## 详细设计说明书（规范）

[详细设计说明书（规范） 1](#_Toc280644296)

[1 引言 5](#_Toc280644297)

[1.1. 编写目的 5](#_Toc280644298)

[1.2. 背景 5](#_Toc280644299)

[1.3. 定义 5](#_Toc280644300)

[1.4. 参考资料 5](#_Toc280644301)

[2 程序系统的结构 6](#_Toc280644302)

[3　程序1用户登录（userLogin）设计说明 6](#_Toc280644303)

[3． 5输出项 8](#_Toc280644304)

[3．6算法 9](#_Toc280644305)

[3．7流程逻辑 10](#_Toc280644306)

[3．8接口 10](#_Toc280644307)

[3．9存储分配 11](#_Toc280644308)

[3．11限制条件 12](#_Toc280644309)

[3．13尚未解决的问题 12](#_Toc280644310)

[4 程序1 查询用户（userQuery）设计说明 12](#_Toc280644311)

[4．1程序描述 12](#_Toc280644312)

[4．2功能 13](#_Toc280644313)

[4．3性能 13](#_Toc280644314)

[4．4输入项 13](#_Toc280644315)

[4．5输出项 14](#_Toc280644316)

[4．6算法 14](#_Toc280644317)

[4.7 流程逻辑 15](#_Toc280644318)

[4．8顺序图 16](#_Toc280644319)

[4．9接口 16](#_Toc280644320)

[4．10存储分配 17](#_Toc280644321)

[4．11注释设计 17](#_Toc280644322)

[4．12限制条件 17](#_Toc280644323)

[4．13测试计划 17](#_Toc280644324)

[4．14尚未解决的问题 18](#_Toc280644325)

[5 程序3增加注册用户（userRegister）设计说明 18](#_Toc280644326)

[5．1程序描述 18](#_Toc280644327)

[5．2功能 18](#_Toc280644328)

[5．3性能 18](#_Toc280644329)

[5．4输入项 19](#_Toc280644330)

[5．5输出项 19](#_Toc280644331)

[5．6算法 20](#_Toc280644332)

[**5．7流程逻辑** 20](#_Toc280644333)

[5．8接口 21](#_Toc280644334)

[5．9存储分配 22](#_Toc280644335)

[5．10注释设计 22](#_Toc280644336)

[5．11限制条件 22](#_Toc280644337)

[5．12测试计划 22](#_Toc280644338)

[5．13尚未解决的问题 23](#_Toc280644339)

[6 程序5 用户信息修改（userEdit）设计说明 23](#_Toc280644340)

[6．1程序描述 23](#_Toc280644341)

[6．2功能 23](#_Toc280644342)

[6．3性能 24](#_Toc280644343)

[6．4输入项 24](#_Toc280644344)

[6．5输出项 25](#_Toc280644345)

[6．6算法 25](#_Toc280644346)

[6．8顺序图 26](#_Toc280644347)

[6．9接口 27](#_Toc280644348)

[6．10存储分配 27](#_Toc280644349)

[6．11注释设计 28](#_Toc280644350)

[6．12限制条件 28](#_Toc280644351)

[6．13测试计划 28](#_Toc280644352)

[6．14尚未解决的问题 29](#_Toc280644353)

[7 程序5 用户删除（deleteUser）设计说明 29](#_Toc280644354)

[7．1程序描述 29](#_Toc280644355)

[7．2功能 29](#_Toc280644356)

[7．3性能 29](#_Toc280644357)

[7．4输入项 30](#_Toc280644358)

[7．5输出项 30](#_Toc280644359)

[7．6算法 30](#_Toc280644360)

[7．7流程逻辑 31](#_Toc280644361)

[7．8接口 32](#_Toc280644362)

[7．9存储分配 32](#_Toc280644363)

[7．10注释设计 32](#_Toc280644364)

[7．11限制条件 33](#_Toc280644365)

[7．12尚未解决的问题 33](#_Toc280644366)

[8 程序6 商品查询（findProudct） 设计说明 33](#_Toc280644367)

[8．1程序描述 33](#_Toc280644368)

[8．2功能 33](#_Toc280644369)

[8．3性能 34](#_Toc280644370)

[8．4输入项 34](#_Toc280644371)

[8．5输出项 34](#_Toc280644372)

[8．6算法 34](#_Toc280644373)

[8.7 流程逻辑 35](#_Toc280644374)

[8．8顺序图 36](#_Toc280644375)

