

**易物——二手交换平台**

**软件设计说明书**



北京航空航天大学

2016-11

|  |  |
| --- | --- |
| **项目组成员信息** | |
| **姓名** | **本文档中主要承担的工作内容** |
| 何玥 | 文档范围描述、体系结构设计、数据结构设计和详细设计，  贡献比60% |
| 周啸辰 | 需求概述，贡献比10% |
| 张诚 | 内部接口设计，贡献比10% |
| 孟祥鑫 | 用户界面设计，贡献比10% |
| 柴舜 | 外部接口设计、引用文档汇总，贡献比10% |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V1.0 | 2016/11/17 | 何玥 | 周啸辰 | 初稿，会议讨论和总结 |
| V2.0 | 2016/11/23 | 何玥 | 张诚 | 添加接口设计 |
| V3.0 | 2016/11/29 | 何玥 | 周啸辰 | 终稿阅读和细节修正 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[1. 范围 1](#_Toc307923001)

[1.1 标识 1](#_Toc307923002)

[1.2 系统概述 1](#_Toc307923003)

[1.3 文档概述 1](#_Toc307923004)

[1.4 术语和缩略词 1](#_Toc307923005)

[2. 引用文档 1](#_Toc307923006)

[3. 需求概述 2](#_Toc307923007)

[4. 体系结构设计 2](#_Toc307923008)

[4.1 总体结构 2](#_Toc307923009)

[4.2 功能分配 2](#_Toc307923010)

[4.3 关键问题及解决方案 2](#_Toc307923011)

[5. 接口设计 3](#_Toc307923012)

[5.1 用户界面设计 3](#_Toc307923013)

[5.2 外部接口设计 3](#_Toc307923014)

[5.3 内部接口设计 3](#_Toc307923015)

[6. 数据结构设计 3](#_Toc307923016)

[6.1 公共数据结构设计 3](#_Toc307923017)

[6.2 数据库设计 3](#_Toc307923018)

[7. 详细设计 4](#_Toc307923019)

# 范围

## 标识

文档标识号：A2016-11-20-00-02

文档标题：系统设计说明

版本号：V3.0

项目/产品中文全称：易物校园二手物品交换平台

项目/产品编码：201611200002

## 系统概述

“易物”是以B/S方式架构的在线应用系统，是为校园用户提供二手物品交换的平台，用户可在平台上发布交易信息，与其他用户进行相关交流。

项目于2016年9月29日提出，至11月25日前端网页已搭建完成，目前正处于数据库优化与测试阶段。

投资方：北京易物科技有限公司

需方：北京易物科技有限公司

用户：高校在校大学生

开发方：北京航空航天大学计算机学院软件工程14级实验小组4组

支持机构：北京航空航天大学计算机学院

当前运行现场：数据库内部测试

计划运行现场：数据库与网页对接

## 文档概述

本文档为系统设计说明书，描述了系统的各方面概述，体系结构，相关接口，数据结构以及系统内部模块设计。

本文档用途是作为开发人员在设计上的参考，因此其中的内容与最终成品密切相关。考虑到软件开发过程中的保密需求，故其中内容仅限开发人员浏览。应当采用加密或身份验证的方式来限制相关人员对本文档的访问权。

## 术语和缩略词

HTML：HyperText Markup Language 超级文本标记语言。

JS：JavaScript 一种运行在浏览器上的脚本语言。

CSS：Cascading Style Sheets 用来表现HTML文件样式的计算机语言。

PHP：Hypertext Preprocessor 超文本预处理器，是一种通用开源脚本语言。

AJAX：Asynchronous JavaScript And XML 创建交互式网页应用的开发技术。

# 引用文档

1. GB-T 8567-2006 计算机软件文档编制规范，国标
2. A2016-09-29-12-30 软件开发计划书
3. A2016-11-02-00-01 系统需求规格说明书
4. A2010-00-02-00.SDD-系统设计说明模板，内部文档

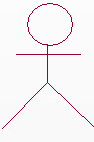
# 需求概述

**概述系统特性和需求：**

易物二手交易平台系统是给用户提供二手商品发布，出售，购买的平台。系统特征是信息吞吐量较大，信息管理需要做到及时同步。系统需求分为三个部分，用户信息管理需求，商品管理需求和交易平台管理需求。

对需求规格说明的更变：

1. 取消系统中“bbs模式”的开发约束，修改为主页分类推荐和分类搜索商品开发模式。主页部分根据用户喜好，以图片方式对商品进行推荐，分类部分按照商品分类进行商品分类，通过商品分类列表定位商品信息。在网页的logo下方新增一行搜索栏，允许用户对商品进行直接搜索。



用户

图1搜索商品用例

用例名：发布商品

用例描述：允许用户搜索商品

参与者：用户

前置条件：用户输入搜索的信息并点击搜索按钮

后置条件：页面展示根据用户搜索条件筛选出来的商品

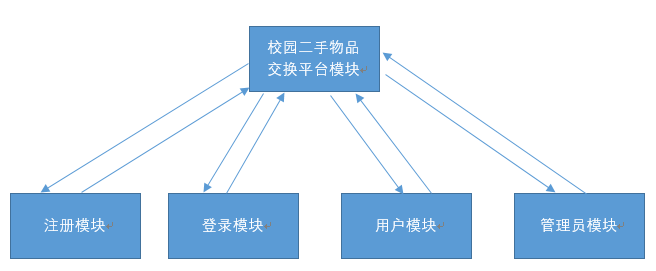
主事件流：

1. 用户输入搜索信息并点击搜索商品按钮,如果用户没有在搜索框输入信息，跳出弹框进行提示，不进行跳转。
2. 修改用户定义中“数据库”，修改为“数据库管理员”，将所有模型中“数据库”角色修改为“数据库管理员”角色。

