**校叮当校园app项目总体设计报告**



|  |  |  |
| --- | --- | --- |
| 文件状态  【 】草稿  【 】正式发布  【√】正在修改 | 文件标识 | SE-2018春-G15-校叮当校园app总体设计报告 |
| 当前版本 | 0.2.0 |
| 作者 | 陈佳敏、徐毓茜、江亮儒、马益亮 |
| 完成日期 | 2018-5-16 |

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 参与者 | 起止日期 | 备注 |
| 0.1.0 | 陈佳敏、徐毓茜、江亮儒、马益亮 | 2018-5-4至2018-5-6 | 完成总体设计初稿，查找模板，各方面进行讨论书写。 |
| 0.2.0 | 陈佳敏、徐毓茜、江亮儒、马益亮 | 2018-5-14至2018-5-16 | 修改完善 |

**目录**

[一、 引言 5](#_Toc17754)

[1.1编写目的 5](#_Toc19395)

[1.2项目背景 5](#_Toc20927)

[1.3定义 6](#_Toc30871)

[1.4参考资料 9](#_Toc3159)

[二、 总体设计 11](#_Toc24621)

[2.1需求规定 11](#_Toc32669)

[2.1.1系统功能 11](#_Toc14309)

[2.1.2系统性能 13](#_Toc21752)

[2.1.3输入输出要求 16](#_Toc20202)

[2.1.4数据管理能力 16](#_Toc17600)

[2.1.5故障处理要求 17](#_Toc29354)

[2.1.6其他专门要求 17](#_Toc19288)

[2.2运行环境 21](#_Toc10239)

[2.2.1设备 21](#_Toc29737)

[2.2.2支持软件 21](#_Toc26297)

[2.2.3接口 21](#_Toc20880)

[2.2.4控制 23](#_Toc12907)

[2.3基本设计概念和处理流程 23](#_Toc12523)

[2.4结构 23](#_Toc678)

[2.5功能需求与系统模块的关系 24](#_Toc21618)

[2.6人工处理过程 25](#_Toc15881)

[2.7尚未解决的问题 25](#_Toc2178)

[三、 接口设计 25](#_Toc26317)

[3.1用户接口 25](#_Toc22650)

[3.2外部接口 26](#_Toc19639)

[3.3内部接口 26](#_Toc1893)

[四、 运行设计 27](#_Toc30614)

[4.1运行模块的组合 27](#_Toc21249)

[4.1.1用户功能模块组合 27](#_Toc22200)

[4.1.2前端信息汇聚、检索功能模块组合 27](#_Toc31711)

[4.1.3管理员功能模块组合 27](#_Toc25848)

[4.2运行控制 27](#_Toc9140)

[4.3运行时间 28](#_Toc11863)

[五、 系统数据结构设计 28](#_Toc23322)

[5.1逻辑结构设计要点 28](#_Toc19260)

[User表 28](#_Toc16965)

[Commodity表 29](#_Toc18125)

[Order表 29](#_Toc31869)

[Message表 30](#_Toc20740)

[Collection表 30](#_Toc29792)

[5.2物理结构设计要点 31](#_Toc25189)

[5.3数据结构与程序的关系 31](#_Toc7991)

[六、 系统出错处理设计 31](#_Toc26251)

[6.1出错输出信息 31](#_Toc8161)

[6.2出错补救措施 32](#_Toc27614)

[6.3系统维护设计 32](#_Toc28188)

[[1]](#footnote-0)

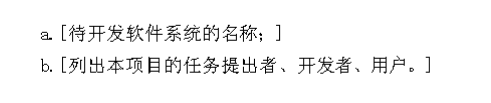
# 引言

## 1.1编写目的



本文档用于指导软件开发者开发基于安卓的浙江大学城市学院校叮当校园二手交易app的过程。通过规范开发人员的开发过程达到保证项目团队完成软件项目的基本要求，提高软件质量的目的。开发者应根据本文档进行软件开发和编制软件开发文档。本文档是对软件项目开发人员的基本要求。

