**校叮当校园app项目用户手册**



|  |  |  |
| --- | --- | --- |
| 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用户手册 |
| 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 | 完成用户手册初稿，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 | 检错细化 |

**目录**

[一、 引言 4](#_Toc9103)

[1.1编写目的 4](#_Toc8321)

[1.2项目背景 4](#_Toc8685)

[1.3定义 5](#_Toc18260)

[1.4参考资料 7](#_Toc25295)

[二、 用途 8](#_Toc4924)

[2.1功能 8](#_Toc18313)

[2.2性能 10](#_Toc12111)

[2.2.1精度 10](#_Toc10667)

[2.2.2事件特性 11](#_Toc9860)

[2.2.3灵活性 11](#_Toc10592)

[2.3安全保密 13](#_Toc15622)

[三、 运行环境 14](#_Toc21527)

[3.1硬设备 14](#_Toc25805)

[3.2支持软件 14](#_Toc3358)

[3.3数据结构 15](#_Toc12351)

[3.3.1 User表 15](#_Toc12368)

[3.3.2 Commodity表 15](#_Toc31407)

[3.3.3 Order表 16](#_Toc26381)

[3.3.4 Message表 17](#_Toc32237)

[3.3.5 Collection表 17](#_Toc10770)

[四、 使用过程 18](#_Toc1351)

[4.1安装与初始化 25](#_Toc32489)

[4.2输入 25](#_Toc12719)

[4.2.1输入数据的现实背景 26](#_Toc13836)

[4.2.2输入格式 26](#_Toc8155)

[4.2.3输入举例 27](#_Toc22256)

[4.3输出 27](#_Toc2202)

[4.3.1输出数据的现实背景 27](#_Toc4497)

[4.3.2输出格式 28](#_Toc29884)

[4.3.3输出举例 28](#_Toc2360)

[4.4文件查询 28](#_Toc10552)

[4.5出错处理和恢复 29](#_Toc10896)

[4.6终端操作 29](#_Toc21873)

# 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二手交易平台项目。

（2）项目的委托单位

G15小组

（3）项目的用户：

浙江大学城市学院的在校大学生。

（4）项目的任务提出者

G15小组

（5）项目的主要承担部门

G15小组

（6）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。

（7）实现

开发与测试基于Visual Stdio Code集成开发软件进行，用Axure RP进行UI的制作，后端用Node.js开发，数据库采用MYSQL，前端采用Vue.js框架 ，数据库和服务器均架设在云端

（8）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。

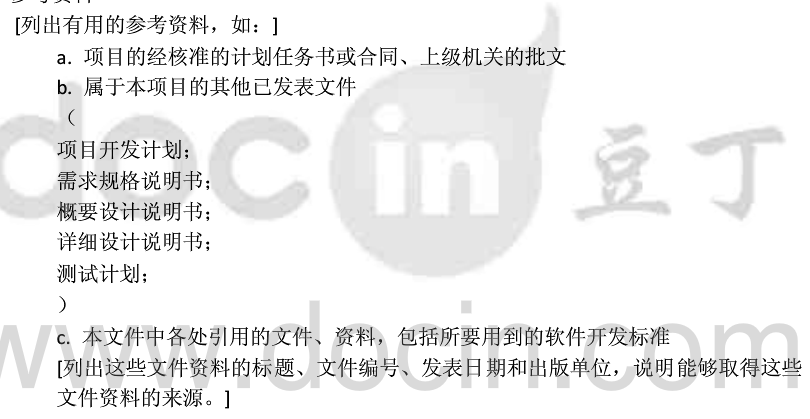
**超级管理员**：对APP内交易的用户进行管理的人员，根据实际需要 对管理员赋予适当的权限。

**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： 模板：<http://www.docin.com/p-1322536948.html>

[2018/5/5 9:00am].