# 体系结构设计

## 总体结构

1. **软件体系结构**

****

**SC图(1)**

**模块说明：**

校园二手物品交换平台模块：总模块，是用户与系统交互的程序段。

注册模块：完成用户的注册。

总模块向注册模块发送用户的注册信息，注册模块返回给总模块注册的状态。

登录模块：识别用户的登录。

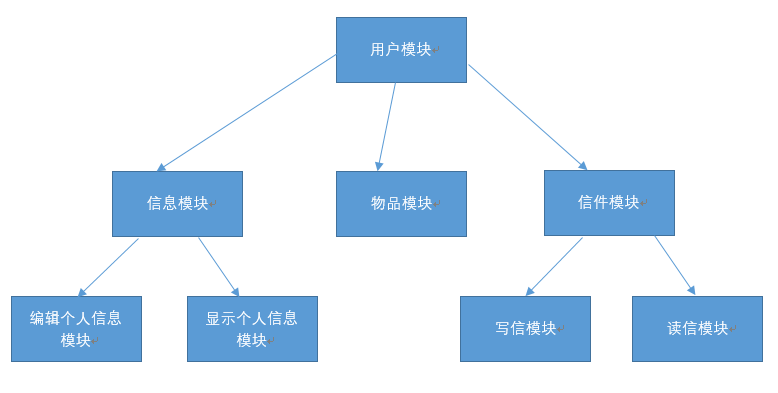
总模块向登录模块发送用户输入的用户名和密码，登录模块返回给总模块登录的结果。

用户模块：用户在平台的行为的集合。

总模块向用户模块发送用户名，用户模块向总模块发送注销信息。

管理员模块：管理员在平台的行为的集合。

总模块向管理员模块发送管理员编号，管理员模块向总模块发送注销信息。

****

**SC图(2)**

**模块说明：**

信息模块：编辑和显示用户的个人信息。

用户模块向信息模块发送用户名。

编辑个人信息模块：编辑用户的个人信息。

信息模块向编辑个人信息模块发送用户名和用户修改的信息。

显示个人信息模块：：显示用户的个人信息。

信息模块向显示个人信息模块发送用户名。

物品模块：用户在平台上对物品的行为的集合。

用户模块向信息模块发送用户名。

信件模块：满足用户和用户之间的通信、交流功能。

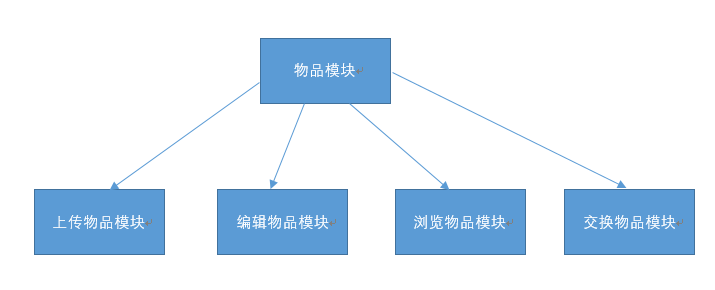
用户模块向信件模块发送用户名。

写信模块：满足用户向某一指定用户发送一封信件。

信件模块向写信模块发送收、发信件的用户名和信件内容。

读信模块：支持用户阅读历史信件。

信件模块向读信模块发送用户名。

****

**SC图(3)**

**模块说明：**

上传物品模块：用户上传一件新的物品，并等待管理员审核。

物品模块向上传物品模块发送新物品的介绍和参数信息。

编辑物品模块：用户可修改已上传的物品的具体信息。

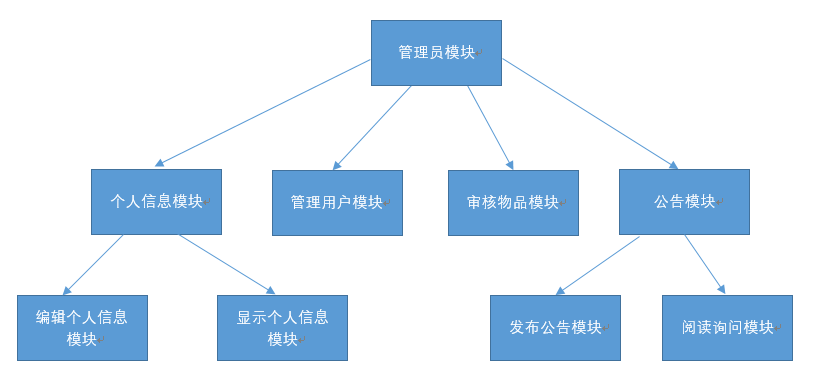
物品模块向编辑物品模块发送物品的修改信息。

浏览物品模块：支持用户根据输入的关键字，搜索符合条件的物品。

物品模块向浏览物品模块发送用户输入的关键字。

交换物品模块：用户可以向某件物品提出交换申请。

浏览物品模块向交换物品模块发送交换双方物品的编号。

****

**SC图(4)**

**模块说明：**

个人信息模块：编辑和显示管理员的个人信息。

管理员模块向个人信息模块发送管理员编号。

编辑个人信息模块：编辑管理员的个人信息。

个人信息模块向编辑个人信息模块发送管理员编号和管理员修改的信息。

显示个人信息模块：显示管理员的信息。

个人信息模块向显示个人信息模块发送管理员编号。

管理用户模块：管理员可以查看用户的个人信息和信用度，以管理用户。

管理员模块向管理用户模块发送用户名。

审核物品模块：管理员对用户上传的物品需要审核，并给出审核意见，只有通过审核的物品会出现在市场中。

管理员模块管理员审核物品模块发送审核物品编号的审核意见。

公告模块：管理员向用户发送通知并回答用户的询问。

管理员模块向公共模块发送用户名集合。

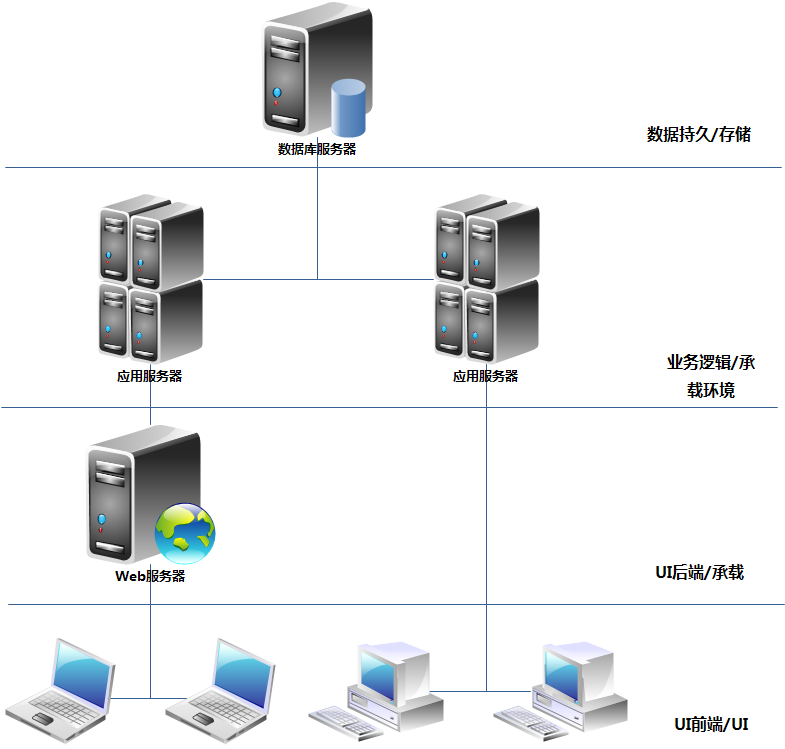
发布公告模块：管理员向指定的用户集合发布公告。

公告模块向发布公告模块发送用户名集合和公告内容。

阅读询问模块：管理员可以阅读用户发布的询问。

公告模块向阅读询问模块发送用户名。

1. **硬件体系结构**

****

系统采用B/S架构，我们将应用服务器和数据库服务器放在一台机器上。

(1)应用服务器

用于实现和运行本系统所需的业务逻辑和应用软件模块,以保证数据访问的高可用性和可靠性。

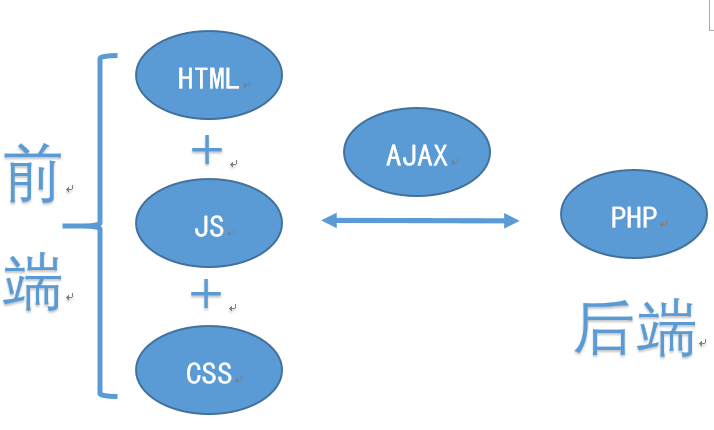
(2)数据库服务器

存储本系统需要的所有数据,当此服务器失败后，系统会自动切换到应用服务器，启动相应的数据库服务器软件，继续提供服务。

(3)WEB服务器

面向用户，提供各种WEB页面。

1. **技术体系结构**

****

整体设计采用了前端HTML+JS+CSS加后端PHP的模式，之间通过AJAX技术进行交互。前端设计采用了热门的bootstrap3样式和jQuery框架。风格以简洁为主，界面简约而不简单，使用了大量的动画操作，用户可以很快的上手操作，不会有明显的交互困难。

1. **支撑体系结构**