## 1.2项目背景



1. 项目的名称

经过需求分析，现为学校学生设计了名为校叮当的校园系统项目。

1. 项目的委托单位

G15小组

1. 项目的用户：

浙江大学城市学院的在校大学生。

1. 项目的任务提出者

G15小组

1. 项目的主要承担部门

G15小组

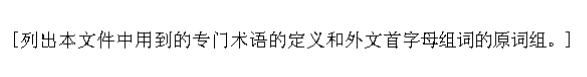
1. 项目背景

随着中国互联网用户群的日益庞大，互联网产业正扮演着市 场经济的重要角色。与此同时，信息时代的来临彻底改变了人们 传统的通信手段。从寄信到打电话，从打电话到通话手机，从通 话手机到能获取大信息量的智能手机，人们获取信息的主要手段 发生了质的变化。人们不会仅仅满足于手机只有通话与发短信的 功能，而更多是把手机当成接受讯息的工具，了解时事的工具， 建立社交网络的工具，便利生活的工具。而大学生对于各种信息 平台的接受度和依赖度普遍非常强，但是市面上的生活服务类应 用软件鱼龙混杂，针对大学生的需求，以大学生的利益为基本出 发点提供信息方便大学生的学习生活的应用软件还少之又少。互 联网技术和移动智能终端的飞速发展，促使越来越多的高校学生 更愿意通过移动智能终端平台连接互联网。于是针对大学生群体 的校园APP也诞生了。与此同时，我们应于课程要求，需要在 一个学期内开发项目，并在期末顺利结束，于是经过小组沟通， 我们决定做校叮当这样一个项目，以此项目计划书来展示整体计 划。

1. 软件系统与机构的关系

本软件不需外包，由G15小组独立研发。

## 1.3定义



|  |
| --- |
| 技术定义 |
| **JavaScript:** 这是一种直译式脚本语言，内置支持类型。浏览器通过 JavaScript 向服务器发出建立 WebSocket 连接的请求，  连接建立以后，客户端和服务器端就可以通过 TCP 连  接直接交换数据。 |
| **CSS（**界面优化）：通过CSS进行优化。 |
| **WebSocket API**:WebSocket是HTML5开始提供的一种在单个TCP连接 上进行全双工通讯的协议，在WebSocket API中，浏  览器和服务器只需要做一个握手的动作，然后，浏览  器和服务器之间就形成了一条快速通道。两者之间就  直接可以数据互相传送。在实现WebSocket连线过程  中浏览器发出WebSocket连线请求，然后服务器发出  响应，这个过程称为“握手”。 |
| **Node.js**（建设网站）： Node.js是一款web框架，用来建设网站。  简单的说 Node.js 就是运行在服务端的  JavaScript。Node.js 是一个基于Chrome  JavaScript 运行时建立的一个平台。Node.js  是一个事件驱动I/O服务端JavaScript环境，  基于Google的V8引擎，V8引擎执行  Javascript 的速度非常快，性能非常好。  Node.js就是让JavaScript(js)可以实现服务  器上的开发。JavaScript(前端语言)+node.js(JavaScript的运行环境)=实现服务器上的开发(后端技术)，也就是说js+node.js实现了后端开发的技术。 |

**C/S结构**：全称Client/Server,即客户机/服务器降低系统通讯开销。

**超级管理员**：对APP内交易的用户进行管理的人员，根据实际需要 对管理员赋予适当的权限。

**用户名ID**：用户输入的用于登录校园交易APP的身份标识。

**管理员身份验证**：根据管理员名和密码判断管理员是否为系统合法 管理员。

**数据有效性验证**：对用户输入的数据进行检查，判断是否合乎系统 要求。

## 1.4参考资料



1.朴灵. 深入浅出NodeJS[M]. 第1版. 人民邮电出版社出版社, 2013.

2.马尔奇·哈弗贝克. JavaScript编程精解[M]. 第2版. 机械工程出 版社, 2016.

