**网上物物交换系统——Uchange设计说明书**

# 1、引言

## 1.1编写目的

在完成Uchage系统需求分析的基础上，我们依据《Uchange需求规格说明书》对Uchange系统进行了设计。为了文档化对系统的设计，编写本文档。本文档主要供团队在具体实现Uchange时参考使用。

## 1.2命名规范

数据库表名、字段名用大写字母开头的英文命名。

私有变量用“-”开头的小写英文命名。其他的变量用大学英文字母开头。

所有的函数用大写英文字母开头。

## 1.3术语定义

总体结构：软件系统的总体逻辑结构，本系统采用面向对象的方法对系统进行设计。

概念数据模型（CDM）：关系数据库的系统概念设计模型，主要表现为E-R图。

物理数据模型（PDM）：关系数据库的物理设计模型。

## 1.4相关文档

Uchange需求规格说明书。

## 1.5版本更新信息

版本更新记录见表1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 创建者 | 创建时间 | 维护者 | 维护日期 | 维护记录 |
| V1.0 | 王汉雄 | 2013-04-06 | 王汉雄 |  |  |
| V1.1 | 王汉雄 | 2013-05-15 | 王汉雄 | 2013-05-16 | 根据实际需求修改模块功能 |
| V2.0 | 王汉雄 | 2013-09-15 | 王汉雄 | 2013-09-15 | 加入封面、目录、详细的数据字典和存储过程 |

# 2、总体设计

## 2.1硬件运行环境

（1）、服务器：PII300 128M 10G以上

（2）、客户端：P166 32M 1G以上

（3）、网络宽带局域网支持10M/100Mbps

Internet带宽：56Kbps拨号上网以上

## 2.2软件运行环境

服务器端：

Windows7以上版本或红旗Linux。

数据库服务器：

SQL Sever2008以上

客户端：  
 IE6.0以上

## 2.3子系统清单

|  |  |  |
| --- | --- | --- |
| 子系统编号 | 子系统名称 | 子系统功能简述 |
| SS1 | 登录注册 | 1. 会员登录系统时，对其身份经行检查和识别。 2. 游客（新用户）可以进行注册。 3. 已注册用户可以修改个人信息，找回密码，注销等。 |
| SS2 | 浏览物品 | 用户可以浏览其他会员发布的物品。 |
| SS3 | 会员发布/修改物品 | 会员可以发布自己拿出来交换的物品以供其他用户浏览。并在发布后一段时间内可以修改对该物品描述信息。 |

## 2.4功能模块清单

|  |  |  |
| --- | --- | --- |
| 模块编号 | 名称 | 模块功能简述 |
| SS1-1 | 用户注册 | 游客注册，成为会员。 |
| SS1-2 | 会员登录 | 会员登录系统。 |
| SS1-3 | 找回密码 | 会员将个人密码遗失后，通过对预留信息的审核重新获得密码。 |
| SS1-4 | 修改个人信息 | 会员登录后进行资料管理，如联系方式修改等。 |
| SS1-5 | 用户注销 | 会员离开系统，进行注销。 |
| SS2-1 | 浏览该物品信息 | 用户可以浏览指定物品的基本信息如：拥有方提供的物品、发布方最希望换到的物品和该物品现在此有的申请数 |
| SS2-2 | 浏览物品所在交易链 | 用户可以查阅自己手上的物品所在的交易链中的信息，如：交易链链头交易需要的物品和交易链链尾交易提供的物品。 |
| SS2-3 | 查看自己发出的申请 | 会员对自己向他人的物品发出申请的详细状态进行查询 |
| SS3-1 | 发布物品 | 为会员提供发布物品的接口，并向会员提出按要求填写信息的要求。 |
| SS3-2 | 修改物品信息 | 会员在发布物品后一段时间内可以对该物品的信息进行修改。 |
| SS3-3 | 物品信息填写 | 会员在发布物品或修改物品信息时需要填写的物品信息表。 |
| SS3-4 | 关闭物品 | 会员向系统提交对已发布的物品的关闭申请。 |
| SS3-5 | 确认申请 | 会员可以查看自己所拥有物品持有的申请的详细信息，并可以选择同意该申请。 |
| SS3-6 | 同意交易 | 会员在确认他人对与自己所持物品发出的申请后，可以同意该申请，同意后，在系统中双方将会显示持有对方的物品，所有对于对方物品的请求也将会转移。 |
| SS3-7 | 完成交易 | 在经过一定的周期（暂定为一周）后，系统将会在一段时间内关闭一部分功能（提出申请和同意交易），并为会员提供他们最终需要联系的两人的联系信息（该给予物品的人与该获取物品的人），在此期间会员与他们联系并在现实世界中完成交换的步骤。此后下一个周期将会开始，所有申请将被清空，会员可以发布新一批自己想要拿出去交换的物品。 |