[8．9接口 36](#_Toc280644376)

[8．10存储分配 37](#_Toc280644377)

[8．11注释设计 37](#_Toc280644378)

[8．12限制条件 37](#_Toc280644379)

[8．13测试计划 37](#_Toc280644380)

[9 程序7 商品发布（addProduct）设计说明 38](#_Toc280644381)

[9．1程序描述 38](#_Toc280644382)

[9．2功能 38](#_Toc280644383)

[9．3性能 38](#_Toc280644384)

[9．4输入项 39](#_Toc280644385)

[9．5输出项 39](#_Toc280644386)

[9．6算法 40](#_Toc280644387)

[9.7 流程逻辑 40](#_Toc280644388)

[9．8顺序图 41](#_Toc280644389)

[9．9接口 41](#_Toc280644390)

[9．10存储分配 42](#_Toc280644391)

[9．11注释设计 42](#_Toc280644392)

[9．12限制条件 42](#_Toc280644393)

[9．13测试计划 42](#_Toc280644394)

[9．14尚未解决的问题 43](#_Toc280644395)

[10 程序10 商品修改(updateproduct) 设计说明 43](#_Toc280644396)

[10．1程序描述 43](#_Toc280644397)

[10．2功能 43](#_Toc280644398)

[10．3性能 43](#_Toc280644399)

[10．4输入项 43](#_Toc280644400)

[10．5输出项 44](#_Toc280644401)

[10．6算法 44](#_Toc280644402)

[10.7 流程逻辑 45](#_Toc280644403)

[10．8顺序图 46](#_Toc280644404)

[10．9接口 47](#_Toc280644405)

[10．10存储分配 47](#_Toc280644406)

[10．11注释设计 47](#_Toc280644407)

[10．12限制条件 48](#_Toc280644408)

[10．13测试计划 48](#_Toc280644409)

[11 程序10 商品删除(deleteproduct) 设计说明 49](#_Toc280644410)

[11．1程序描述 49](#_Toc280644411)

[11．2功能 49](#_Toc280644412)

[11．3性能 49](#_Toc280644413)

[11．4输入项 49](#_Toc280644414)

[11．5输出项 49](#_Toc280644415)

[11．6算法 50](#_Toc280644416)

[11.7 流程逻辑 50](#_Toc280644417)

[11．8顺序图 51](#_Toc280644418)

[11．9接口 52](#_Toc280644419)

[11．10存储分配 52](#_Toc280644420)

[11．11注释设计 52](#_Toc280644421)

[11．12限制条件 53](#_Toc280644422)

[11．13测试计划 53](#_Toc280644423)

[12 程序10 下订单（addOrder）设计说明 53](#_Toc280644424)

[12．1程序描述 53](#_Toc280644425)

[12．2功能 53](#_Toc280644426)

[12．3性能 54](#_Toc280644427)

[12．4输入项 54](#_Toc280644428)

[12．5输出项 54](#_Toc280644429)

[12．6算法 54](#_Toc280644430)

[12．7流程逻辑 54](#_Toc280644431)

[12．8接口 56](#_Toc280644432)

[12.9存储分配 56](#_Toc280644433)

[12．10注释设计 56](#_Toc280644434)

[12．11限制条件 57](#_Toc280644435)

[12．12测试计划 57](#_Toc280644436)

[12．13尚未解决的问题 57](#_Toc280644437)

[13 程序11 订单查询（findOrder）设计说明 57](#_Toc280644438)

[13．1程序描述 57](#_Toc280644439)

[13．2功能 57](#_Toc280644440)

[用户查询自己以前所下过的订单信息 57](#_Toc280644441)

[13．3性能 57](#_Toc280644442)

[13．4输入项 57](#_Toc280644443)

[13．5输出项 58](#_Toc280644444)

[13．6算法 58](#_Toc280644445)

[13．7流程逻辑 58](#_Toc280644446)

[13．8接口 60](#_Toc280644447)

[13．9存储分配 60](#_Toc280644448)

[13．10注释设计 60](#_Toc280644449)

[13．11限制条件 61](#_Toc280644450)

[13．12测试计划 61](#_Toc280644451)

[13．13尚未解决的问题 61](#_Toc280644452)

[14 程序12 修改订单（changeOrder）设计说明 61](#_Toc280644453)

[14．1程序描述 61](#_Toc280644454)

[14．2功能 61](#_Toc280644455)

[14．3性能 61](#_Toc280644456)

[14．4输入项 61](#_Toc280644457)