1. ustbhuangyi. Vue.js 高仿饿了么外卖APP 收藏 [EB/OL]. [2018-3-30 3:30pm].

<https://coding.imooc.com/class/chapter/74.html#Anchor>.

4.计算分院老师. 数据库系统设计与开发[M]. 第1版. 浙江大学城市 学院, 2017.

5.上野宣. 图解HTTP[M]. 第1班. 人民邮电出版社, 2014.

6.网页： <https://wenku.baidu.com/view/afd60dccce2f0066f53322ed.html> [2018/4/21 9:00am].

# 总体设计

## 2.1需求规定



### 2.1.1系统功能

|  |  |
| --- | --- |
| 系统管理 | 管理员功能：对用户发出警告，从用户信息表中删除非法用户。这项功能只能为管理员所有。 |
| 用户管理 | 这个模块实现了用户请求交易，发布、更新交易信息的功能，并使用户能够更新自己的注册信息以及基本信息。 |
| 交易管理 | 在用户交易管理中，根据用户的提供的交易物品生成了“出售信息表”，用户可以随时对自己发布的信息进行更改或删除。在交易的过程中，系统会根据物品的出售信息目录供用户浏览参考。对于没有明显出售目标的用户，也可以浏览由出售信息表生成的出售物品目录 |
| 出售管理 | 用户发出出售请求后，系统会自动检查用户是否已登录。  用户录入出售信息（包括出售物品名称（gods），出售物品价格（sellprice），出售物品种类（category），出售物品描述（selldes），联系电话（contactp），联系QQ（contactq），联系微信（contactw））。其中物品名称、价格、种类为必填字段。联系方式为电话、微信或QQ，考虑到用户可能不愿意留下自己的电话、微信或QQ，我们将联系方式设为可选字段。当有交易请求时，系统通知用户。  经用户检查录入信息正确无误，生成出售物品信息单。  系统将出售信息纳入出售信息表，并根据出售信息表上的内容发布出售信息 |
| 买卖信息更新管理 | 用户可以随时更新自己的买卖信息。 |
| 搜索 | 用户录入搜索信息开始搜索，系统匹配后生成搜索结果。 |
| 用户信息管理 | 用户信息管理包括更改用户注册手机、更改用户密码、更改用户基本信息；交易管理包括出售管理这些管理操作，主要针对用户信息表进行修改，每个用户只有在登陆后才能修改自己的信息 |
| 更新注册手机 | 用户修改自己的注册手机信息时，必须输入原来的账户信息（注册手机和密码），经系统审核无误后，用户可以修改注册手机 |
| 更新用户密码 | 用户修改密码时，需要输入原密码，并两次输入新密码，经系统审核无误后，用户可以修改密码 |
| 更新用户基本信息 | 用户可以随时更新自己的买卖信息 |
| 用户注册 | 用户录入账户信息，进行有效性检查，生成用户信息表，返回登录界面进行登录 |

### 

### 2.1.2系统性能

#### 2.1.2.1精度

1. 相互合作的用户数量：目前是所服务的用户为浙江大学城

市学院里的12000名左右的学生。

1. 二手交易金额：整数最高6位，小数2位。
2. 浮点数据（输入、输出、传输）：小数点后5位。

（4）查询信息时应保证查全率，所有相应域包含查询关键字的

记录都应该查到。

（5）查询信息应保证查准率，查到的记录应与给定的查询条件

完全匹配。

（6） 主存容量：2G

磁盘容量：40G

安全性：阿里云服务器的安全性能

#### 2.1.2.2时间特性要求

在硬件和网络条件满足的前提下，所有日常性操作事务的平均响应时间应小于10秒，最长响应时间应小于30秒；对于查询性事务的平均响应时间应小于60秒，最长响应时间应小于180秒。