# 3、数据库设计

## 3.1 数据库中的表

|  |  |  |
| --- | --- | --- |
| 编号 | 表名 | 表功能说明 |
| 1 | Item | 用于存储物品的信息 |
| 2 | Person | 用于存储用户的信息 |
| 3 | Request | 用于存储用户突出的交换申请 |
| 4 | Deal | 用于存储所有完成后的交易记录 |

## 3.2数据库表间的关系

## 3.3数据库表的详细清单

Item表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是为允许为空 | 主/外键 | 长度 | 默认值 | 说明 |
| 1 | id | 物品编号 | Int | 否 | 主键 | 8 |  | 物品标识 |
| 2 | name | 物品名字 | Char() | 是 |  | 100 |  | 物品的名字 |
| 3 | description | 物品描述 | Char(500) | 是 |  |  |  | 用户对该物品的描述 |

Person表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主/外键 | 长度 | 默认值 | 说明 |
| 1 | Id | 学生编号 | Int | 否 | 主键 |  |  | 学生标识 |
| 2 | student\_id | 学号 | Char() | 否 |  | 10 |  | 学生的学号 |
| 3 | first\_name | 名 | Char() | 是 |  | 50 |  | 学生的名字 |
| 4 | last\_name | 姓 | Char() | 是 |  | 50 |  | 学生的姓氏 |
| 5 | email | 邮箱地址 | Char() | 是 |  | 50 |  | 学生的邮箱 |
| 6 | Item\_now | 现有物品 | Int | 否 | 外键 | 8 |  | 现在持有的物品 |
| 7 | Item\_origual | 发布物品 | Int | 否 | 外键 | 8 |  | 该生在本轮交换中发布的物品 |

Deal表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主/外键 | 长度 | 默认值 | 说明 |
| 1 | Id | 交易编号 | Int | 否 | 主键 | 8 |  | 交易标识 |
| 2 | item | 物品 | Char() | 否 | 外键 | 100 |  | 此交易记录所涉及的物品 |
| 3 | P1 | 原物主 | Char() | 否 | 外键 | 10 |  | 交易前的物主 |
| 4 | P2 | 物主 | Char() | 否 | 外键 | 10 |  | 交易后的物主 |
| 5 | Deal\_time | 交易时间 | time | 是 |  |  |  | 交易确认的时间 |

Request表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主/外键 | 长度 | 默认值 | 说明 |
| 1 | id | 申请编号 | Int | 否 | 主键 | 8 |  | 请求标识 |
| 2 | Person | 申请者 | Char() | 否 | 外键 | 10 |  | 申请者的学号 |
| 3 | Item | 物品 | Char() | 否 | 外键 | 100 |  | 物品的名称 |

# 4、功能模块设计

此项目中共涉及到8个页面，分别是：Login、Register、StudentInfo、ItemInfo、MyRequests、ItemList、MyItemRequests、ItemForm。

## 4.1、登录页面：Login

调用背景：

用户连接到网页后，进入的页面。

页面组成：

由一个登录框组成。登录框包含用户名和密码两个文本框、一个登录按钮和一个注册按钮。

调用描述：

（1）当前用户没有登录，在用户名和密码填入后，单击“登录”按钮。在数据库中进行查询，若存在此记录，跳转到“StudentInfo”页面。若没有此记录，则提示登录失败信息。

