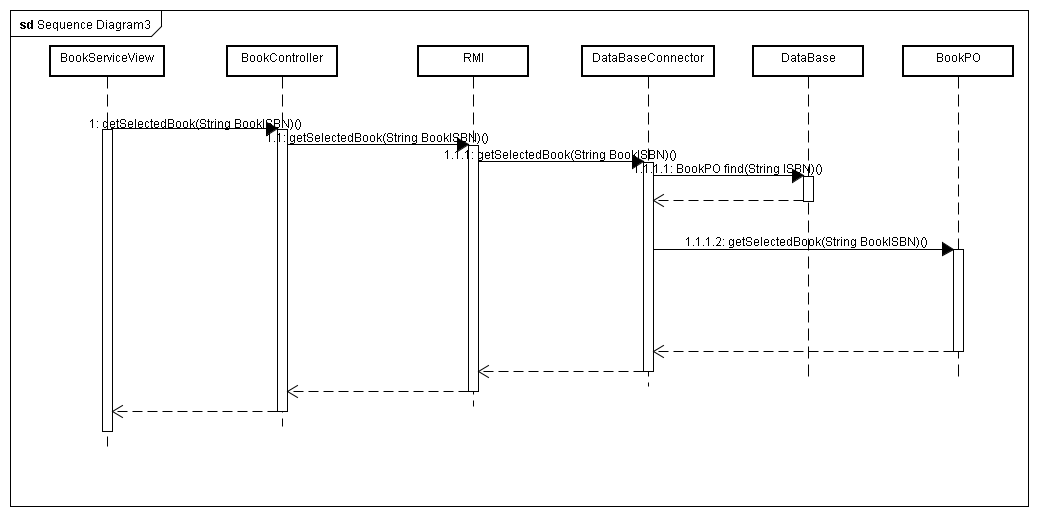
添加图书类别的顺序图

****

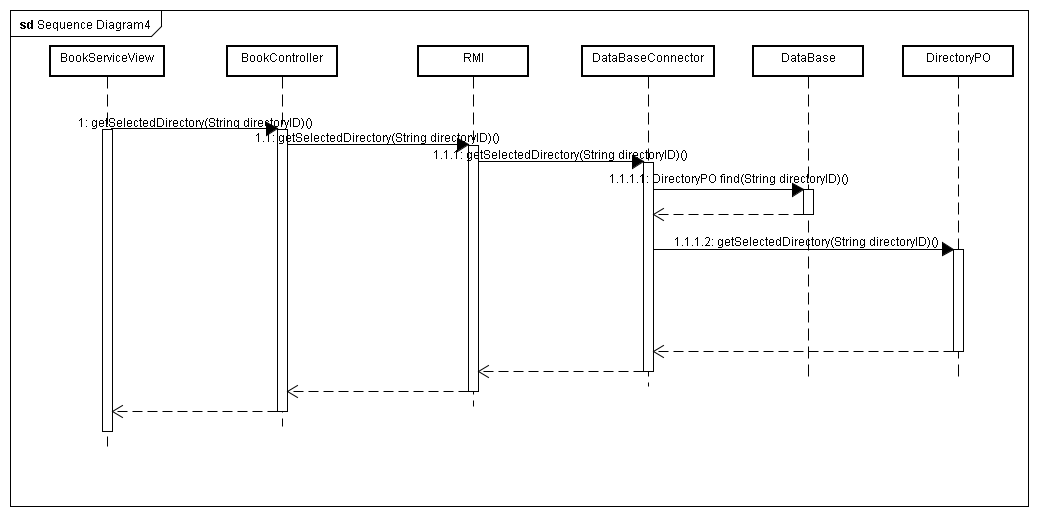
修改图书类别的顺序图

****

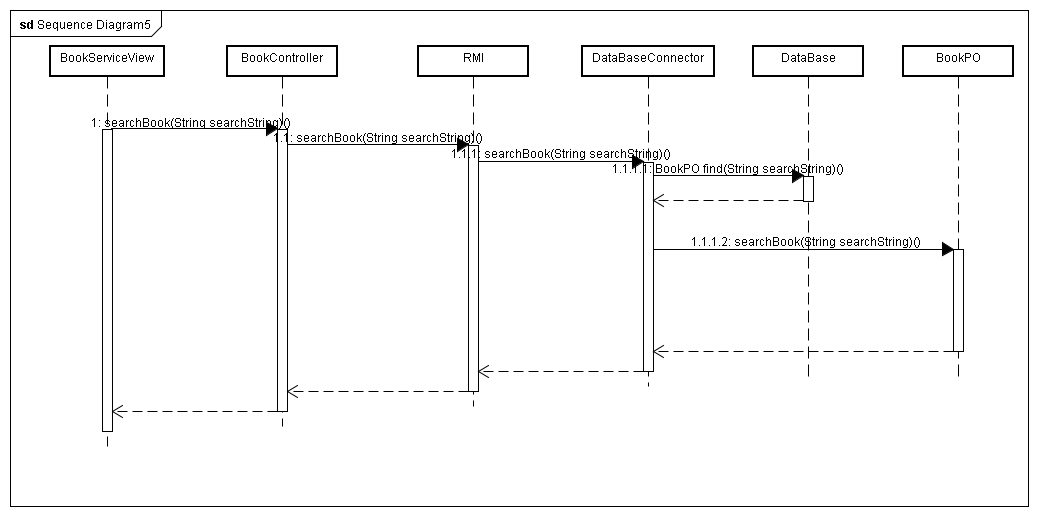
删除图书类别的顺序图

****

获取指定图书信息的顺序图

****

获取指定图书类别的顺序图