设计负载下：

注册用户：3500以上；

在线用户：1000以上；

响应时间：<1s；

信息量速率：1M带宽；

更新处理时间：<500ms；

数据转换和传送时间：<1000ms；

#### 2.1.2.3可靠性

对于编写好的软件，会进行大量的测试，不断地查找里面出现的bug，并及时的对其进行修改，尽可能的减少bug的数量。随着用户量的增加，我们会及时的更新我们的服务器和数据库，从而保证app使用的可靠性。避免用户量太大，而造成服务器瘫痪，影响app使用的可靠性。

系统一个月内不能出现三次以上故障

#### 2.1.2.4灵活性

本项目采用Vue.js+Node.js+Express+MySQL进行开发，可以跨平台运行且前后端分离，开发程序逻辑清晰

Express则是Node.js的一个框架

|  |  |
| --- | --- |
| Node.js优点： | 1. 采用事件驱动、异步编程，为网络服务而设计。其实Javascript的匿名函数和闭包特性非常适合事件驱动、异步编程。而且JavaScript也简单易学，很多前端设计人员可以很快上手做后端设计。   2、Node.js非阻塞模式的IO处理给Node.js带来在相对低系统资源耗用下的高性能与出众的负载能力，非常适合用作依赖其它IO资源的中间层服务。3、Node.js轻量高效，可以认为是数据密集型分布式部署环境下的实时应用系统的完美解决方案。Node非常适合如下情况：在响应客户端之前，您预计可能有很高的流量，但所需的服务器端逻辑和处理不一定很多。 |
| Vue.js优点： | 在现在单页应用这么火爆的年代，路由已经成为了我们开发应用必不可少的利器；而纵观各大框架，都会有对应的强大路由支持。Vue.js 因其性能、通用、易用、体积、学习成本低等特点已经成为了广大前端们的新宠，而其对应的路由 vue-router 也是设计的简单好用，功能强大。 |

### 2.1.3输入输出要求

本APP中数据存储采用MYSQL 数据库软件，数据表使用UTF-8编码存储；

图像文件可接受JPEG、GIF、PNG等常见格式。

用户登录和注册功能：

1.用户账号名为自定义创建，但会有格式的约束

2.用户密码为自定义创建，但会有格式的约束，暂定为必须由字母和数字组成且密码串长度为10-15个字符

3.在注册时会要求用户重复输入密码来核实密码，当两次密码不一致时创建账号失败

搜索功能：

搜索功能采用模糊搜索机制，用户只需输入所查商品的部分字符即可完成查询。

### 2.1.4数据管理能力

系统开发采用vue-router，mint-ui，vue-resource进行界面处理，并用Node.js进行数据库访问：

1、软件开发工具MYSQL +webstorm+visual stdio code

1. 使用MYSQL作为后台数据库

3、在Windows10操作系统平台上使用

设计数据库规模如下：

订单记录：>100,00条

各类信息记录：>100,00条

最大数据库单表规模：2 GB

### 2.1.5故障处理要求

弹出警告框

|  |  |  |
| --- | --- | --- |
| 错误名称 | 输出形式 | 处理方法 |
| 用户名或密码错误 | 弹窗提醒 | 重新输入正确的用户名密码 |
| 搜索目标不存在 | 弹窗提醒 | 没有搜到您想要的内容 |
| 服务器数据库操作出错 | 弹窗提醒服务器正忙 | 等待服务器修复 |
| 网络错误 | 弹窗提醒网络连接出错 | 等待网络连接 |