[14．5输出项 62](#_Toc280644458)

[14．6算法 62](#_Toc280644459)

[14．7流程逻辑 62](#_Toc280644460)

[62](#_Toc280644461)

[14．8接口 63](#_Toc280644462)

[14．11限制条件 64](#_Toc280644463)

[14．12测试计划 64](#_Toc280644464)

[14．13尚未解决的问题 64](#_Toc280644465)

[15 程序13 删除订单（deleteOrder）设计说明 64](#_Toc280644466)

[15．1程序描述 64](#_Toc280644467)

[15．2功能 64](#_Toc280644468)

[15．3性能 64](#_Toc280644469)

[15．4输入项 64](#_Toc280644470)

[15．5输出项 65](#_Toc280644471)

[15．6算法 65](#_Toc280644472)

[15．7流程逻辑 65](#_Toc280644473)

[15．8接口 66](#_Toc280644474)

[15．9存储分配 66](#_Toc280644475)

[15．10注释设计 66](#_Toc280644476)

[15．11限制条件 67](#_Toc280644477)

[15．12测试计划 67](#_Toc280644478)

[15．13尚未解决的问题 67](#_Toc280644479)

# 1 引言

## 1.1. 编写目的

在完成了系统的可行性研究和需求分析的基础上，为了明确软件需求、安排项目规划与进度、组织软件开发与测试、项目小组在考虑了几种可能的解决方案，并与程序员进行了较为深入地探讨和分析之后，提出了这概要设计说明书。此概要设计说明书对EC-校园二手商品交易系统的解决方案、功能分配、模块化分、程序的总体结构、输入输出和接口设计、运行设计及出错设计等方面作了全面的概括性的说明，为该软件的详细设计奠定了基础。

本概要设计适用于用户、项目管理人员、系统分析人员、程序设计人员以及系统测试人员阅读和参考，评选时也可作为组委会参考的依据。

## 1.2. 背景

|  |  |
| --- | --- |
| 背景 | 解释 |
| 软件系统 | EC-校园二手商品交易系统 |
| 任务提出者 | 软件工程导论老师廖竞 |
| 开发者 | 二手交易系统网站开发小组（赵芹、何均、郑弦、唐帅、刘训明） |
| 用户 | 绵阳市各高校师生 |

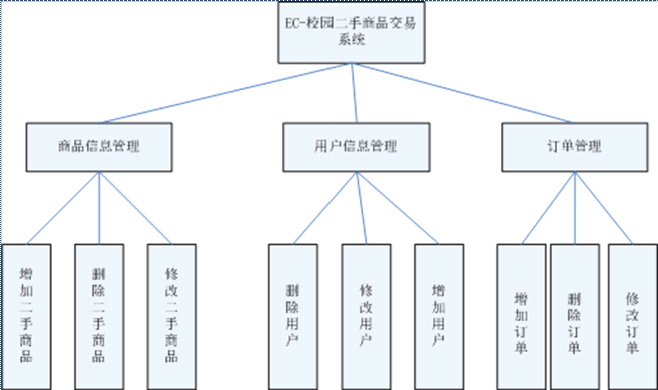
## 1.3. 定义

|  |  |  |
| --- | --- | --- |
| **序号** | **术语名称** | **术语定义** |
| 1 | SH | SecondHandShopt简称，即校园二手交易系统 |
| 2 | 数据结构 | 数据结构包括：数据库表的结构、其他数据结构等 |
| 3 | MySQL | 系统服务器所使用的数据库管理系统(DBMS) |
| 4 | 参考资料 | 指本文件书写时用到的其他资料。如各种有关规范、模板、标准、准则 |
| 5 | 商品 | 本文档中的商品均指二手商品 |

## 1.4. 参考资料

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 书籍名 | 主编 | 出版社 |
| 1 | 软件工程导论 | 张海藩 | 清华大学出版社 |
| 2 | UML系统分析与设计开发教程 | 冀振燕 | 人民邮电出版社 |
| 3 | 需求分析规格说明书 | 何均 |  |

# 2 程序系统的结构



|  |  |  |
| --- | --- | --- |
| **序号** | **模块程序名称** | **标识符** |
| 1 | 用户登录 | userLogin |
| 2 | 查询用户 | userQuery |
| 3 | 增加注册用户 | userRegister |
| 4 | 用户信息修改 | userEdit |
| 5 | 用户删除 | deleteUser |
| 6 | 商品查询 | findProudct |
| 7 | 商品发布 | addProduct |
| 8 | 商品修改 | Updateproduct |
| 9 | 商品删除 | deleteproduct |
| 10 | 下订单 | addOrder |
| 11 | 订单查询 | findOrder |
| 12 | 修改订单 | changeOrder |
| 13 | 删除订单 | deleteOrder |