****

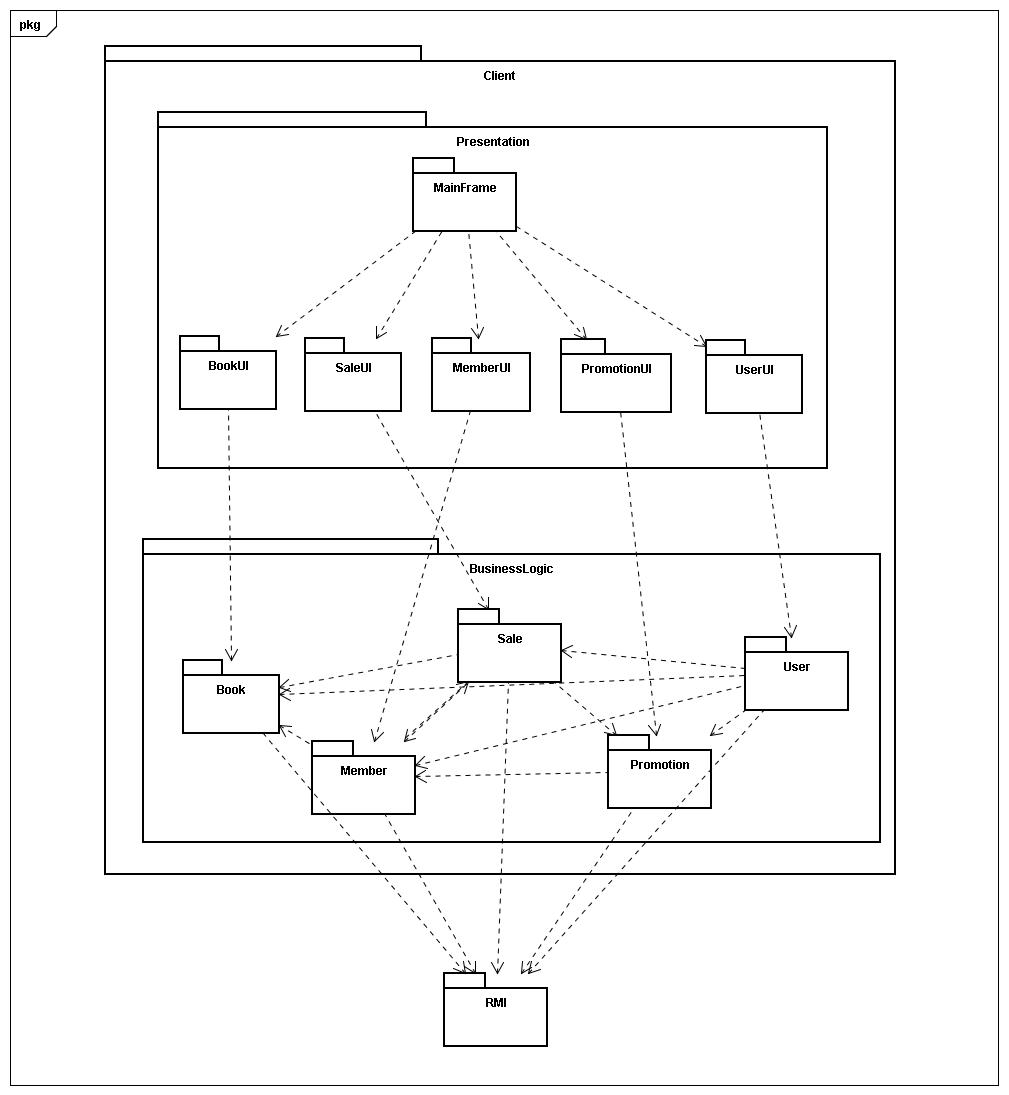
查找图书的顺序图

**五、业务逻辑层的设计原理**

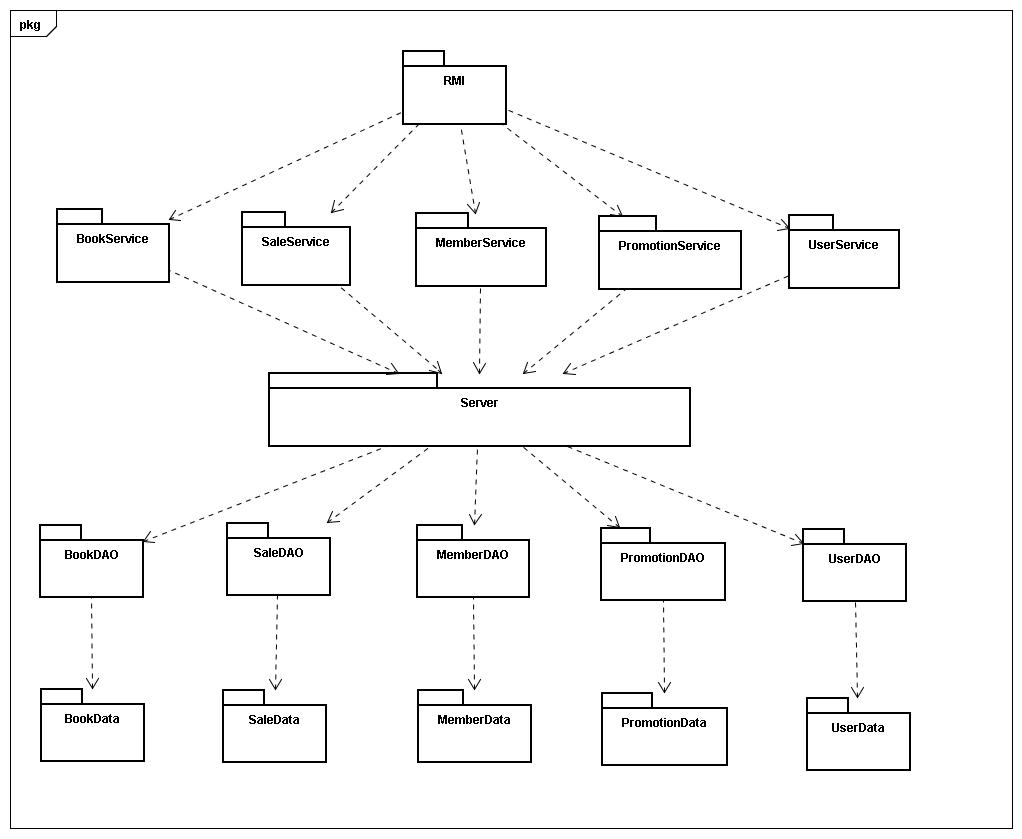
book模块的界面需要的业务逻辑通过BookService委托给book和bookclassify对象，通过将业务逻辑分解为对不同领域对象的操作及协作，并且委托给各自领域对象的持久化操作的数据操作接口DAO层，实现了委托式的控制。

**5.依赖视角**

以下是客户端和服务器端的包之间的依赖关系。



**客户端包图**



**服务器端包图**

**6.非业务功能需求说明**

服务器端应具备系统级功能。

系统级功能包括服务器参数的配置、用户日志的记录、自动的数据备份、密文保存密码。

客户端应具备稳定性。

在与服务器意外断开连接时，会出现提示“与服务器断开连接”的提示。并且自动尝试重连。