# 用途

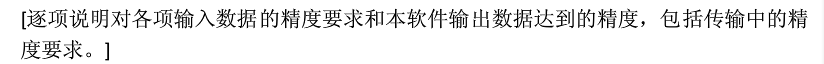
## 2.1功能



|  |  |
| --- | --- |
| 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 |

## 2.2性能

### 2.2.1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

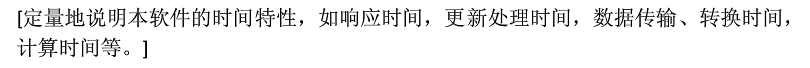
完全匹配。

（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

### 2.2.2事件特性



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；

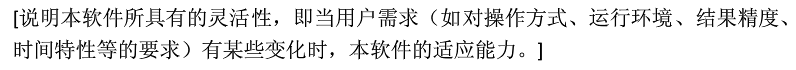
响应时间：<1s；

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

### 2.2.3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。 |

## 2.3安全保密



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. 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

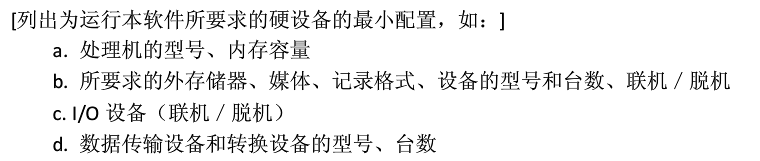
机等问题

1. 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

# 运行环境

## 3.1硬设备



服务器：云服务器ECS

处理器: 1核CPU，Intel Xeon E5-2682 v4

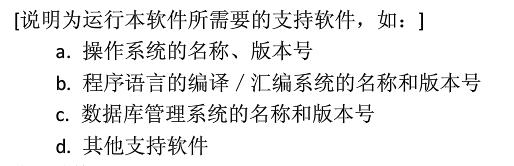
内存：2G内存，最新一代DDR4 内存硬盘：至少80G以上；

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

## 3.2支持软件



1. 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：
2. 软件开发工具MYSQL +webstorm+visual stdio code
3. 使用MYSQL作为后台数据库
4. 在Windows10操作系统平台上使用

## 3.3数据结构



### 3.3.1 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 |

### 3.3.2 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 |

### 3.3.3 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 |  |

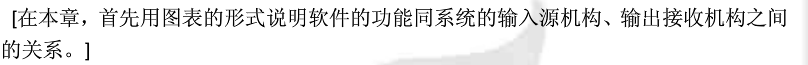
### 3.3.4 Message表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |  |