3　程序1用户登录（userLogin）设计说明   
　　3．1程序描述

本模块包括会员用户登录系统、非会员用户的注册登录，密码验证，修改用户个人信息等功能。其中客户进入系统主界面后，如果是已经注册的会员用户，可以登录进入自己的主页，以便进行下一步商品选购等操作；如果是非注册的用户，可以马上注册会员并进入主页完善个人信息最后进行会员可进行的操作。本程序模块属于登录独立模块，设计该模块的目的在于让用户可以进入自己的会员界面，方便进行各种选购商品、查询商品等操作，也是便于系统对各用户的管理。本模块是系统的入口，以下所有操作都是在本模块的基础上完成的。  
　　3．2功能

实现用户的注册、登录、修改密码、完善客户个人信息的功能。下图此模块的IPO图：

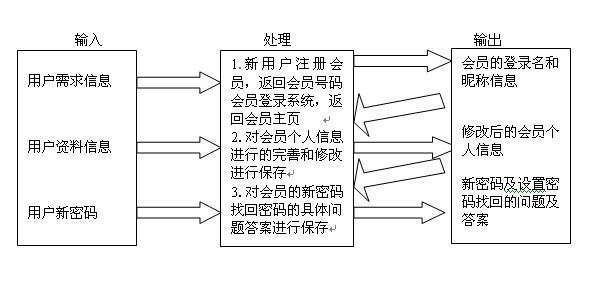


图3-1 用户登录IPO  
3．3性能  
　　程序要求有较好的容错能力和灵活性，能够在用户能接受的时间内响应用户的请求。

对用户输入的违规的用户名（用户名只能用英文字母、数字、下划线“\_”组成），能拒绝响应并提示用户名无效；

对注册新会员输入的用户名，能检测出是否已经存在，若存在相同名字，则拒绝响应并给出提示“该用户名已存在”；

对用户的密码要求在8位到16位之间，必须为数字、英文字母（有大小写之分）和下划线组成；对用户设定的密码给出密码的等级（包括 保密低等、中等保密、高级保密 三个级别）；

对用户密码找回操作，给出十个以上可选的问题。  
3．4输人项

用户需求信息（Require\_Info），即用户登录或者注册的命令，根据用户的需求给出相应的回应。

用户资料信息（User\_Info），用户通过给出自己需要完善或者修改的信息，确认修改或保存后即可生效。

用户新密码（User\_Keyword），为了对用户的密码进行更好的保密工作，用户应定期对密码进行修改，以防止密码被盗等，可以设置密码找回，设置问题及答案。

表 3-1 用户登录输入项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 用户名称 | UserName | String | 10位 | 键盘输入 | 1 | 一般 | 用户输入 | 无 |
| 用户密码 | User\_Keyword | String | 16位以内 | 键盘输入 | 1 | 一般 | 用户已设密码 | 无 |
| 用户信息资料 | User\_Info | String | 系统规定合法的信息，某些项有固定选择范围，如出生年月城市等 | 键盘输入 | 大于等于0 | 低 | 用户输入 | 无 |
| 验证码 | IdentificationCode | String | 4位 | 键盘输入 | 1 | 较高 | 系统随机给出 | 无 |

## 3． 5输出项

表 3-2 用户登录输出项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **数量** | **频度** | **输出说明** | **安全保密条件** |
| 新注册会员名称 | NewUserName | 字符串型 | 符合条件的未注册过的用户名称 | 详细列出 | 1 | 一般 | 新用户注册成功后显示有效的用户登录名 | 无 |
| 用户详细个人信息 | UserInfo | 字符串型 | 系统检测合法的用户个人信息 | 逐项列出 | >=0 | 较高 | 输出用户完善或者修改后保存的个人全部信息 | 无 |
| 密码 | UserKeyWord | 字符串型 | 密码为系统认可的8到16位有效字符 | 字符串形式 | 1 | 较高 | 输出新用户刚设置的密码或者老会员刚修改的密码 | 无 |
| 密码找回问题及答案 | FindKeyword | 字符串型 | 系统提供选择的用户选中的问题及用户输入的答案 | 分项列出 | >=1 | 一般 | 为防止密码被盗或丢失后找回密码而增加该功能 |  |