上文已经说过，应用服务器和数据库服务器在一台机器上。使用一台计算和存储性能高的处理器，计算性能：8G显存、64G内存、双CPU；存储性能：2T内存以上。安装MYSQL数据库和服务器，做后端支撑。租用WEB服务器执行和处理网页服务。

## 功能分配

总体结构中，将需求文档中的各个功能转换用户、管理员和平台三者联系，总体结构就是对各个功能的程序化描述，功能的具体实现是各个模块的基本功能和相互的调用关系。

编辑个人信息模块实现了用户和管理员维护个人信息的功能；

上传物品模块实现了用户发布个人物品的功能；

编辑物品模块实现了用户维护物品信息的功能；

浏览物品模块实现了用户搜索物品的功能；

交易模块实现了用户交易物品和相互评价的功能；

信件模块实现了用户之间交流，用户向管理员提问的功能；

管理用户模块实现对管理和约束用户的功能，保证了平台秩序；

审核物品模块保证了平台物品的合法性；

公告模块实现了管理员发布公告和回答用户提问的功能。

## 关键问题及解决方案

作为一个提供物品搜索的平台，保证数据库存储信息的安全性、正确性是十分重要的，这里我们使用MYSQL数据库的特性解决这一问题，使用触发器保证数据的一致性更新，使用事物处理保证操作的完整性，再使用授权机制保护数据库的安全。

第二个问题是如何更好地用户提供搜索，我们建立了日志机构，记录下用户在平台上操作的每一步操作，尤其是浏览，为用户推荐物品时，根据用户的浏览和交易记录能够更好为用户提供需要的物品的信息，对物品添加了如类别、标签等多种修饰的方法，并由用户自己为物品添加标签，以更广泛的定义和推荐物品。

第三个问题是保持大数据量存储和访问的稳定问题。方案：服务器每个月月初凌晨1点到4点服务器停机，进行服务器维护，检测服务器运行情况并对服务器运行参数进行备份。每周日凌晨1点到4点进行数据库检测及数据备份。

第四个是保持并发的修改和访问操作的正确性问题。我们把对数据库的全部操作进行类型和性质的划分，把这些划分都定义为事务。事务是数据库中一个基本操作单位，事务中包含一段连续的操作，这些操作要不全做，要不不做。对事务中的任一条操作失败，立即回滚，回到未操作状态。对数据库的操作转化为对调用事务。

# 接口设计

## 用户界面设计

1. **系统主页**

一个好的主页总是能够帮助开发者吸引更多人的眼球，因此我们在主页的设计上下了一番功夫。我们从“循环利用”、“双倍享受”、“扩展社交”这三个角度来阐释系统区别于其他购物网站的特征，并通过一个“交换礼物”的小故事来激发用户对我们网站的探索欲望。

1. **注册功能**

我们提供了新用户的注册功能。与一般网站注册有所不同的是，由于我们侧重的是校园内学生之间的交易，为了提供最有效也是最安全的保障，我们强制要求新用户必须以学号为基本项进行实名化注册，并留下正确可用的手机号以及邮箱地址，工作人员会在信息核查后公布这个用户是否注册成功。

1. **登录功能**

点击主页的右上角“登录”按钮，可以弹出登录的窗口，我们在这里没有将点击事件转换为一次页面的切换，是考虑到登录功能经常使用，且需要的内容极简，因此将它做成弹出窗口的形式，能够提升用户的使用舒适度。

1. **修改信息功能**

系统为每个注册过的用户建立了个人主页，用户可以在自己的主页中查看和修改自己的用户信息。其中用户能够自己修改的信息只有两项，是“密码”和“头像”，其他信息例如“用户名”、“学号”、“性别”、“邮箱”、“电话”、“信誉度”和“用户类型”是不能自己更改的信息，其中前五项是自己注册时填写的信息，出于对每个用户、每单交易的负责，我们不允许用户自己修改这些信息，以免出现利用信息的修改来实现交易欺骗的不良事件。而后两项是系统根据用户的交易历史以及在线时长为用户做出的评估，同时也是为交易的双方提供一个客观的信用参考。如果用户想要修改自己无法修改的信息，可以发消息向管理员提出申请。

1. **卖方功能**

在我们的系统中，实质上没有买方和卖方的区分，毕竟这是一个物品与物品的交易平台，而不是实际意义上的购物平台。因此，每个注册用户的角色都是一样的，既可以将自己的物品简介上传到自己的主页，又可以在电子商场中搜索别人的物品。但是对于某一次的交易，必定会有偏向卖方的一方，用户的物品被其他用户询问是达成本次交易的开端。因此，我们作为卖方角色的用户提供了物品信息的上传以及下架功能。

1. **买方功能**

同卖方功能的叙述，我们创建了一个“市场”，它是一个依托于数据库系统的前端搜索设计，用户可以在这里搜索关键字，来检索自己喜爱的物品名称或者其他信息，这样能够扩大系统用户之间的有效连接，能够使得用户之间尽快地达成更多的交易。买方可以主动向卖方发送消息，来进行对于该商品交易的交谈活动。

