# 1.Advertising模块设计

## 1.1概述

### 1.1.1承担的需求（主要秘密）

本模块旨在根据不同合作商户的要求，动态地在网页添加广告

涉及需求：

R3 系统安全性 屏蔽99%以上的插件扫描攻击和特定的IP

R4入侵检测可拓展性 增加新的检测规则所需时间低于0.25人月

R11 浏览器兼容性 兼容Firefox版本号>=3.5;兼容Chrome版本号>=3;兼容IE版本号>=6;兼容Opera版本号>=10.5;兼容Safari版本号>=3.1

R31 植入广告 广告展示的形式多样化

R34 广告可扩展性 广告的位置和数量随时可以发生变化

### 1.1.2可能会修改的实现（次要秘密）

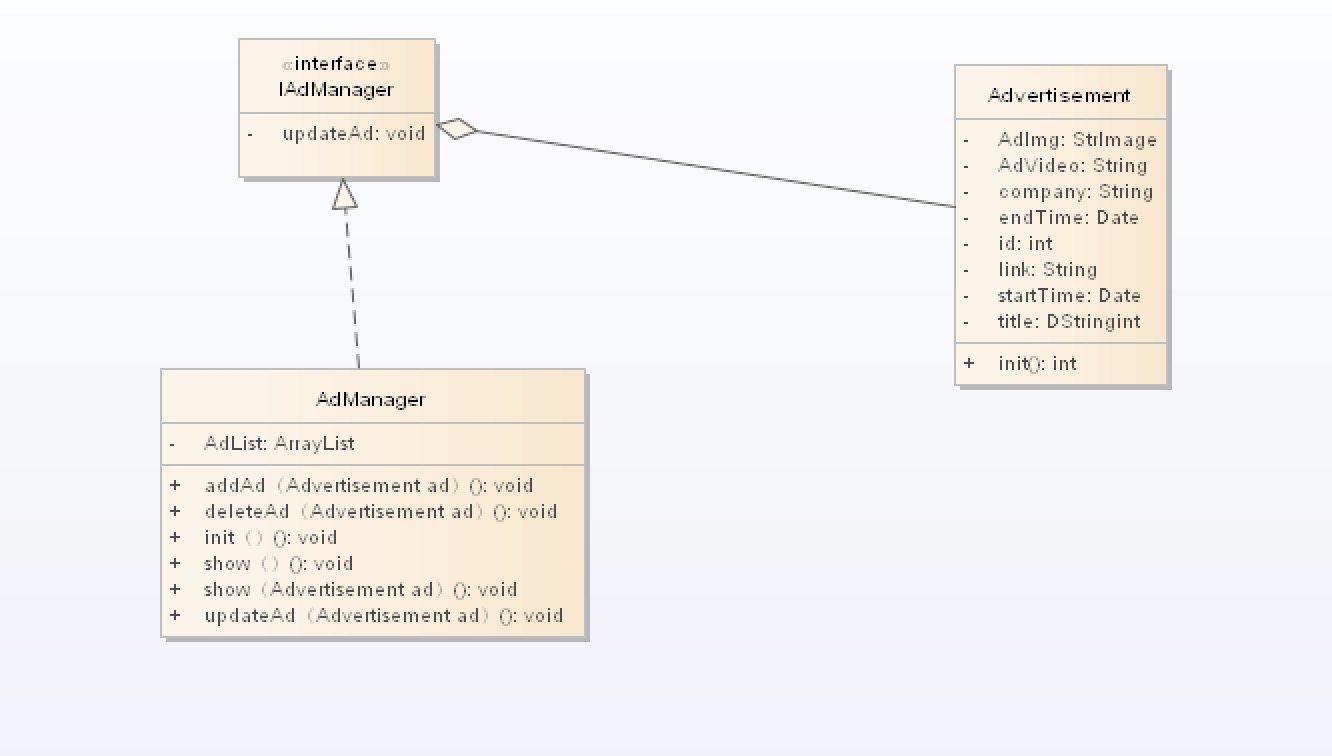
**广告的类型**

### 1.1.3角色

### 1.1.4对外接口

|  |  |
| --- | --- |
| **接口名** | **职责** |
| Public void updateAdvertisementInfo(Advertisement info) | 更新广告内容信息 |
| Public void showAdvertisement (Advertisement info) | 在当前页面显示广告信息 |