3．6算法   
　 此模块包括会员用户登录系统、非会员用户的注册登录，密码验证，修改用户个人信息，用户登录和注册都可以单独设置功能按钮，设置多个Text文本框来存放信息。通过用户点击注册、登录、修改资料Button进入各种操作界面。

本模块中主要是完成简单的存储用户信息，算法简单，只是对用户输入信息进行合法检测时得逐个检测。对用户登录程序的算法如下：

（1）首先程序要获取用户登录名、登录密码和验证码；

（2）将三个参数传入函数UserLogin（）

（3）对获取的用户名进行合法检查，看是否是已注册的合法用户及对密码和验证码的正确性进行判断；

BOOL UserLogin（string UserName, string UserKeyword, string IdentificationCode ）

{

CheckUserName(UserName);

CheckUserKeyword(UserKeyword);

CheckIdentificationCode(IdentificationCode);

EnterInterface（UserName）；

}

（4）系统检测信息合法后就进入用户个人主页；

## 3．7流程逻辑

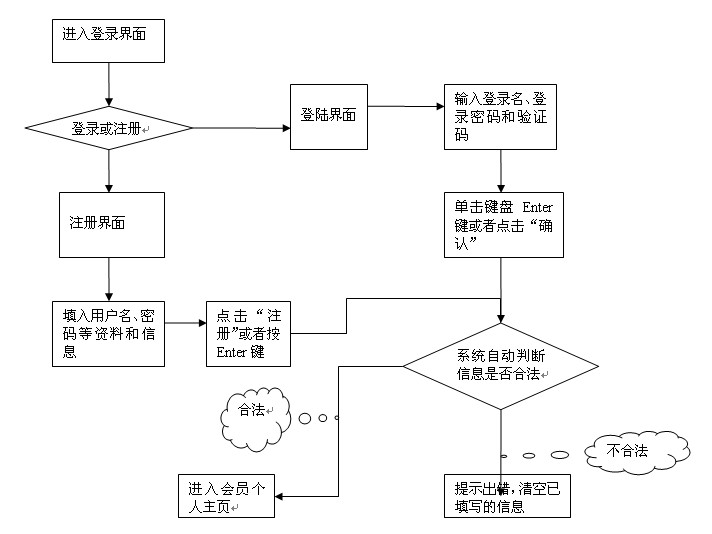


图 3-2 用户登录流程图逻辑

## 3．8接口

## 3．9存储分配

本程序存储主要是对用户的登录名、登录密码、密码找回及各种个人资料进行存储，主要存在数据库中。  
　　   
3．10注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\* 创建标识：（作者姓名+8位日期）

\* 修改标识：（作者姓名+8位日期）

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各分枝点处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

c．对各变量的功能、范围、缺省条件等所加的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*变量名称：

\*变量功能：

\*取值范围：

\*缺省值：

\*注释：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

3．11限制条件   
　　所有用户都可以进入登录及注册界面，但只有注册成功的会员才能进入自己的主页，并对自己的信息及密码等具有修改权限。此模块是整个系统的基础，只有进入该模块才能进行相关操作。  
3．12测试计划   
　　测试时主要是对用户输入的用户名、密码、验证码等信息的合法信进行检测，看系统是否能把不合法的信息检测出来。

（1）输入测试数据：输入非法的用户名登录系统（即不存在的用户名）。

预期结果：给出提示“您输入的用户名不存在，请重新输入”。

（2）输入测试数据：输入正确的用户名，但密码或者验证码错误。

预期结果：给出提示“您输入的密码错误，重新输入”或者“验证码错误”。

（3）输入测试数据：输入不合法的用户名或密码注册会员。

预期结果：给出提示“该用户已注册”或者“用户名只能使只能用英文字母、数字、下划线‘\_’”或者“用户密码必须为数字、英文字母（有大小写之分）和下划线‘\_’”。

（4）输入测试数据：输入正确合法的用户名、密码和验证码。

预期结果：成功进入该用户的个人主页界面。

3．13尚未解决的问题  
　　无。

# 4 程序1 查询用户（userQuery）设计说明

## 4．1程序描述

本模块只包括查询用户信息及更新。在用户信息查看中，管理员可以查看用户的详细信息，查看其中有否有与实际不符合的信息。本程序模块属于管理子系统，设计该程序模块目的在于让管理员更好地管理用户情况，防止网络诈骗等；它是整个系统有机组成部分。本模块处理方式为并行处理。