1. **管理员功能**

任何一个系统中，管理员的角色都是不可或缺的。我们的系统为管理员提供了一个专用的账号，用来区别于普通用户。管理员具有以下职能：审核用户的商品信息上传申请，处理用户的修改信息申请，获取并核实用户的投诉申请，发布平台公告，管理用户名单等。

## 外部接口设计

应用服务器和数据库服务器由一台计算和存储性能高的计算机支持，配置WINDOWS操作系统。使用WampServer集成安装环境搭建数据库(MySQL)、后端程序(PHP)的连接，并把对数据库的增、删、改、查操作封装在一个个小的PHP子程序中。前端web程序安装在外部租用的WEB服务器上，用户使用浏览器如IE、火狐等即可使用系统。

用户通过输入框输入查询或文本信息，通过按钮发出命令或选择功能。

外部接口分类说明：

1. 文本框：输入信息。
2. 点击“注册用户”：进入注册界面。
3. 点击“登录”：进入登录界面。
4. 点击“确认”：提交输入信息。
5. 点击“查看个人信息”：显示个人信息。
6. 点击“编辑”：编辑信息。
7. 点击“上传图片”：选择本地图片。
8. 点击“添加物品”：添加一件物品。
9. 点击“删除物品”：删除物品。
10. 点击“物品表”：显示用户个人物品。
11. 点击“物品名”：查看物品详情。
12. 点击“申请交易”：填写交易方案。
13. 点击“取消交易”：取消交易。
14. 点击“确认交易”：填写评价。
15. 点击“发送信件”：发送信件或公告。
16. 选择“审核意见”：选择一个审核意见。
17. 点击“搜索”：搜索物品。
18. 选择“有序搜索”：选择如“价格最低”等选项。

## 内部接口设计

校园二手物品交换平台模块向注册模块发送用户的注册信息，注册模块返回给总模块注册的状态；向登录模块发送用户输入的用户名和密码，登录模块返回给总模块登录的结果；向用户模块发送用户名，用户模块向总模块发送注销信息；向管理员模块发送管理员编号，管理员模块向总模块发送注销信息。 用户模块向信息模块发送用户名；向信息模块发送用户名；信件模块发送用户名。

信息模块向编辑个人信息模块发送用户名和用户修改的信息；向显示个人信息模块发送用户名。

信件模块向写信模块发送收、发信件的用户名和信件内容；向读信模块发送用户名。

物品模块向上传物品模块发送新物品的介绍和参数信息；向编辑物品模块发送物品的修改信息；向浏览物品模块发送用户输入的关键字。

浏览物品模块向交换物品模块发送交换双方物品的编号。

管理员模块向个人信息模块发送管理员编号；向管理用户模块发送用户名；审核物品模块发送审核物品编号的审核意见；向发布公告模块发送用户名集合和公告内容。

个人信息模块向编辑个人信息模块发送管理员编号和管理员修改的信息；向显示个人信息模块发送管理员编号。

公告模块向发布公告模块发送用户名集合和公告内容；向阅读询问模块发送用户名。

# 数据结构设计

## 公共数据结构设计

$\_POST：连接网页前端和数据库后端的PHP语言预定义的全局变量，用于收集来自 method = "post" 的表单中的值。

$\_FILES：通过 HTTP POST 方式上传到当前脚本的项目的数组，是一个自动化的全局变量。

$CONN：脚本文件中对数据库创建的 mysqli 对象，使用 $CONN 对象和对象的方法对数据库进行操作。

$JSON：全局数组，根据前后端约定，在特定位置填写数据库的查询结果的特定字段，使用 json\_encode () 方法，返回JSON数组的格式。

$DBNAME：常量，为数据库的名字。

## 数据库设计

**数据流图**







**数据库表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Student | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Student\_no | 学号 | CHAR (8) | ‘0’ <= char <= ‘9’ | 键 | Sno |
| Student\_password | 密码 | VARCHAR  (40) | char | NOT NULL | Spassword |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| User | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| User\_name | 用户名 | VARCHAR (40) | char | 键 | Uname |
| Student\_no | 学号 | CHAR (8) | ‘0’ <= char <= ‘9’ | 外码 | Sno |
| User\_root | 用户权限 | VARCHAR (40) | “用户” or  “管理员” | NOT NULL | Uroot |
| User\_sexy | 用户性别 | VARCHAR (40) | “男” or “女” | NOT NULL | Usexy |
| User\_credit | 用户信用 | INT | [-1000, 1000] | NOT NULL | Ucredit |
| User\_address | 用户头像 | VARCHAR (40) | char | UNIQUE | Uaddress |
| User\_phone | 用户联系方式 | CHAR (11) | ‘0’ <= char <= ‘9’ | NOT NULL | Uphone |
| User\_email | 用户邮箱 | VARCHAR (40) | char | NOT NULL | Uemail |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Credit | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Credit\_level | 信用等级 | VARCHAR (40) | char | 键 | Clevel |
| Credit\_left  \_value | 信用度\_低值 | INT | [-1000, 1000] | UNIQUE | Cleft |
| Credit\_right  \_value | 信用度\_高值 | INT | [-1000, 1000] | UNIQUE | Cright |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Root | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Root\_no | 权限编号 | INT | Root\_no >= 1 | 键 | Rno |
| Root\_content | 权限内容 | VARCHAR (40) | char | UNIQUE | Rcontent |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Goods | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Goods\_no | 物品编号 | INT | Goods\_no >= 1 | 键 | Gno |
| Goods\_name | 物品名称 | VARCHAR (40) | char | NOT NULL | Gname |
| Goods\_user  \_name | 物品  所有者 | VARCHAR (40) | char | 外码 | Uname |
| Goods\_type | 物品类型 | VARCHAR (40) | char | NOT NULL | Gtype |
| Goods\_state | 物品状态 | VARCHAR (40) | 有限状态 | NOT NULL | Gstate |
| Goods\_address | 物品图片 | VARCHAR (40) | char | UNIQUE | Gaddress |
| Goods\_check | 物品审核 | VARCHAR (40) | 有限状态 | NOT NULL | Gcheck |
| Goods  \_timestamp | 物品审核时间 | TIMESTAMP | CURRENT  \_TIMESTAMP | NOT NULL | G  timestamp |
| Goods  \_instruction | 物品简介 | VARCHAR (400) | char |  | G  instruction |
| Goods  \_parameter | 物品参数 | VARCHAR (400) | char |  | G  parameter |
| Goods\_time | 使用年限 | INT | Goods\_time >= 0 |  | Gtime |
| Goods\_price | 估计价格 | INT | Goods\_prive >= 0 |  | Gprice |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Tag | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Tag \_no | 标签编号 | VARCHAR (40) | 有限种标签 | 键 | Tno |
| Tag \_content | 标签内容 | VARCHAR (40) | char | UNIQUE | Tcontent |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Message | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Message\_no | 信件编号 | INT | Message\_no >= 1 | 键 | Mno |
| Message  \_content | 信件内容 | VARCHAR (400) | char | UNIQUE | Mcontent |
| Message  \_timestamp | 信件时间 | TIMESTAMP | CURRENT  \_TIMESTAMP | NOT NULL | M  timestamp |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | | 完成日期 |
| User  \_Administrator  \_Root | 何玥 | 周啸辰 | | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | | 缩写词 |
| User\_root | 用户类型 | VARCHAR (40) | “用户” or  “管理员” | 键 |  | Uroot |
| Root\_no | 权限编号 | INT | Root\_no >= 1 | 外码 | Rno |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | | 完成日期 |
| Describle | 何玥 | 周啸辰 | | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | | 缩写词 |
| Goods\_no | 物品编号 | INT | Goods\_no >= 1 | 键 | 外码 | Gno |
| Tag\_no | 标签编号 | VARCHAR (40) | 有限种标签 | 外码 | Tno |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | | 完成日期 |
| Search | 何玥 | 周啸辰 | | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | | 缩写词 |
| User\_name | 用户名 | VARCHAR (40) | char | 键 | 外码 | Uname |
| Goods\_no | 物品编号 | INT | Goods\_no >= 1 | 外码 | Gno |
| BRO\_timestamp | 浏览时间 | TIMESTAMP | CURRENT  \_TIMESTAMP | NOT NULL | | BRO  timestamp |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Charge | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Goods\_plan | 申请方  物品编号 | INT | Goods\_plan >= 1 | 键 | Gnoplan |
| Goods\_adopt | 接收方  物品编号 | INT | Goods\_adopt >= 1 | 外码 | Gnoadopt |
| Charge\_money | 补贴差价 | INT | Charge\_money >= 0 | NOT NULL | CHA  money |
| State\_plan | 申请方  交换意见 | VARCHAR (40) | 有限状态 | NOT NULL | CHA  planstate |
| Credit\_plan | 对接收方评价 | INT | [-5,5] | NOT NULL | CHA  plancredit |
| State\_adopt | 接收方  交换意见 | VARCHAR (40) | 有限状态 | UNIQUE | CHA  adoptstate |
| Credit\_adopt | 对申请方评价 | INT | [-5,5] | NOT NULL | CHA  adoptcredit |
| Charge  \_timestamp | 物品审核时间 | TIMESTAMP | CURRENT  \_TIMESTAMP | NOT NULL | CHA  timestamp |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据库的表结构 | | 表名称 | 设计者 | 审核者 | 完成日期 |
| Notify | 何玥 | 周啸辰 | 2016.10.29 |
| 字段代号 | 名称 | 类型 | 值域 | 索引或键 | 缩写词 |
| Message\_no | 信件编号 | INT | Message\_no >= 1 | 键 | Mno |
| User\_name\_send | 发信人 | VARCHAR (40) | char | NOT NULL | User\_name\_send |
| User\_name\_receive | 收信人 | VARCHAR (40) | char | NOT NULL | User\_name\_receive |