## 1.2类图



## 1.3类描述

1.3.1 IAdManager接口

该接口封装了广告管理的方法，根据各商家的出价进行竞价排序，更新网页上的广告信息

1.3.2 AdManager类

该类的职责为对广告的增加，删除，更新，显示进行管理

|  |  |
| --- | --- |
| 方法名 | Public void addAd(Advertisement ad) |
| 前置条件 | ad不为null,ad不存在于数据库中，管理员已经登陆 |
| 后置条件 | 无 |
| 作用 | 在数据库中新增一个广告 |

|  |  |
| --- | --- |
| 方法名 | Public void deleteAd(Advertisement ad) |
| 前置条件 | ad不为null，管理员已经登陆 |
| 后置条件 | 数据库中没有该广告 |
| 作用 | 在数据库中删除一个广告 |

|  |  |
| --- | --- |
| 方法名 | Public void updateAd(Advertisement ad) |
| 前置条件 | ad不为null,数据库中能找到对应id的ad，管理员已经登陆 |
| 后置条件 | 在数据库中更新该广告信息 |
| 作用 | 在数据库中更新一个广告 |

|  |  |
| --- | --- |
| 方法名 | Public void show(Advertisement ad) |
| 前置条件 | ad不为null,ad存在于数据库中，管理员已经登陆 |
| 后置条件 | 无 |
| 作用 | 在页面上显示一个广告 |

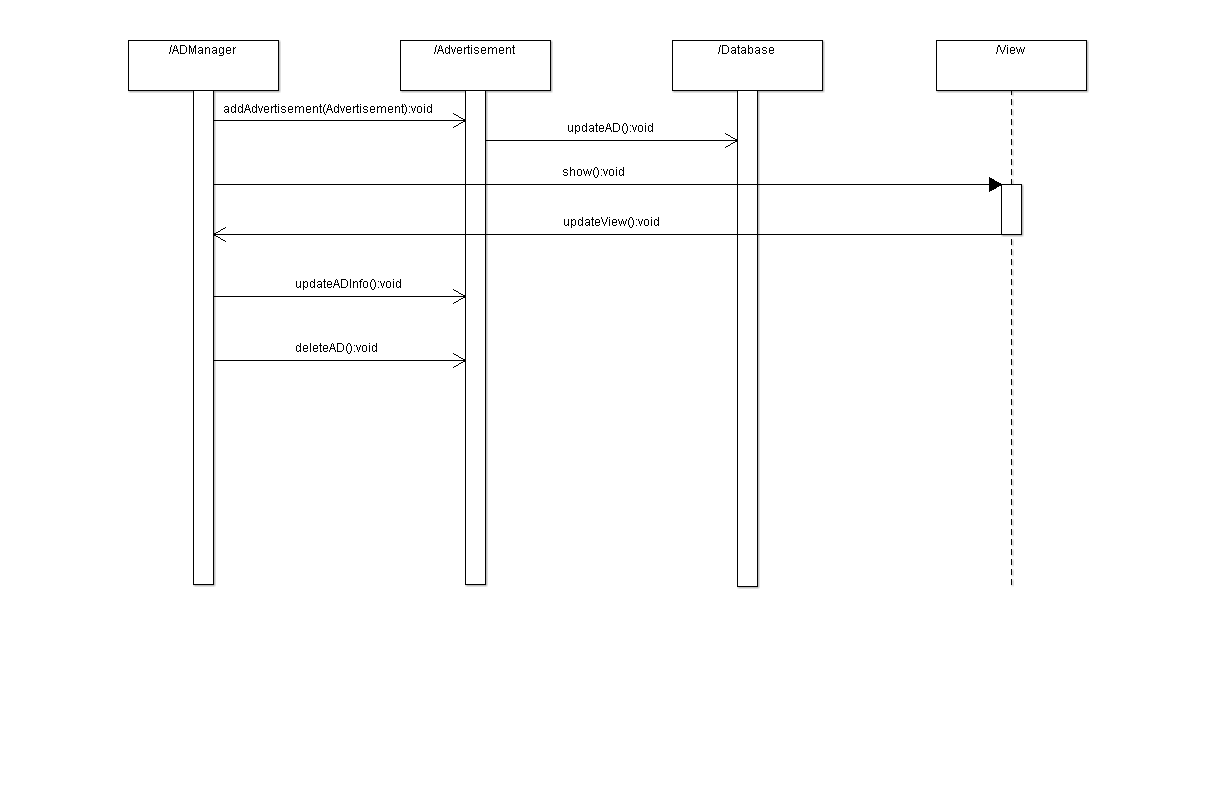
|  |  |
| --- | --- |
| 方法名 | Public void show() |
| 前置条件 | 数据库中广告不为空，管理员已经登陆 |
| 后置条件 | 无 |
| 作用 | 显示所有广告 |

|  |  |
| --- | --- |
| 方法名 | Public void init() |
| 前置条件 | 管理员已经登陆 |
| 后置条件 | 数据库中广告都进行初始化 |
| 作用 | 在数据库中初始化广告 |