### 2.1.6其他专门要求

1. 安全设施要求：

暂无

1. 安全性要求：

用户登录时需输入账户密码，密码输入三次不正确，则需等待一分钟。

1. 设置权限：

管理员功能：对用户发出警告，从用户信息表中删除非法用户。这 项功能只能为管理员所有。

1. 对于敏感的数据，要存入数据库进行加密操作。
2. 对于数据库中的数据，一定要进行实时的备份操作。每当数据

库中发生一些修改操作时，就进行备份操作，这样的操作，在

后来的数据丢失时，就可以进行及时的恢复，从而避免了数据

丢失造成的损失。

1. 用户文档：

* 安装指南

纸质文档，16开本；

* 用户手册

纸质文档，16开本；

* 在线帮助
* 电子文档，与软件产品一同分发、配置；
* 使用教程电子文档，与软件产品一同分发、配置。

1. 软件质量属性：
2. 正确性：

要求发布的项目app达到用户的预期目标，运行时基本无错 误。

1. 安全性：

1. 因为数据库和服务器是部署在云端上的，诸如百度云，新浪云等等的云平台的安全性还是有保障的，所以不用担心服务器宕机等问题

2. 需要注意的是用户个人信息不能被泄露，云端的数据库不会被除了管理员之外的人或物篡改

1. 可靠性：

对于编写好的软件，会进行大量的测试，不断地查找里面出现的bug，并及时的对其进行修改，尽可能的减少bug的数量。随着用户量的增加，我们会及时的更新我们的服务器和数据库，从而保证app使用的可靠性。避免用户量太大，而造成服务器瘫痪，影响app使用的可靠性。

系统一个月内不能出现三次以上故障

1. 可用性：

最大并发操作数量8000

1. 效率：

对于浏览、查询、添加、删除、更新等一般操作，要求

及时响应，在2~3秒内。

1. 完整性：

要求能在发生意外的情况下，保证不丢失数据。

1. 易使用性：

对于app的主界面设计，我们采用原创性设计，在必要的情况下去参考一些成功的经验。组员研究rp的使用。

1. 可维护性：

当软件运行发生错误时，能够快速、准确对其定位、诊 断和修改恢复。

在设计app的时候，将每个模块都分别独立开来，对于一些页面，我们将其做成了模板。这样可以集中精力放在代码块的构造与实现上。避免了一些不必要的困扰。在代码设计过程中，尽可能的减少模块之间的耦合性。做到模块和模块之间的分离。这样，日后的维护具有较好的方便性。

1. 可测试性：

设计师尽可能减少测试本软件的各项功能所需的工作量。

1. 复用性：

设计时应采取模块化的方法进行设计，对系统内的各模块接口尽可能达到聚、低耦合的程度，提高各模块的复用性。

1. 可理解性：

对于本软件提供的各种菜单、按钮，其功能应该一目了然，易于理解。

1. 互联性：

要求提供数据得到如何导入和导出接口，以易于同其他系统连接。

1. 逆向需求

用户输入正确的身份信息，能够进行注册

不同用户之间的ID不能相同

1. 将来可能提出的要求

可能会添加支付接口

可能增添以下几个方面功能：

1. 课程表查询、成绩查询、公交车查询、校园通知、校园活动。

校园通知：重要事务通知，校历等。

校园活动：社团活动，校园晚会，讲座活动等。

1. 失物招领

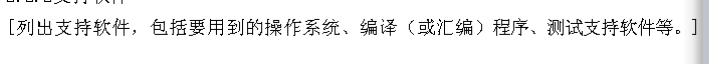
## 2.2运行环境

### 2.2.1设备



后台程序运行在云服务器上，数据库存储在云服务器上

### 2.2.2支持软件



操作系统：Windows10

软件开发环境：Visual Studio Code, MYSQL，WebStorm

办公软件：Microsoft Office 2016、Microsoft project 2016

界面设计：Axure RP

### 2.2.3接口



1. 用户接口需求：

页面简洁，突出功能模块重点

（2）硬件接口需求：

——服务器：云服务器ECS

处理器: 1核CPU，Intel Xeon E5-2682 v4

内存：2G内存，最新一代DDR4 内存硬盘：至少80G以上；

网络：1M带宽，VPC专有网络, I/O 优化；

网络适配器：10MB/100MB自适应；

磁盘大小：40G系统盘，高效云盘

——软、硬件之间交流的数据；

通过数据库来管理软、硬件之间交流的数据。

持久数据：存储在数据库、文件等。

感官数据：显示在界面上的数据、播放的声音等

内存数据：程序中的变量容纳的数据，存放在内存

——使用的通讯协议；

采取Websocket来实现双向通信（服务器端和客户端可以同 时发出请求），且Socket.IO支持以事件为基础的实时双向 通讯，它可以工作在任何平台、浏览器或移动设备