# 详细设计

每一小节给出一个模块（构件）的详细设计方案。包括模块概述、模块的接口说明（即输入、输出）、以及内部结构设计。其中内部结构又可以考虑从静态、动态结构两个方面阐述；静态结构应给出该模块（构件）的类结构（类图），动态结构应给出该模块关键业务流程的交互模型（顺序图），还可根据实际情况给出状态图（某个构件或对象的状态迁移）和活动图（某个算法的实现流程）等内容。

1. **校园二手物品交换平台模块**

**模块描述：**

系统中与用户交互的第一个模块，是系统的最顶层的模块。

**输入：**

用户的请求与上传的信息。

**输出：**

响应用户的请求，显示用户操作和查询的结果和数据。

**静态结构：**

类名：home

属性：

Login\_Button 登录按键；

Register\_Button 注册按键；

一些提供显示组件。

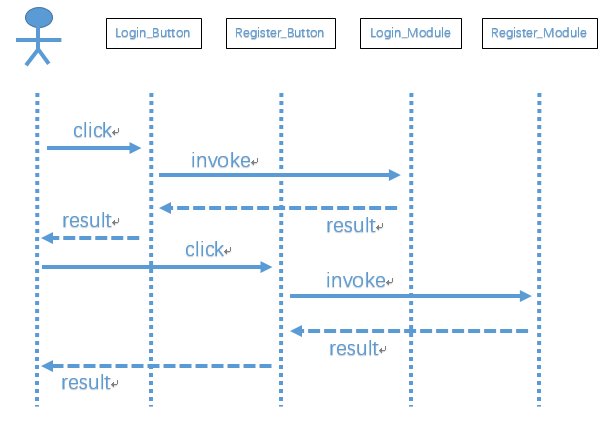
方法：

\_Ini ()方法初始化各组件，并调用系统主页。

Login\_Button\_Listener ()，响应用户的登录操作，调用登录模块，根据登录模块的返回结果，显示信息，或者调用用户模块、管理员模块。

Register\_Button\_Listener ()，响应用户的注册操作，调用注册模块，并显示注册模块的返回结果。

**动态结构（交互模型）：**

****

1. 注册模块

模块描述：

系统中用户在平台上注册账户的模块。

输入：

用户的信息，包括学号、校园密码、用户名、密码、联系方式和邮箱地址。

输出：

用户的注册结果，注册成功或者注册失败，一个学号在平台上只能注册一个账户。

静态结构：

类名：register

属性：

Register\_Frame 注册输入框；

Register\_Comfirm\_Button 注册提交按键；

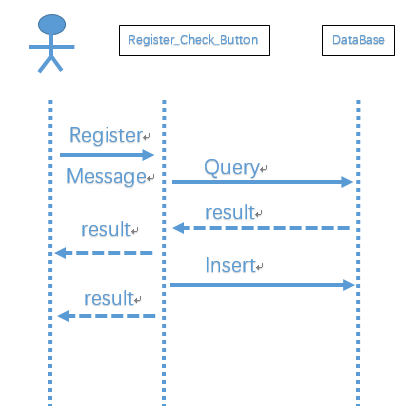
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，并调用注册页面。

Register\_Comfirm\_Button\_Listener ()，记录用户在输入框填入的注册信息，并使用数据库完成查询和插入新用户信息的工作。

动态结构（交互模型）：



1. 登录模块

模块描述：

系统中用户登录账户的模块。

输入：

用户的用户名、密码和账户类型。

输出：

如果用户的登录信息输入正确，用户进入用户模块或者管理员模块；如果用户的登录信息错误，输出错误提示信息。

静态结构：

类名：login

属性：

Login\_Frame 登录信息输入框；

Login\_Comfirm\_Button 登录提交按键；

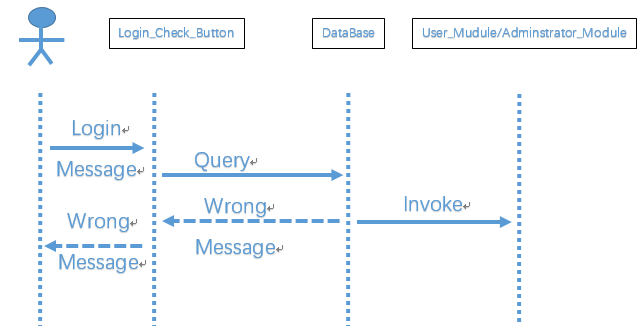
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，并调用登录页面。

