作业3：设计模型

# 系统架构

软件构架设计可以建立软件合理的层次结构，提供软件设计的质量。医院预约挂号系统进行的系统架构如图 1所示。



图 1 医院预约挂号系统系统架构图

# 用例实现

就系统核心业务用例（预约用例[未扩展打印、支付等用例]）给出该用例的交互图和VOPC类图，并描述其基本路径交互。

## 交互图

交互图用来描述系统中的对象是如何进行相互作用的。即一组对象是如何进行消息传递的。可以进行类的职责分配，完成用例实现。预约用例的交互图如图 2所示。

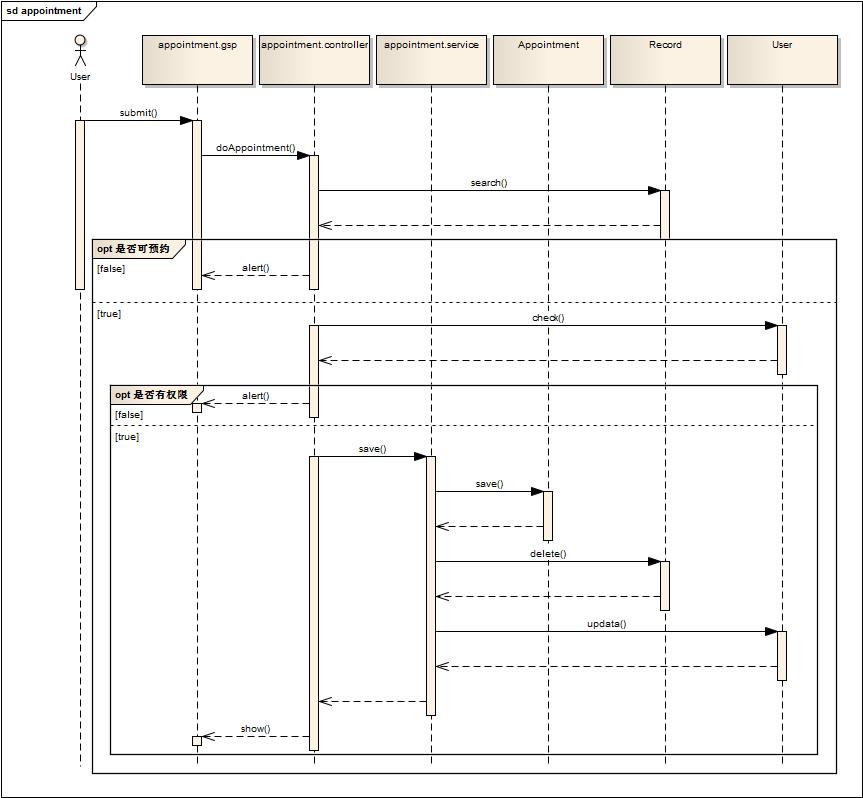


图 2 预约用例基本路径交互图

从图中可以看出，用户通过View层的gsp页面类提交表单到Controller层的控制类，控制类负责判断用户提交预约的出诊记录是否已被预约并校验用户的权限（信用等级及同时预约数量是否超额），不通过则提醒用户，若两者均通过则调用Service层的save方法保存记录。Service层具有事务管理，保存失败会进行回滚。保存成功（包括联级修改其他实体的相应字段）后向Controller层返回结果，Controller层将结果返回给View层。

## 类图

类图显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。预约用例的类图如图 3所示。

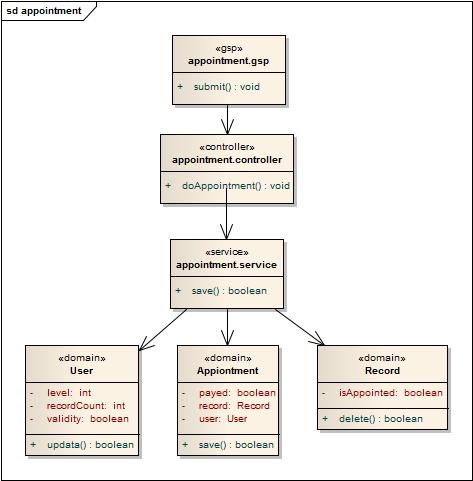


图 3 预约用例VOPC类图

# 支付接口的设计

有关预约挂号业务的费用支付问题，目前需要支持挂号处的现金交费和通过支付宝网上支付，为考虑支持其它可能的支付接口，方便日后系统扩展，我们采用GoF-策略模式设计该系统的费用支付接口，以使得系统能够适应多种方式。

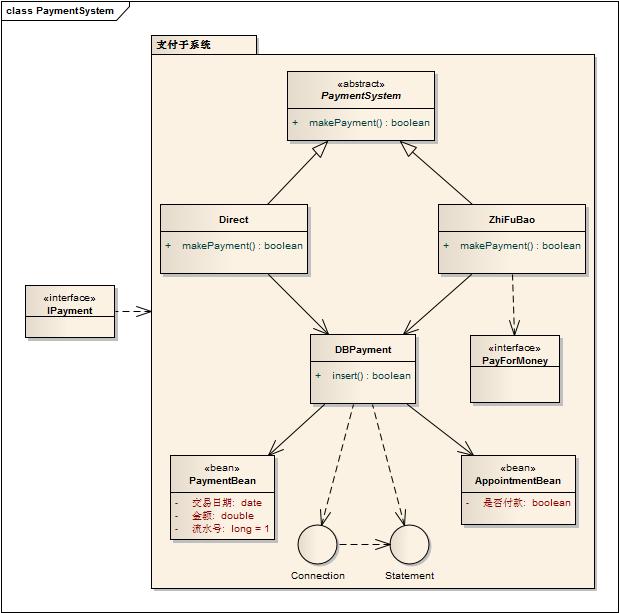


图 4 支付接口设计图示

如图 4所示，多种支付策略继承支付接口的实现基类，覆写基类的支付方法实现支付，无论何种支付方式均要有数据库操作，保证支付明细的产生于预约单实体相应标识字段的更新，同时采用“支付宝”的支付方式要额外调用第三方支付所提供的接口，此处不再展开。

# 数据库设计

数据库设计的目的是：

1. 为了确定设计中的持久性类；
2. 设计适当的数据库结构已存储持久化类；
3. 为存储和取得持久性数据定义机制和策略，以满足系统的性能标准。

因此，针对医院预约挂号系统的数据库设计如图 5所示。

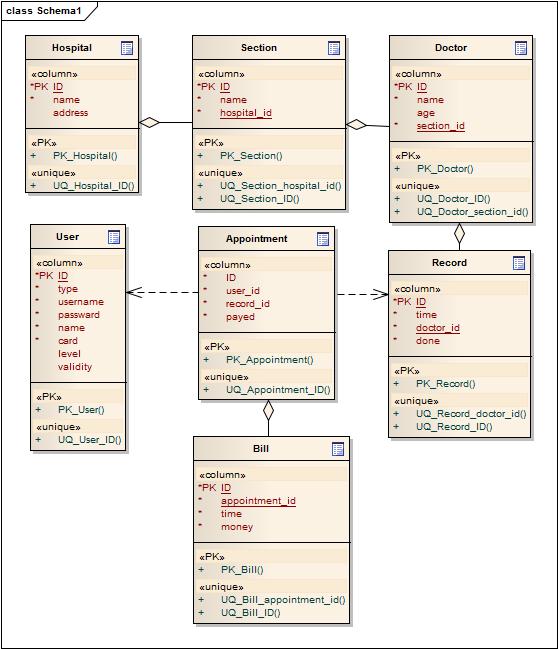


图 5 医院预约挂号系统数据库设计