## 4．2功能

为管理员方便查询用户信息。下图为此模块的IPO图：

输入 处理 输出

管理员登录后，进入管理员界面

用户信息

用户信息更新

1. 验证管理员ID和密码，防止非本网站人员进入后台界面，进行破坏。

2.通过用户ID或用户名在数据库中搜索用户。

3.在数据库中更新对应用户信息

1.管理员登录

2.用户ID或用户名

3.用户信息更新

图 4-1查询用户IPO

## 4．3性能

程序要求有较好的容错能力、精度和灵活性，能够即时响应用户请求。

查询、登录以及更新维护时，系统响应时间要求小于1s;

## 4．4输入项

用户名（User Name）或用户ID（UserID），根据系统规定的原则或协议来进行用户注册；方便系统对用户进行查询。

表3-1 用户查询输入项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 用户ID | User ID | char | 8位数字 | 键盘输入 | 1 | 高 | 用户名称对应 | 无 |
| 用户名称 | Name | Nvarchar | 无限制 | 键盘输入 | 1 | 低 | 用户输入 | 无 |

## 4．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。

表3-2 用户管理输出项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **数量** | **频度** | **输出说明** | **安全保密条件** |
| 用户详细信息 | User Information | 字符型 | 显示用户ID、用户名、性别以及年龄等详细信息 | 图文并茂 | >=0 | 较高 | 输出用户的详细信息以及用户头像 | 登录时配以密码使用 |

## 4．6算法

此模块中只包括查询用户信息及更新，通过设置多个Text文本输入框和Combox选择框来组合用户信息，使用ListView或DataGridview来显示用户信息，通过管理员点击查询来查询用户信息及用户信息更新。

本模块中的用户查询和更新算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高查询速度。查询用户信息的算法如下：

（1）首先程序要输入用户ID或用户名;

（2）再获得用户ID或用户名；

（3）将两个参数传入函数Query（）

Bool Query (string UserID,string User Name)

{

Cin>>UserID;

Cin>> User Name;

Get User Name;

Get UserID;

Write Back to Database;

}；

（3）通过用户ID实现管理用户信息的生成并返回给用户；

## 4.7 流程逻辑

登录成功

退出系统

管理员登录

进入后台界面

输入用户ID或用户名

查询成功

查询用户

图3-2 查询用户流程逻辑

## 4．8顺序图

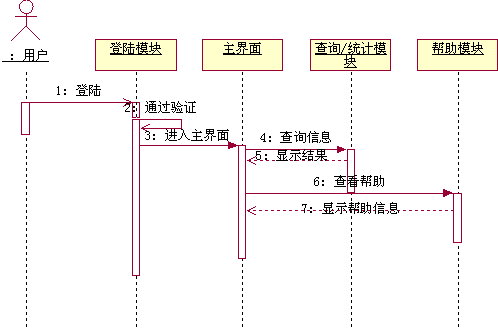


图3-3 查询用户的顺序图

## 4．9接口



图3-3 查询用户接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为查询用户信息及更新。通过管理员进入本系统后，管理员欲进入本在线系统对数据对象进行操作时开始了对本程序的调用。同样，本程序通过将用户ID或用户名作为参数传递完成对用户的查询，进入该系统。

## 4．10存储分配

本程序存储主要是对用户信息（如用户ID或用户名等等信息）的存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 4．11注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（作者姓名+8位日期）

\*

\* 修改标识：（作者姓名+8位日期）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 4．12限制条件

只有本网站管理员具有查询用户信息的权限。此模块是属于管理子系统下的子模块，登录后即可进入执行该模块。

## 4．13测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：选择用户名或用户ID，进行分别的测试。

预期结果：输出符合条件的用户信息。

（2）输入测试数据：输入无效的信息。

预期结果：程序显示输入信息，要求重新输入。

此模块的测试安排在该模块完成，当该模块完成后即对些模块进行测试，保证单个部分的正确运行，当整体完成后再回归测试。

## 4．14尚未解决的问题

目前，只有通过用户名或用户ID来查询用户，输出的结果凌乱不堪，没有优先显示或设定如按用户活跃次数来显示与查询有关的用户.