Login\_Comfirm\_Button\_Listener ()，记录用户在输入框填入的登录信息，并使用数据库完成查询用户信息的工作，如果用户登录信息正确，调用用户模块或者管理员模块。

动态结构（交互模型）：



1. 用户模块

模块描述：

系统中用户（非管理员）登录后进入的模块，完成用户在平台上的最外层操作，用户的请求调用相应的模块，进入相应的页面。

输入：

用户的操作请求。

输出：

根据用户的请求，调用相应的模块，进入相依的页面。

静态结构：

类名：user\_home

属性：

Edit\_Buttom 用户查看个人详细信息和编辑个人信息的按键；

Goods\_Button 用户进入物品页面的按键；

Message\_Button 用户收发信件的按键。

一些提供显示组件。

方法：

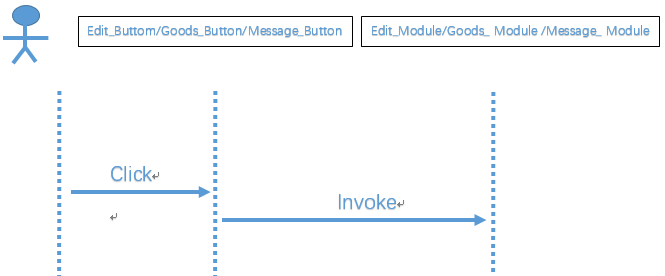
\_Ini ()方法初始化各组件，并调用登录页面。

Edit\_Buttom\_Listener ()，进入个人详细信息页面，调用个人信息模块。

Goods\_Button \_Listener ()，进入物品页面，调用物品模块。

Message\_Button\_Listener ()，进入信件页面，调用信件模块。

动态结构（交互模型）：



1. 个人信息模块

模块描述：

系统中用户（非管理员）编辑个人信息并查询个人详细信息的模块。

输入：

用户编辑个人信息的请求。

输出：

用户的个人详细信息。

静态结构：

类名：user\_information

属性：

Edit\_Buttom 用户请求编辑个人信息的按键；

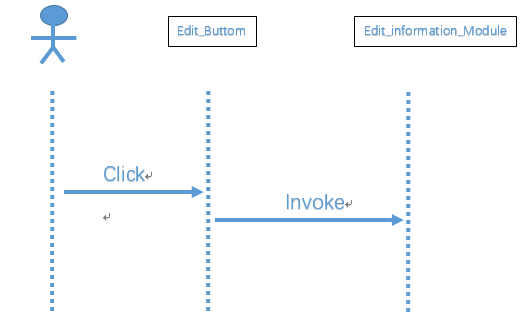
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，自动调用显示个人信息模块，显示个人信息。

Edit\_Buttom\_Listener ()，调用编辑个人信息模块。

动态结构（交互模型）：



1. 编辑个人信息模块

模块描述：

系统中用户（非管理员）编辑个人信息的模块。

输入：

用户输入修改的个人信息、上传的个人头像。

输出：

用户编辑个人信息结果。

静态结构：

类名：user\_edit\_information

属性：

Edit\_Frame 用户编辑个人信息的输入框；

Edit\_image 用户上传个人头像；

Edit\_confirm\_Button 用户确认提交个人信息的按键；

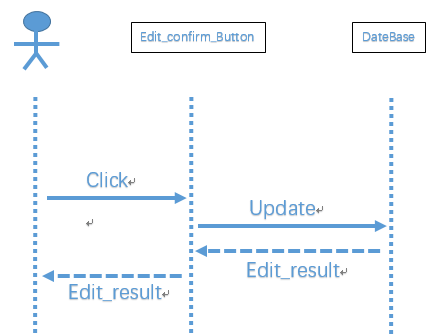
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，显示基本界面。