（3）软件接口需求：

操作系统：Windows10

软件开发环境：Android Studio，MYSQL，WebStorm

（4）通信接口需求：

——WEB浏览器：99%的浏览器都支持该协议

——网络通讯标准或者协议：采取Websocket来实现双向通信（服 务器端和客户端可以同时发出请求），且Socket.IO支持以事件 为基础的实时双向通讯，它可以工作在任何平台、浏览器或移动 设备。

——消息格式：采用UTF-8编码

——通讯安全或加密问题：采用md5加密方式

——数据传输速率：数据传输速度在：10MＢ~100MＢ

——同步和异步通讯机制：Ajax：异步JavaScript和XML，通过在后 台与服务器进行少量数据交换，Ajax可以使网页实现异步更新

——采用第三方IM通讯技术嵌入本应用，实现买卖双方的沟通交流

### 2.2.4控制

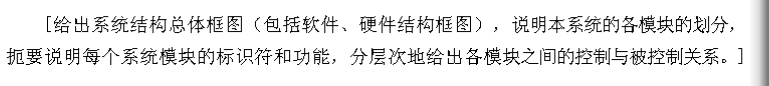


## 2.3基本设计概念和处理流程



数据流图（or系统流程图？？）

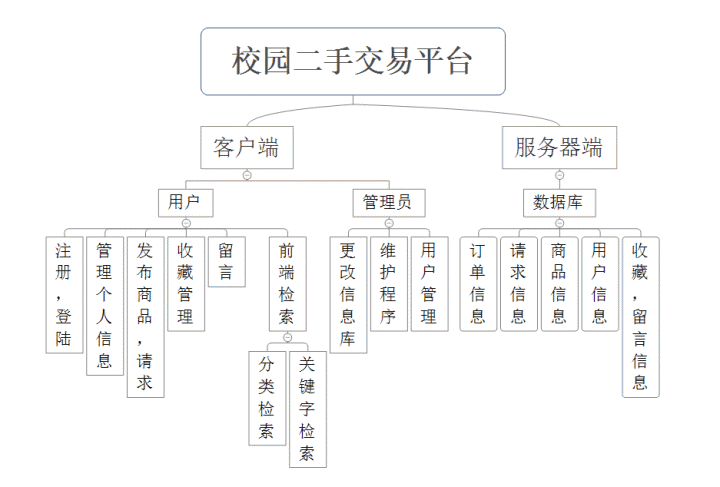
## 2.4结构



硬件结构图



软件结构图



## 2.5功能需求与系统模块的关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 用户管理模块 | 类别管理模块 | 商品管理模块 | 收藏管理模块 |
| 用户管理模块 | √ |  |  |  |
| 类别管理模块 |  | √ |  |  |
| 商品管理模块 |  |  | √ |  |
| 收藏管理模块 |  |  |  | √ |

## 2.6人工处理过程



暂无

## 2.7尚未解决的问题

无

# 接口设计

## 3.1用户接口

登录与注册接口：

系统提供图像接口。用户点击登录或注册按钮，系统会自动验证登录表单或注册表单的完整性和正确性。如果登录成功，系统将页面跳转到程序主页；如果登录失败，系统提示：帐号或密码错误。如果注册成功，系统将保存用户的注册信息，并将页面跳转到登录页面；如果注册失败，系统会给予错误信息，并停留在原页面。

商品评价接口：

系统提供图像接口。用户输入评价信息后点击评价按钮，系统会将评价信息保存在数据库，并发布在对应商品的评论区，使对应商品的评论量加一。

商品发布接口：

系统提供图像接口。用户填写商品信息后点击发布按钮，系统会验证商品信息的完整性。如果正确，系统会将商品信息提交到数据库，并通知用户发布成功；如果失败，系统将提示错误信息。