（2）单击“注册”按钮，则页面跳转到“Register”页面。游客进行注册。若注册成功，则提示注册成功，之后，页面跳转到“StudentInfo”页面。

## 4.2、注册页面：Register

调用背景：

当用户在“Login”页面中单击“注册”按钮后，跳转到此页面。

页面组成：

由一个表格构成，表格中的项即是要填写的详细个人信息。表格下面有“确认”按钮。

调用描述：

点击确认以后，会在数据库中检索是否有重复的student\_id，并且检验是否有没填的必填项（必填项以\*标出），如填写正确，则提示注册成功，跳转到“Login”页面。否则提醒注册失败并将部分文本框清空。

## 4.3、个人信息页面：StudentInfo

参数：

id=？，指定显示哪位会员的信息。

调用背景：

（1）当用户登录成功后，自动跳转到此页面。

（2）当用户在别的页面点击“个人信息”按钮时，跳转到此页面。

页面组成：

由一个表格构成，表格中有姓、名、邮箱和自己拥有的物品列表。其中拥有的物品均为超链接。表格一侧有“发布物品”按钮、“我发出的请求”按钮、“物品列表”按钮和“我收到的请求”按钮。

调用描述：

点击物品超链接以后，会在数据库中检索对应ID的Item，然后跳转到ItemInfo（？id=？）页面。点击“发布物品”按钮，跳转到ItemForm页面。点击“我发出的请求”按钮，会在数据库中检索对应ID的请求，然后跳转到MyRequest（？id=？）页面。点击“物品列表”按钮时，跳转到ItemList页面。点击“我收到的请求”按钮时，会在数据库检索当前用户所拥有的物品，然后跳转到MyItemRequest（？id[]=？）页面。

## 4.4、物品信息页面：ItemInfo

参数：

id=？，指定显示哪件物品的信息。

调用背景：

（1）当用户在“StudentInfo”页面点击某样物品的超链接时跳转到此页面。

（2）当用户在“ItemList”页面点击某样物品的超链接时跳转到此页面。

页面组成：

由两个表格构成。第一个表格中有物品的名字和描述等信息。另外一个表中还有该物品的持有者信息，并且显示出该物品的交易链信息。当查看着不是这件物品现在的持有者时，会出现“发送请求”按钮。

调用描述：

点击“发送请求”按钮后，会在数据库中检索是否有重复的用户——物品请求，如果有，则弹出重复错误提示。如果没有，则创建请求。

## 4.5我的请求页面：MyRequest

参数：

id=？，指定显示哪位会员发出的申请。

调用背景：

当用户在StudentInfo页面点击“我发出的请求”按钮时，跳转到此页面。

页面组成：

页面由一个表格组成，表格中记录这用户所发出的所有有效请求信息，以及物品名均为超链接。

调用描述：

点击物品超链接会以id在数据库中检索物品并跳转到ItemInfo（？id=？）页面。

## 4.6物品列表页面：ItemList

调用背景：

当用户在“StudentInfo”页面点击“物品列表”按钮时跳转到此页面。

页面组成：

页面有一张表格组成，表中各显示现在在线的所有有效物品，所有物品均为超链接。

调用描述：

点击物品超链接，会在数据库中检索相应物品ID的物品信息并跳转到ItemInfo（？id=？）页面。

## 4.7收到的请求页面：MyItemRequest

参数：

Id[]=?，指定需要显示的多个物品。

调用背景：

用户在“StudentInfo”页面点击“我收到的请求页面”按钮时跳转到此页面。

页面组成：

页面由一个表构成，其中表项分为两级，一级项为当前用户所持有的物品，二级项为该物品所收到的请求（显示为对方所持有的物品）。其中所有的物品名均为超链接。

调用描述：  
 点击物品超链接，会在数据库中检索相应物品ID的物品信息并跳转到ItemInfo（？id=？）页面。

## 4.8物品信息填写页面：ItemForm

调用背景：  
 用户在“StudentInfo”页面点击“发布物品”按钮时跳转到此页面。

页面组成：

页面由一个信息收集单元构成，该单元中有两个文本框分别对应物品的名称及描述。文本框下方有发布“确认”按钮。

调用描述：

在点击“确认”按钮后，会检验是否有没填的必填项（必填项以\*标出），如填写正确，则会提醒发布成功，在数据库中增加相应的条目后跳转到“StudentInfo”页面。否则提醒发布失败并将部分文本框清空。