### 3.3.5 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 |  |

# 使用过程



起始界面（普通用户登录界面）

用户进入APP后第一眼看到的界面，已经注册完毕的普通用户可通过输入用户名和密码来登陆。



注册界面：

若还没有账户，则点击“注册账户”按钮，及跳转到注册界面，通过输入正确的信息，进行账号注册。



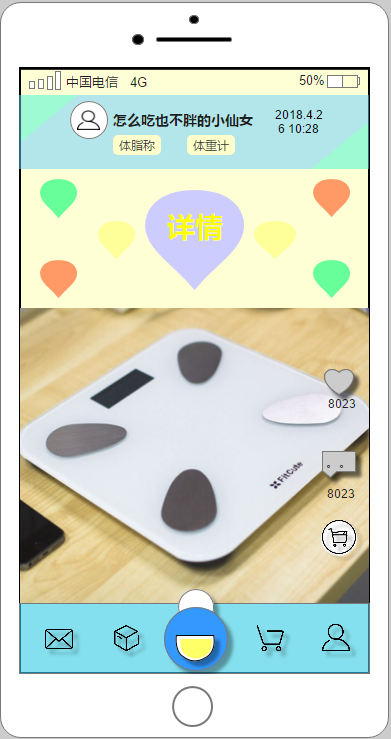
管理员登陆界面：

若是管理员登陆，则点击登陆界面的“管理员登陆”按钮，及跳转到管理员登陆界面，填入正确的账号秘密后，及可登陆。



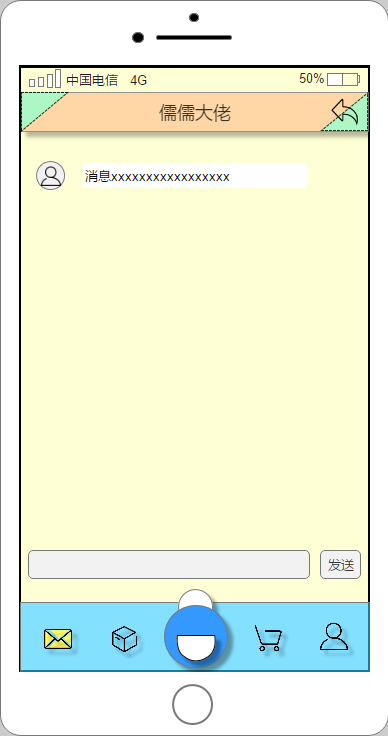
商品浏览界面：

普通用户登陆完成后，及跳转到浏览界面，可上下滑动，随机显示商品信息，用户及拥有了查看详情、收藏、留言、购买的权利。

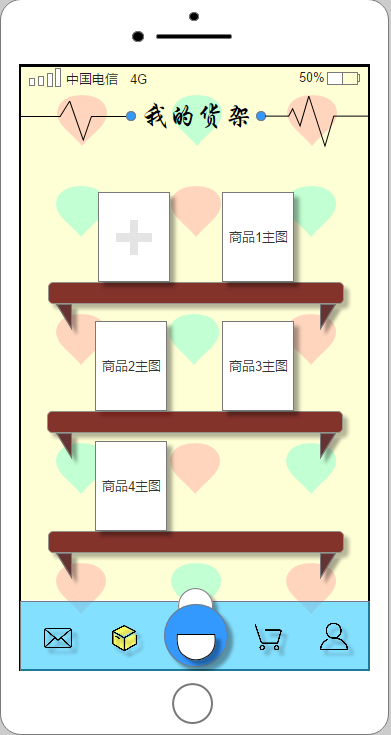
消息通知界面：

买卖双方的沟通都在消息通知中。



我的货架界面：

显示我发布的商品，并可以新增、删除、修改发布。

我的订单界面：

分别显示，我卖出的和我买入的商品订单，可进行删除操作。

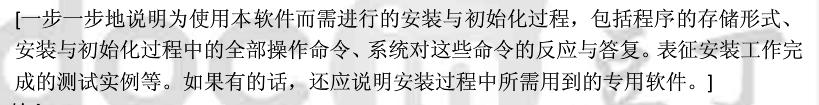