Edit\_confirm\_Buttom\_Listener ()，提交用户修改的个人信息。

动态结构（交互模型）：



1. 显示个人信息模块

模块描述：

查询并返回用户（非管理员）个人信息的模块。

输入：

用户名。

输出：

数据库中用户的个人信息的json数组。

静态结构：

类名：user\_show\_information

属性：

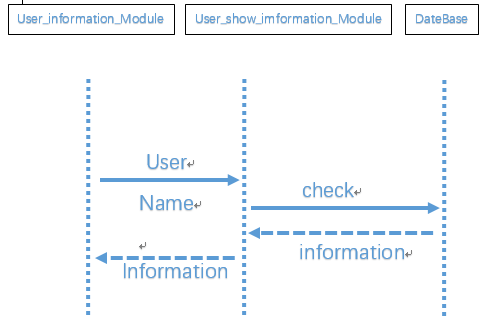
User\_name 记录用户名。

方法：

\_Ini ()记录查询的用户名。

check\_information ()，在数据库中查询用户的详细个人信息，并打包成Json数组，返回给调用模块。

动态结构（交互模型）：



1. 物品模块

模块描述：

系统中用户上传、编辑、浏览、交换物品的最外层模块。

输入：

用户的操作和请求，输入的查询信息。

输出：

用户的个人物品、查询物品的结果。

静态结构：

类名：goods

属性：

Add\_goods\_Button 上传物品的按键；

Edit\_goods\_Button 物品的按键；

Search\_frame 查询条件的输入框；

Search\_ Button 查询浏览物品的按键；

一些提供显示组件。

方法：

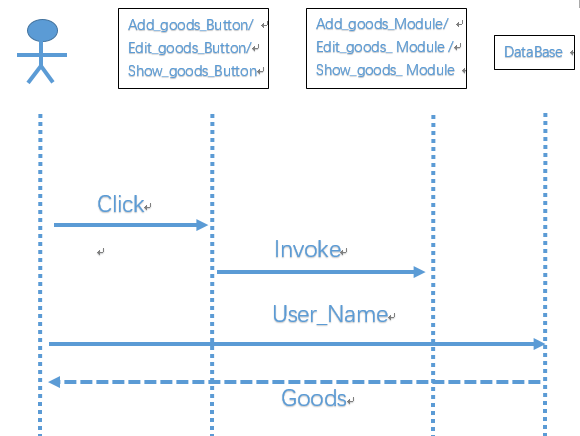
\_Ini ()方法初始化各组件，显示基本界面。

Add\_goods\_Button\_Listener () 调用添加物品模块。

Edit\_goods\_Button\_Listener () 调用编辑物品模块。

Search\_Button\_Listener () 调用浏览物品模块。

动态结构（交互模型）：



1. 信件模块

模块描述：

系统中用户和其他用户、和管理员发送和接收信件的模块。

输入：

收信用户名或者管理员和信的内容。

输出：

用户的全部信件的信件。

静态结构：

类名：message

属性：

Write\_message\_frame 写信的输入框；

Read\_message\_frame 读信的展板；

Send\_message\_Button 发送信件的按键；

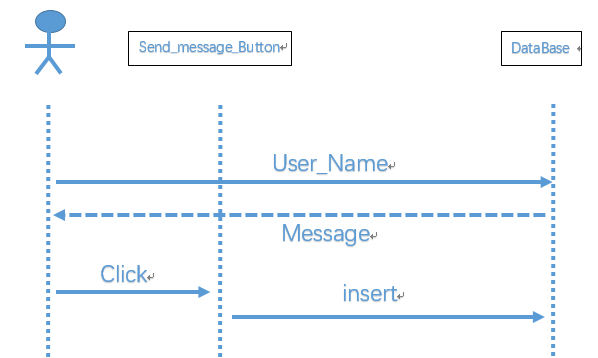
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，显示基本界面和用户的全部历史信件。

Send\_message\_Button\_Listener () 在数据库中插入这封信件。

动态结构（交互模型）：



1. 上传物品模块

模块描述：

系统中用户上传物品的模块。

输入：

用户上传的物品的名称、类型、图片、简介、参数、估计价格等具体信息，上传的物品不会立即进入市场中，需要等待管理员的审核。

输出：

用户上传物品的结果。

静态结构：

属性：

Goods\_information\_frame 写物品具体信息的输入框；

Submit\_goods\_Button 提交物品的按键；

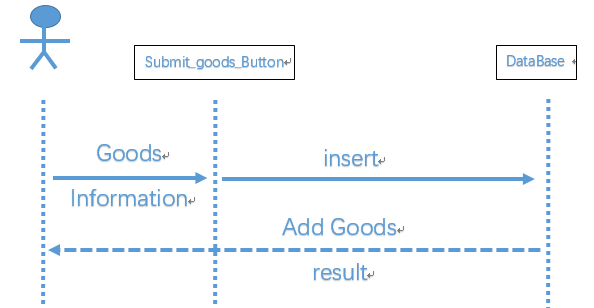
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，显示基本界面。

Submit\_goods\_Button \_Listener () 在数据库中插入物品信息，等待管理员审核。

动态结构（交互模型）：



1. 编辑物品模块

模块描述：

系统中用户管理和编辑物品的模块。

输入：

用户编辑的物品的名称、类型、图片、简介、参数、估计价格等具体信息以及删除物品的请求，修改信息的物品不会立即进入市场中，需要等待管理员的审核。

输出：

用户修改物品的信息或删除物品结果。

静态结构：

属性：

Goods\_information\_frame 修改物品具体信息的输入框；

Submit\_goods\_Button 提交修改物品的按键；

Delete\_goods\_Button 提交修改物品的按键；

一些提供显示组件。

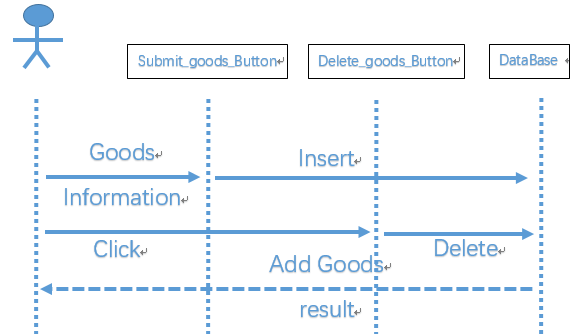
方法：

\_Ini ()方法初始化各组件，显示基本界面。

Submit\_goods\_Button \_Listener () 在数据库中更新物品信息，等待管理员审核。

Delete\_goods\_Button \_Listener () 在数据库中删除物品，。

动态结构（交互模型）：



1. 浏览物品模块

模块描述：

系统中用户浏览物品并申请物品交易的模块。

输入：

用户输入的查找条件和申请交易的操作。

输出：

符合用户查询条件的物品。

静态结构：

属性：

Search\_frame 输入查找条件的输入框；

Search\_goods\_Button 查找物品的按键；

Apply\_Charge\_goods\_Button 申请交换物品模块；

一些提供显示组件。

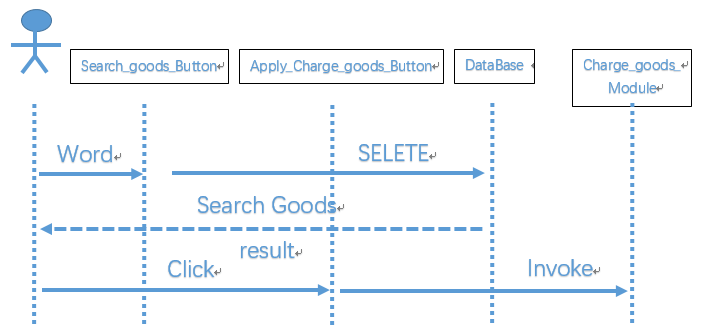
方法：

\_Ini ()方法初始化各组件，显示基本界面。

Search\_goods\_Button\_Listener () 在数据库查找并返回符合条件的物品。

Apply\_Charge\_goods\_Button\_Listener () 调用交换物品模块。

动态结构（交互模型）：



1. 交易物品模型

模块描述：

系统中用户交换物品的模块。

输入：

用户希望得到的物品、想要交换的物品和补贴的差价。

输出：

用户全部交换物品的交易情况。

静态结构：

属性：

Charge\_frame 输入交换信息的输入框；

Credit\_frame 信用评价的输入框；

Charge\_goods\_Button 交换物品的按键；

Delete\_goods\_Button 取消交换的按键；

Confirm\_goods\_Button 确认交换的按键；

一些提供显示组件。

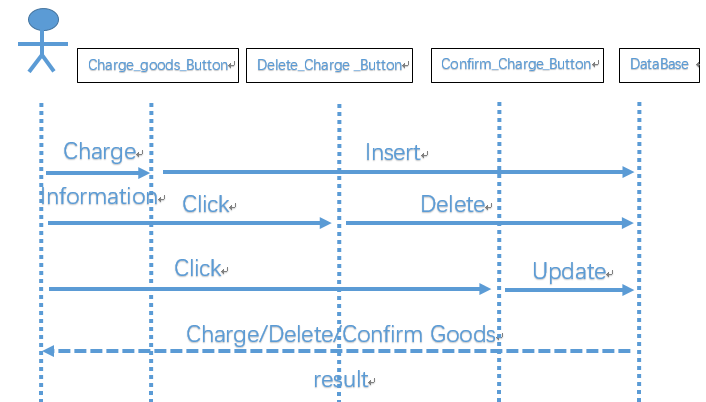
方法：

\_Ini ()方法初始化各组件，显示基本界面。

Charge\_goods\_Button\_Listener () 在数据库插入物品的交换信息。

Delete\_goods\_Button\_Listener () 在数据库中删除此交换信息，取消交换Confirm\_goods\_Button\_Listener () 确认交换，并根据双方评价，修改用户的信用度。