商品收藏接口：

系统提供图像接口。用户点击收藏按钮，系统会生成收藏信息并保存在数据库，对应的商品收藏量加一，系统将收藏信息发布在用户的收藏列表，提示用户收藏成功。

商品搜索接口：

系统提供图像接口。用户输入搜索信息后点击搜索按钮，系统会根据搜索信息在数据库里搜索相关的商品，并将结果返回到搜索页面。

提交订单接口：

系统提供图像接口。用户在商品主页点击购买按钮，系统会结合商品信息和订单人信息生成订单，并将订单信息保存在数据库，然后将订单信息分别发到卖家和买家的用户窗口

## 3.2外部接口

无

## 3.3内部接口

无

# 运行设计

## 4.1运行模块的组合

### 4.1.1用户功能模块组合

登录注册，用户个人信息管理，发布商品，买卖信息管理

### 4.1.2前端信息汇聚、检索功能模块组合

前端检索，分类查询，收藏，评论

### 4.1.3管理员功能模块组合

管理员管理

## 4.2运行控制

操作步骤：主程序运行，等待不同权限的用户进行操作请求，调用各个子模块。

## 4.3运行时间

各个运行模块的时间均控制在1-2秒内。系统采用消息驱动的方式，能提高计算机的利用率。

# 系统数据结构设计

## 5.1逻辑结构设计要点

### User表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User表 | | | | | |
| 字段名 | 实际字段名 | 类型 | 是否能为空 | 键型 | 说明 |
| 帐号 | user\_id | varchar | 不能 | 主键 |  |
| 密码 | user\_pw | varchar | 不能 |  |  |
| 姓名 | user\_name | varchar | 不能 |  |  |
| 性别 | user\_sex | varchar | 不能 |  | 可以设置为保密 |
| 生日 | user\_birth |  | 不能 |  |  |
| 手机号 | user\_phone | varchar | 不能 |  | 用于找回帐号，以及与交易对象进行联系 |
| 邮箱地址 | user\_email | varchar | 不能 |  | 用于找回帐号，以及与交易对象进行联系 |
| 拥有积分 | user\_score | int | 不能 |  | 上架商品需要积分 |
| 学号 | user\_sid | varchar |  |  | 未知 |
| 个人简述 | user\_resume | varchar |  |  | 简单的表述自己 |
| 照片 | user\_picture | varchar |  |  | 数据库只保存图片的地址，图片保存在硬盘里 |

### Commodity表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| commodity | | | | | |
| 商品编号 | commodity\_id | varchar | 不能 | 主键 |  |
| 商品名称 | commodity\_name | varchar | 不能 |  |  |
| 商品图片 | commodity\_picture | varchar | 不能 |  | 数据库只保存图片的地址，图片保存在硬盘里 |
| 商品价格 | commodity\_price | int | 不能 |  | 该价格只做参考，交易双方可私下协商 |
| 上架者帐号 | user\_id | varchar | 不能 | 外键 |  |
| 商品介绍 | commodity\_intro | varchar | 不能 |  |  |
| 商品关键词 | commodity\_key | varchar | 不能 |  | 关键词用于方便商品的查找 |
| 商品收藏量 | commodity\_scl | int | 不能 |  | 商品被用户收藏的数量，该数量作为放在主页的依据之一 |
| 商品评论量 | commodity\_pll | int | 不能 |  | 评论的数量，该数量作为放在主页的依据之一 |
| 商品上架时间 | commodity\_uptime | datetime | 不能 |  | 其他条件相同是，先上架的物品放在主页 |
| 商品上架积分 | commodity\_score | int | 不能 |  | 商品上架积分所需要的积分 |