我的信息界面：

显示用户的基本信息，可自行修改。



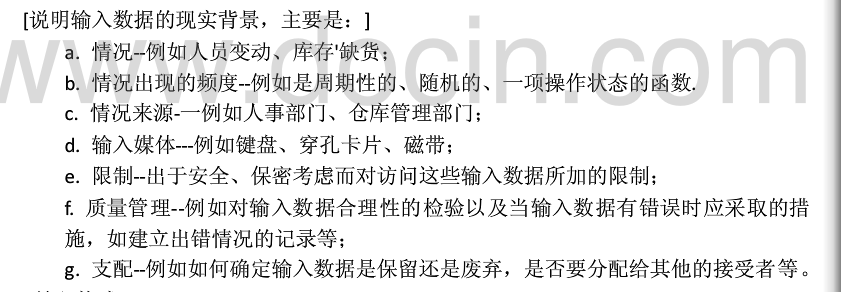
## 4.1安装与初始化



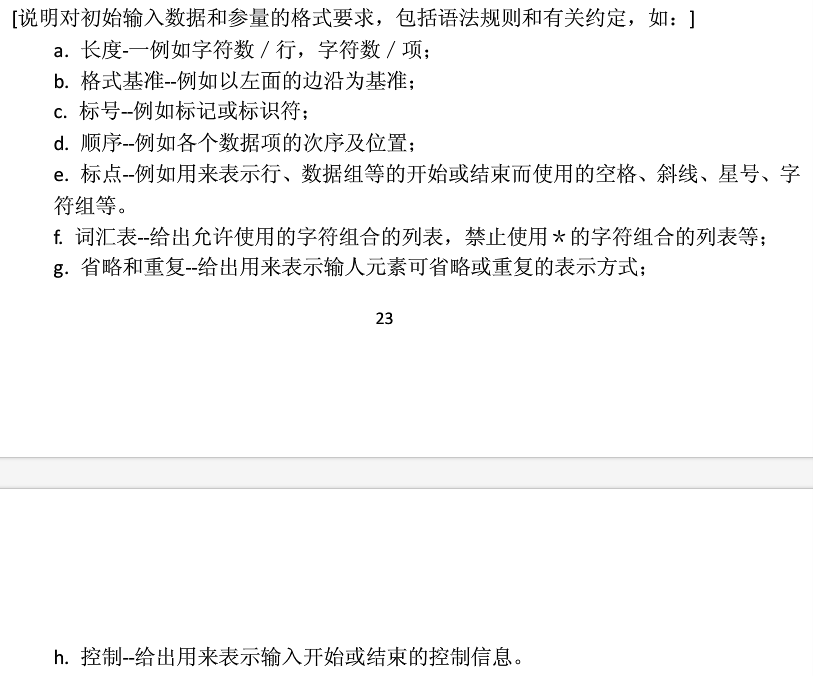
## 4.2输入



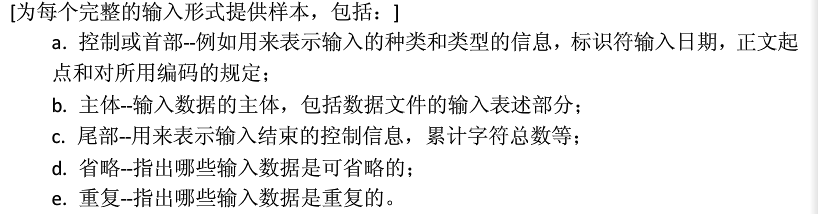
### 4.2.1输入数据的现实背景



### 4.2.2输入格式



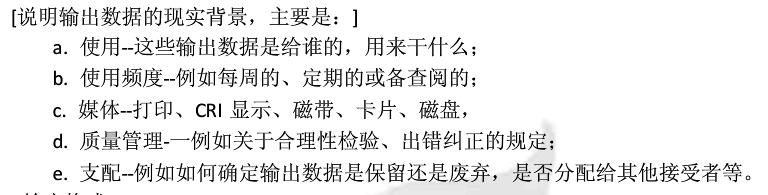
### 4.2.3输入举例



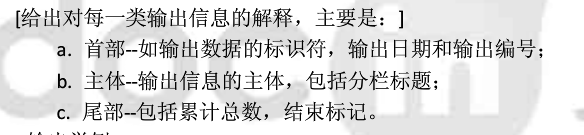
## 4.3输出



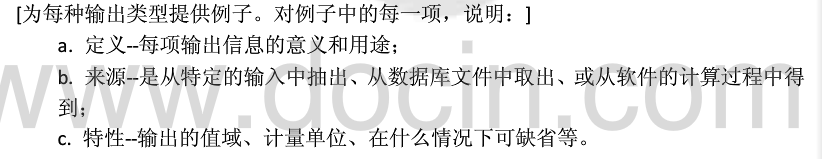
### 4.3.1输出数据的现实背景



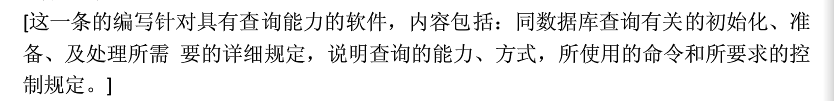
### 4.3.2输出格式



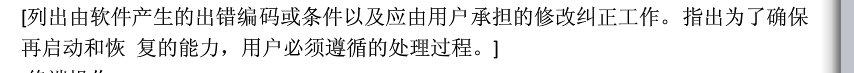
### 4.3.3输出举例



## 4.4文件查询



## 4.5出错处理和恢复



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 |

## 4.6终端操作