数据结构描述：

|  |  |
| --- | --- |
| 数据结构 | 描述 |
| ArrayList AdList | 广告集合 |

## 1.4重要协作



协作描述：

广告管理模块通过实现IAdManager接口的AdManager类来进行广告的增加，删除，更新，初始化到数据库，然后通过界面显示出来

## 1.5使用的设计模式

## 1.5.1 MVC模式

该模块使用MVC模式构建，将网页，逻辑算法和数据库中的数据分离开

# 2.BuyGoods模块设计

## 2.1概述

### 2.1.1承担的需求（主要秘密）

该模块旨在满足用户的购买商品需求，用户在进行搜索得到合适的商品后，可以将商品加入购物车，并进行网银付款。

涉及需求

R30 购买商品 提供指向购物网站商家的链接

### 2.1.2可能会修改的实现（次要秘密）

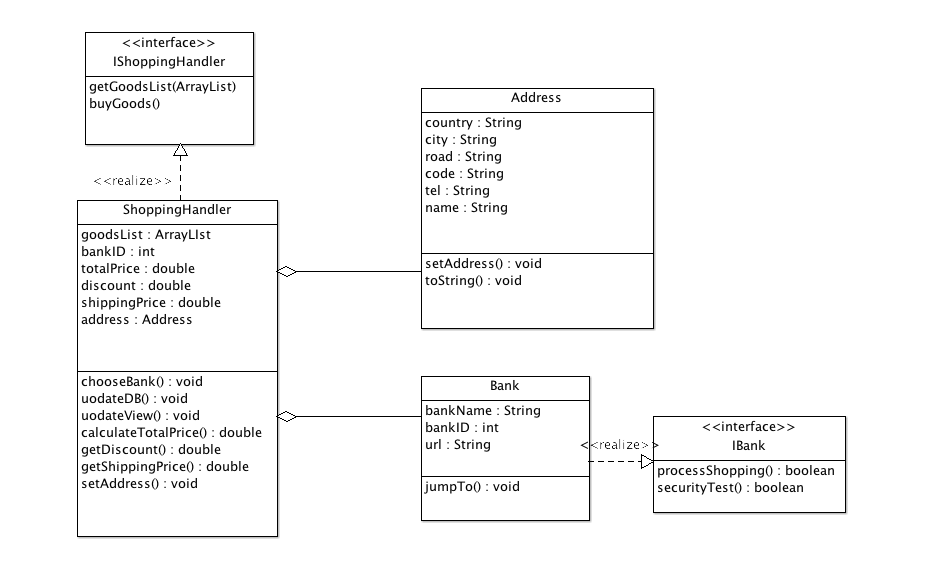
用户进行网银付款时候系统所支持的银行类别

### 2.1.3角色

### 2.1.4对外接口

|  |  |
| --- | --- |
| **接口名** | **职责** |
| Public void updateShoppingInfo (Shopping info) | 在数据库中更新支付内容信息 |
| Public void buyGoods(Shopping info) | 购买商品 |

## 2.2类图



## 2.3类描述

### 2.3.1 IshoppingHandler接口

该接口实现购买商品，使用策略模式将ShoppingHandler的方法抽象出来，便于以后进行改进

### 2.3.2 ShoppingHandler类

该类的职责是获得用户所选取的所有商品，计算总价、运费、要求用户填写收获地址以及联系方式，待用户确定购买之后，提示用户选择相应网上银行，用户付款之后进行数据库的更新，并更新网页信息，告知用户购买结果。