# 5、存储过程设计

## 5.1 涉及Item表的存储过程

1. GetAllItems(string studentID)

功能描述：得到除用户的当前物品外的所有物品的信息。

入口参数: 当前用户的账号。

出口参数： 物品集

伪语言实现：

CREATE PROCEDURE GetAllItems(string studentID)

AS

SELECT \* FROM item;

WHERE name NOT IN(

SELECT item\_now FROM person

WHERE student\_id = studentid);

RETURN;

2.ItemInfo(int id)

功能描述：得到选定物品的基本信息。

入口参数：选定物品的标识。

出口参数：选定物品的详细信息。

伪语言实现：

CREATE PROCEDURE ItemInfo(int id)

AS

SELECT \* FROM item;

WHERE id=item.id;

RETURN;

3.EditItem(int Id,string Name,string Description)

功能描述：修改选定物品的名字和描述。

入口参数：Id:选定物品的标识，Name:修改后的名字，Description:修改后的描述。

出口参数：无。

伪语言实现：

CREATE PROCEDURE EditItem(int Id,string Name,string Description)

AS

UPDATE item;

SET name=Name,description=Ddscription;

WHERE id=Id;

## 5.2涉及Person表的存储过程

1.GetPersonInfo(int Id)

功能描述：得到当前用户的信息

入口参数：当前用户的标示

出口参数：当前用户的信息

伪语言实现：

CREATE PROCEDURE GetPersonInfo(int Id)

AS

SELECT student\_id,first\_name,last\_name,item\_now;

FROM person;

WHERE student\_id=Id;

2.GetItemOwner(int Id)

功能描述：获取选定物品的当前所有者信息

入口参数：选定物品的标示

出口参数：选定物品的当前所有者的信息

伪语言实现：

CREATE PROCEDURE GetItemOwner(int Id)

AS

SELECT student\_id,first\_nname,last\_name

FROM person

WHERE item\_now=Id;

RETURN;

## 5.3涉及DEAL表的存储过程

1.GetDealChain(int Id)

功能描述：获取选定物品的交易链

入口参数：选定物品的标示

出口参数：交易链集

伪语言实现：

CREATE PROCEDURE GetDealChain(int Id)

AS

SELECT p1,p2,deal\_time

FROM deal;

WHERE item=id;

RETURN;

## 5.4涉及Request表的存储过程

1.GetItemRequest(int id)

功能描述：获取选定物品的请求物品

入口参数：选定物品的标示。

出口参数：请求物品集

伪语言实现：

CREATE PROCEDURE GetItemRequest(int Id)

AS

SELECT R.Person,P..item\_now;

FROM Person P,Request R;

WHERE P.student\_id=R.Person,R.item=id;

RETURN;

2.NewRequest(int student\_Id,int item\_id)

功能描述：为选定物品添加新的请求

入口参数：item：选定物品的标示，student\_id：请求用户的标示

出口参数：无。

伪语言实现：

CREATE PROCEDURE NewRequset(int student\_id,int item\_id)

AS

INSERT Request VALURE

(student\_id,item\_id,system.Date);

# 6、角色授权设计

|  |  |  |
| --- | --- | --- |
| 模块 | 游客 | 会员 |
| 注册登录模块 | ● | ● |
| 浏览物品模块 |  | ● |
| 发布/修改物品模块 |  | ● |

# 7、系统错误处理

## 7.1、错误信息

（1）对会员输入的各项内容均进行有效性、安全性检查，减少错误发生几率。

（2）对程序运行中的一场均进行捕获，按统一的方式将出错提示信息提供给会员。

## 7.2故障预防与补救

以统一 的机制进行网站权限的控制。对程序中用到的数据尽量进行加密，以减少黑客攻击的危害，定期数据库中的数据进行海量备份以及增量备份。

## 7.3系统维护设计

（1）编码实现时采用模块化思想，令程序高内聚低耦合，使可维护性争强。

（2）面向数据与面向对象结合，模块符合面向对象思想。

# 8.测试设计

本系统测试将从单元测试、集成测试、系统测试3个步骤来进行。