# 5 程序3增加注册用户（userRegister）设计说明

## 5．1程序描述

本模块只包括用户注册。其中用户注册中可以分为高级会员、普通会员以及游客，各种用户的权限不一样；在用户登录时，还应配合密码一起进行登录验证。本程序模块属于管理子系统，设计该程序模块目的在于让用户可以登录该系统，从而让用户去进行其他操作，它是整个系统有机组成部分。本模块为每个想注册该系统的用户创建自己的用户ID和密码以及一些便于系统管理的用户信息，能够实现重用，处理方式为并行处理。

## 5．2功能

为用户提供注册用户信息的更新。下图为此模块的IPO图：

输入 处理 输出

用户注册表，返回系统主界面

1.为用户在数据库中创建一条注册记录

1.用户信息

图 5-1用户注册IPO

## 5．3性能

程序要求有较好的容错能力、精度和灵活性，能够即时响应用户请求。

用户注册、用户登录时，系统响应时间要求小于1s;

用户ID和登录密码要求在6位到16位之间，必须为数字或英文组成（注意：不要用除数字和字母意外的其他任何符号表示）。

## 5．4输入项

用户信息（User User Information），根据系统规定的原则或协议来进行用户注册；方便系统对用户进行维护管理以及用户对自己信息的更新和修改。

用户ID（UserID）和密码（SecretID），通过验证用户ID和密码是否和注册的用户ID和密码一致，直接点击登录就可登录该在线系统。

表5-1 用户管理输入项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 用户ID | User ID | char | 8位数字 | 系统自动识别 | 1 | 高 | 用户名称对应 | 无 |
| 用户名称 | Name | Nvarchar | 无限制 | 键盘输入 | 1 | 低 | 用户输入 | 无 |
| 用户密码 | SecretID | Nchar | 6-16位字符 | 键盘输入 | 1 | 高 | 用户输入 | 无 |
| 年龄 | Age | Char | 1-100之间的数字 | 界面上直接选择 | 1 | 低 | 用户选择 | 无 |
| 性别 | Sex | Char | ‘男’或‘女’ | 界面上直接选择 | 1 | 低 | 用户选择 | 无 |
| 联系方式 | tel | char | 无 | 用户输入 | 1 | 低 | 用户输入 | 无 |

## 5．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。

表4-2 用户管理输出项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **数量** | **频度** | **输出说明** | **安全保密条件** |
| 用户注册情况 | User Register | 字符型 | 符合条件的用户信息 | true  or  false | >=0 | 较高 | 为布尔变量，true为成功，false为失败 | 无 |

## 5．6算法

此模块中包括用户注册、用户登录以更新，通过设置多个Text文本输入框和Combox选择框来组合用户信息，使用ListView或DataGridview来显示用户信息，通过用户点击注册、登录或确认提交Button，生成用户注册、登录以及用户信息更新。

本模块中的用户注册、用户登录以及更新算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高注册速度。在线注册用户信息的算法如下：

（1）首先程序要输入用户ID和用户密码;

（2）再获得用户ID和用户密码；

（3）将两个参数传入函数Register（）

Bool Register(string UserID,string SecretID)

{

Cin>>UserID;

Cin>>SecretID;

Get ClothID;

Get UserID;

Write Back to Database;

}；

（3）通过用户ID实现管理用户信息的生成并返回给用户；

5．7流程逻辑

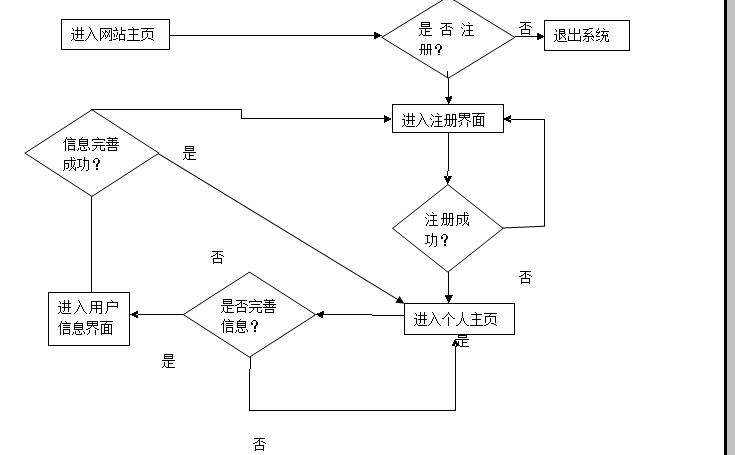


图4-2 用户注册流程逻辑

## 5．8接口



图4-3 用户注册接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为用户注册，用户登录以及用户信息更新。通过用户浏览本系统后，用户欲进入本在线系统对数据对象进行操作时开始了对本程序的调用。同样，本程序通过将用户ID作为参数传递完成用户注册，将用户ID和密码作为参数传递完成系统登录，进入该系统。