类方法：

|  |  |
| --- | --- |
| 方法名 | Public void getGoodsList(ArrayList list) |
| 前置条件 | 用户已经登陆，list不为nulll |
| 后置条件 | 显示用户选择的所有商品 |
| 作用 | 从用户输入模块获得用户所选择的所有商品 |

|  |  |
| --- | --- |
| 方法名 | Public void buyGoods() |
| 前置条件 | 用户已经登陆，用户点击提交按钮 |
| 后置条件 | 跳转到相应的购买页面 |
| 作用 | 用户点击提交按钮以后进行商品的购买 |

|  |  |
| --- | --- |
| 方法名 | Public void chooseBank(Bank bank) |
| 前置条件 | 用户已经登陆，选择银行种类 |
| 后置条件 | 跳转到下一信息填写页面 |
| 作用 | 供用户选择所要使用的银行种类 |

|  |  |
| --- | --- |
| 方法名 | Public void updateDB() |
| 前置条件 | 用户已经登陆，用户成功购买商品 |
| 后置条件 | 数据库信息进行更新 |
| 作用 | 用户购买成功以后，进行数据库的更新操作 |

|  |  |
| --- | --- |
| 方法名 | Public void updateView() |
| 前置条件 | 用户已经登陆，用户成功购买商品 |
| 后置条件 | 跳转到购买成功后页面 |
| 作用 | 用户购买成功之后，进行网页的更新 |

|  |  |
| --- | --- |
| 方法名 | Public void calculateTotalPrice() |
| 前置条件 | 用户已经登陆，用户选定需要购买的商品 |
| 后置条件 | 显示选定商品的总价格 |
| 作用 | 计算所有商品的总额 |

|  |  |
| --- | --- |
| 方法名 | Public ArrayList<String> getDiscount() |
| 前置条件 | 用户已经登陆 |
| 后置条件 | 显示所有打折活动信息 |
| 作用 | 检查是否有打折活动 |

|  |  |
| --- | --- |
| 方法名 | Public void getShippingPrice() |
| 前置条件 | 用户已经登陆,用户选定地址 |
| 后置条件 | 显示数据库中该地址对应的运费 |
| 作用 | 查询数据库，根据用户的所在地获得运费 |

|  |  |
| --- | --- |
| 方法名 | Public void setAddress(String address) |
| 前置条件 | 用户已经登陆，用户填写好地址 |
| 后置条件 | 在数据库中更新好地址信息 |
| 作用 | 存储用户填写的地址 |

|  |  |
| --- | --- |
| 方法名 | Public void chooseAddress() |
| 前置条件 | 用户已经登陆 |
| 后置条件 | 显示数据库中存储的用户常用地址 |
| 作用 | 根据用户以前填写的地址状况，供用户选择常用地址 |

### 2.3.3 Address类

该类的职责是存储用户的地址，以便用户在购买时填写

类方法

|  |  |
| --- | --- |
| 方法名 | Public void setAddress(Addressaddress) |
| 前置条件 | 用户已经登陆，已经填写好地址信息 |
| 后置条件 | 在数据库中更新地址信息 |
| 作用 | 设置用户的地址 |

|  |  |
| --- | --- |
| 方法名 | Public String toString(Addressaddress) |
| 前置条件 | Address不为null |
| 后置条件 | 返回String格式的地址 |
| 作用 | 将地址变成String的格式，提供给外部模块 |