动态结构（交互模型）：



1. 个人信息模块

模块描述：

系统中管理员编辑个人信息并查个人详细信息的模块。

输入：

管理员编辑个人信息的请求。

输出：

管理员的个人详细信息。

静态结构：

类名：admintrator\_information

属性：

Edit\_Buttom 管理员请求编辑个人信息的按键；

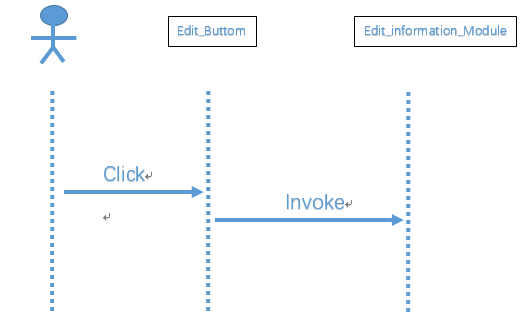
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，自动调用显示个人信息模块，显示个人信息。

Edit\_Buttom\_Listener ()，调用编辑个人信息模块。

动态结构（交互模型）：



1. 编辑个人信息模块

模块描述：

系统中管理员编辑个人信息的模块。

输入：

管理员输入修改的个人信息。

输出：

管理员编辑个人信息结果。

静态结构：

类名：admintrator\_edit\_information

属性：

Edit\_Frame 管理员编辑个人信息的输入框；

Edit\_confirm\_Button 管理员确认提交个人信息的按键；

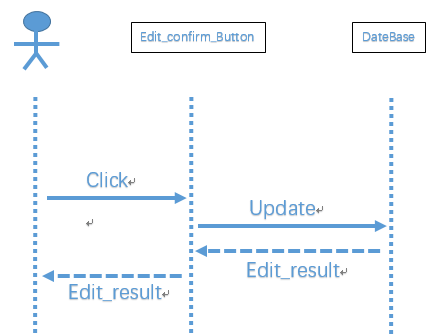
一些提供显示组件。

方法：

\_Ini ()方法初始化各组件，显示基本界面。

Edit\_confirm\_Buttom\_Listener ()，提交管理员修改的个人信息。

动态结构（交互模型）：



1. 显示个人信息模块

模块描述：

查询并返回管理员个人信息的模块。

输入：

管理员编号。

输出：

数据库中管理员的个人信息的json数组。

静态结构：

类名：adminstrator\_show\_information

属性：

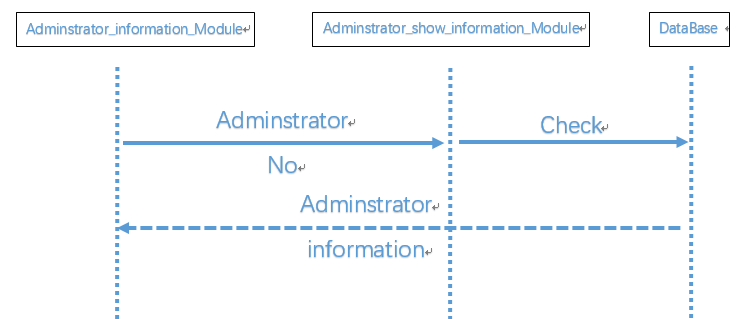
adminstrator\_no 记录管理员编号。

方法：

\_Ini ()记录查询的管理员编号。

check\_information ()，在数据库中查询管理员的详细个人信息，并打包成Json数组，返回给调用模块。

动态结构（交互模型）：



1. 管理用户模块

模块描述：

管理员管理平台上全部用户、调整用户信用度的模块。

输入：

查看某位用户的个人信息和全部物品，修改的用户信用度。

输出：

显示用户的信息，修改用户信用度的结果。

静态结构：

类名：manage\_user

属性：

Credit\_Frame 修改信用度的输入框。

Credit\_Button 确认修改信用度的按键。

Show\_user 显示用户信息的按键。

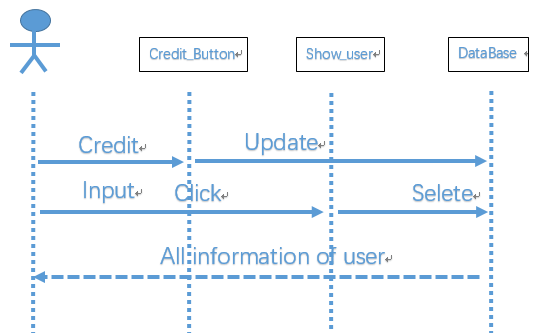
方法：

\_Ini ()方法初始化各组件，显示基本界面。

Credit\_Button\_Listener ()，根据输入的信用度值，修改数据库中用户的信用度。

Show\_user\_Listener()，根据数据库中的信息，显示某位用户的个人信息和物品。

动态结构（交互模型）：



1. 审核物品模块

模块描述：

管理员审核用户上传的物品的模块。

输入：

用户审核物品的结果。

输出：

所有未被审核的物品信息。

静态结构：

类名：check\_goods

属性：

Check\_Frame 审核结果的输入框。

Check\_Button 确认审核结果的按键。

Show\_goods 显示物品信息的按键。

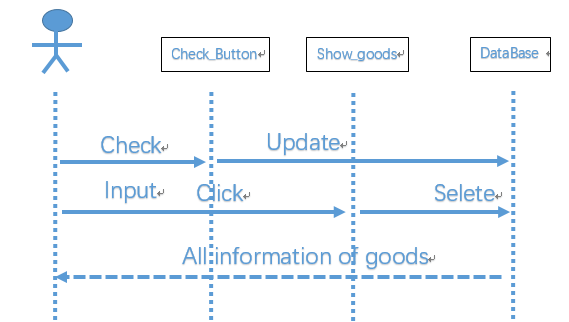
方法：

\_Ini ()方法初始化各组件，显示基本界面。

Check\_Button \_Listener ()，根据输入的审核结果，修改数据库中物品的审核信息。

Show\_goods\_Listener()，根据数据库中的信息，显示物品的详细信息。

动态结构（交互模型）：



1. 公告模块

模块描述：

管理员发布公告或回答用户提问的模块。

输入：

管理员发布的公告，回复的信息。

输出：

所有用户的提问。

静态结构：

类名：notify

属性：

Answer\_Frame 回复用户提问的输入框。

Answer\_Button 回复用户提问的按键。

Notify\_Frame 发布公告信息的输入框。

Notify\_Button 发布公告信息的按键。

方法：

\_Ini ()方法初始化各组件，显示基本界面。

Answer\_Button \_Listener ()，提交回复用户的提问的信件。

Notify\_Button \_Listener()，提交系统公告。

动态结构（交互模型）：