### Order表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order | | | | | |
| 订单编号 | order\_id | varchar | 不能 | 主键 |  |
| 买家帐号 | user\_id1 | varchar | 不能 | 外键 |  |
| 卖家帐号 | user\_id2 | varchar | 不能 | 外键 |  |
| 交易地址 | order\_address | varchar | 不能 |  | 双方面对面交易的地址 |
| 买家手机号 | order\_tel1 | varchar | 不能 |  | 也可以是微信等联系方式 |
| 卖家手机号 | order\_tel2 | varchar | 不能 |  | 也可以是微信等联系方式 |
| 下单时间 | order\_time | datetime | 不能 |  | 买家在APP上购买的时间 |
| 交易额 | order\_cost | double | 不能 |  |  |
| 交易花费积分 | commodity\_score | int | 不能 |  |  |
| 商品编号 | commodity\_id | varchar | 不能 | 外键 |  |

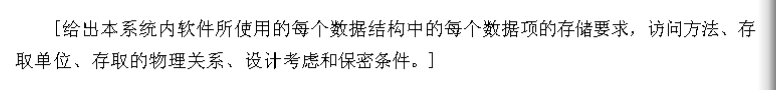
### Message表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Message | | | | | |
| 留言编号 | pl\_id | varchar | 不能 | 主键 |  |
| 留言时间 | pl\_time | datetime | 不能 |  |  |
| 留言内容 | pl\_content | varchar | 不能 |  | 内容上限为300个字符 |
| 商品编号 | commodity\_id | varchar | 不能 | 外键 |  |
| 发布者编号 | user\_id | varchar | 不能 | 外键 |  |

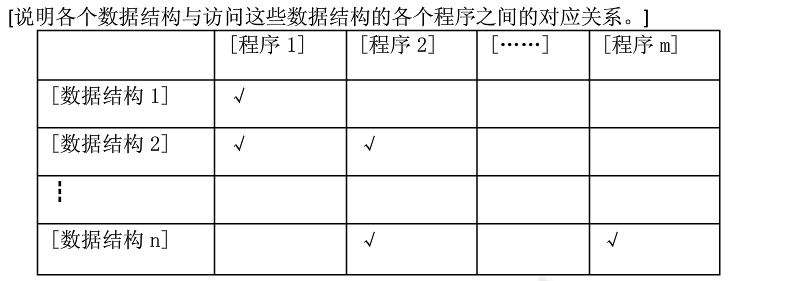
### Collection表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Collection | | | | | |
| 收藏编号 | sc\_id | varchar | 不能 | 主键 |  |
| 收藏时间 | sc\_time | datetime | 不能 |  |  |
| 收藏商品的价格 | sc\_price | varchar | 不能 |  |  |
| 商品编号 | commodity\_id | varchar | 不能 | 外键 |  |
| 收藏者编号 | user\_id | varchar | 不能 | 外键 |  |

## 5.2物理结构设计要点



## 5.3数据结构与程序的关系



# 系统出错处理设计

## 6.1出错输出信息

弹出警告框

|  |  |  |
| --- | --- | --- |
| 错误名称 | 输出形式 | 处理方法 |
| 用户名或密码错误 | 弹窗提醒 | 重新输入正确的用户名密码 |
| 搜索目标不存在 | 弹窗提醒 | 没有搜到您想要的内容 |
| 服务器数据库操作出错 | 弹窗提醒服务器正忙 | 等待服务器修复 |
| 网络错误 | 弹窗提醒网络连接出错 | 等待网络连接 |

## 6.2出错补救措施

若出现上述错误，应及时对系统数据库进行备份还原，处理对应的错误信息，及时让系统回复运行。

## 6.3系统维护设计

系统维护工作贯穿于系统的整个运行过程中，包括：系统应用程序的维护、数据的维护、代码的维护、硬件设备维护，系统维护的重点是系统应用软件的维护工作。而系统维护工作不应总是被动的等待用户提出要求后才进行，应进行主动的预防性维护。系统在运行时，也要随着环境的变化根据不同的需求及变化对系统进行必要的修改，使得系统功能更加完善。

1. [↑](#footnote-ref-0)