### 2.3.4 IBank接口

改接口将不同银行共有的方法抽象出来，方便以后添加新的功能

### 2.3.5 Bank类

该类实现了IBank接口，职责是在用户选择银行的时候，查询数据库，将用户引导到正确的网银付款窗口

类方法

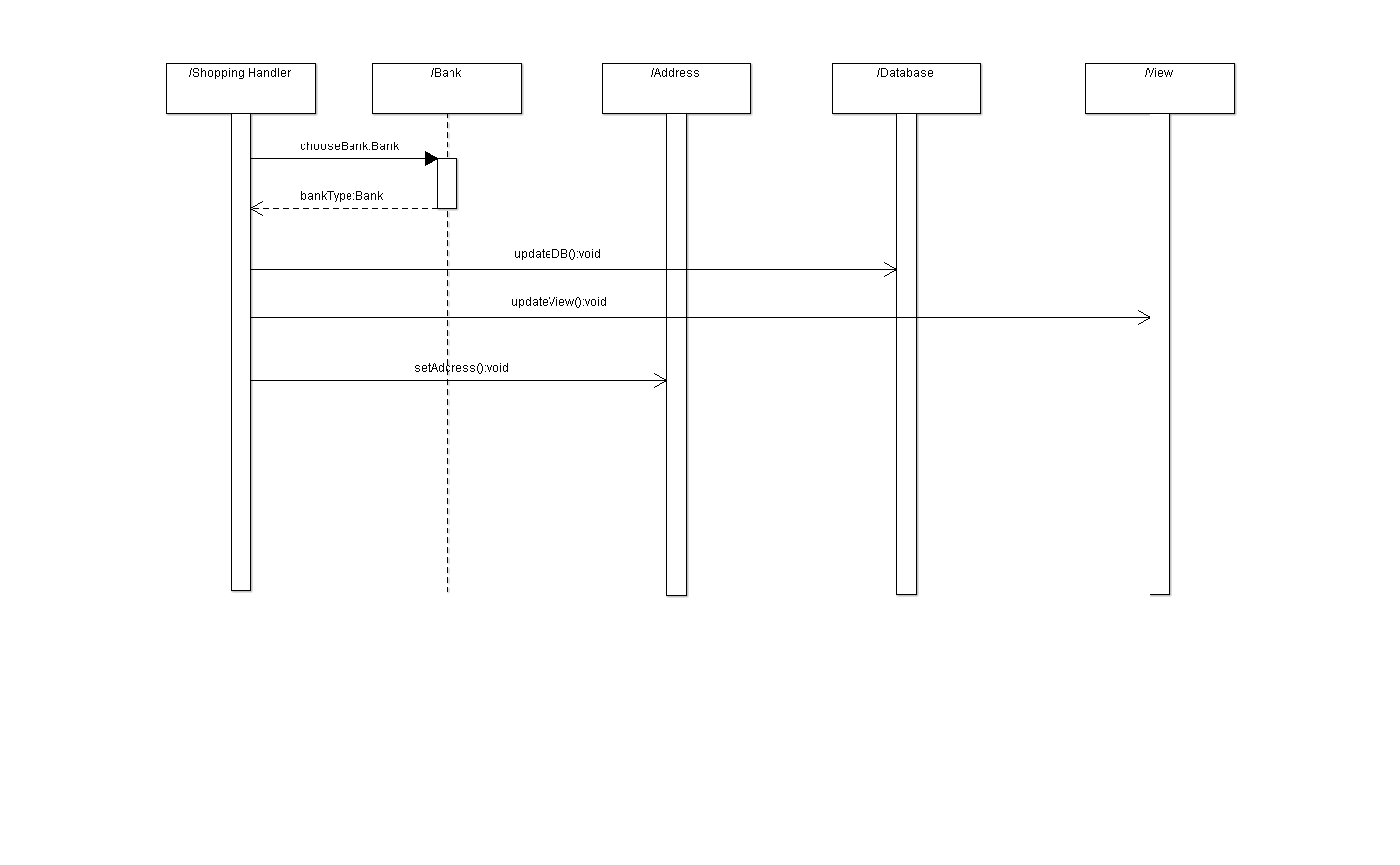
|  |  |
| --- | --- |
| 方法名 | Public void jumpTo(String link) |
| 前置条件 | 用户已经登陆，用户已经选择好支付银行 |
| 后置条件 | 网页跳转到正确的付款窗口 |
| 作用 | 根据用户选择的银行，查询数据库，将网页跳转到正确的付款窗口 |

|  |  |
| --- | --- |
| 方法名 | Public Boolean processShopping(Shopping info) |
| 前置条件 | 用户已经登陆，用户已经选定商品并选择好支付银行 |
| 后置条件 | 显示执行结果是否成功 |
| 作用 | 执行付款操作，根据银行的不同种类，该方法有不同的实现方式。该方法返回执行结果的布尔值 |

|  |  |
| --- | --- |
| 方法名 | Public Boolean securityTest() |
| 前置条件 | 用户已经登陆，用户已经填写好支付信息 |
| 后置条件 | 显示检测的结果是否满足付款要求 |
| 作用 | 检测当前网页安全环境是否满足付款要求，返回检测结果的布尔值。根据不同的银行种类，该方法有不同的实现。 |

数据结构描述

## 2.4重要协作



协作描述

商品购买模块从选定商品进行购买后，设置支付的银行，地址等信息，然后将信息更新到数据库中，并且在界面上显示出来。

## 2.5使用的设计模式

### 2.5.1策略模式

将银行进行付款的算法封装出来，便于以后修改。保证了高内聚和低耦合

### 2.5.2 MVC模式

该模块使用MVC模式构建，将网页、逻辑算法和数据库的数据分开