## 5．9存储分配

本程序存储主要是对用户信息（如用户ID、密码等等信息）的存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 5．10注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（作者姓名+8位日期）

\*

\* 修改标识：（作者姓名+8位日期）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 5．11限制条件

所有用户都具有注册的权限，在注册之后才有登录的权限。此模块是属于管理子系统下的子模块，注册后即可进入执行该模块。

## 5．12测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：选择男女、年龄等，进行分别的测试。

预期结果：输出符合条件的用户信息。

（2）输入测试数据：输入无效的信息。

预期结果：程序显示输入信息，要求重新输入。

（3）输入测试数据：输入正确和错误的密码。

预期结果：密码正确时登录成功，否则提示密码出错，五次密码错误后24小时内不得再次登录。

此模块的测试安排在该模块完成，当该模块完成后即对些模块进行测试，保证单个部分的正确运行，当整体完成后再回归测试。

## 5．13尚未解决的问题

目前，未为用户灵活的等级升级，即用户权限只能在建立会员表的时候已固定，升级的时候只能通过修改会员表的会员等级。

# 6 程序5 用户信息修改（userEdit）设计说明

## 6．1程序描述

本模块只包括用户信息修改。其中用户注册成为本站会员后，在用户信息维护中，管理员可以对用户信息进行维护，即如果并非用户本人，是不允许修改用户的详细信息；在用户信息修改中，用户可以修改其基本信息以及完善自己的信息。本程序模块属于管理子系统，设计该程序模块目的在于让用户可以登录该系统，从而让用户去进行其他操作，以及方便系统对用户进行管理；它是整个系统有机组成部分。本模块为每个想注册成为本站的用户修改其基本信息以及完善自己的信息，能够实现重用，处理方式为并行处理。

## 6．2功能

为用户提供修改其基本信息以及完善自己的信息。下图为此模块的IPO图：

输入 处理 输出

登录成功，进入会员信息界面

确认修改后的用户信息

1. 验证用户ID和密码

2. 在数据库中更新用户信息

3.在数据库中更新对应用户信息

1. .用户登录

2.修改的基本信息

图 5-1 用户信息修改IPO

## 6．3性能

程序要求有较好的容错能力、精度和灵活性，能够即时响应用户请求。

用户信息修改时，系统响应时间要求小于1s;

查看用户详细信息时，系统可以显示用户的好友；

用登录密码要求在6位到16位之间，必须为数字或英文组成（注意：不要用除数字和字母意外的其他任何符号表示）。

## 6．4输入项

用户信息（User User Information），根据系统规定的原则或协议来进行对自己信息的修改。

表5-1 用户管理输入项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 用户ID | User ID | char | 8位数字 | 系统自动识别 | 1 | 高 | 用户名称对应 | 无 |
| 用户名称 | Name | Nvarchar | 无限制 | 键盘输入 | 1 | 低 | 用户输入 | 无 |
| 用户密码 | SecretID | Nchar | 6-16位字符 | 键盘输入 | 1 | 高 | 用户输入 | 无 |
| 年龄 | Age | Char | 1-100之间的数字 | 界面上直接选择 | 1 | 低 | 用户选择 | 无 |
| 性别 | Sex | Char | ‘男’或‘女’ | 界面上直接选择 | 1 | 低 | 用户选择 | 无 |
| 联系方式 | tel | char | 无 | 用户输入 | 1 | 低 | 用户输入 | 无 |

## 6．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。

表5-2 用户管理输出项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **数量** | **频度** | **输出说明** | **安全保密条件** |
| 用户详细信息 | User Information | 字符型 | 显示用户ID、用户名、性别以及年龄等详细信息 | 图文并茂 | >=0 | 较高 | 输出用户的详细信息以及用户头像 | 登录时配以密码使用 |
| 用户信息更新 | Update | 字符型 | 无 | 对话框 | 1 | 较高 | 更新是否成功以及详细说明 | 无 |

## 6．6算法

此模块中只包括用户信息修改及完善，通过设置多个Text文本输入框和Combox选择框来组合用户信息，使用ListView或DataGridview来显示用户信息，通过用户信息修改来完成对用户自己的信息进行修改和完善。

本模块中的用户信息修改及完善算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高修改速度。

6．7流程逻辑

登录成功

用户登陆

退出系统

进入系统主页

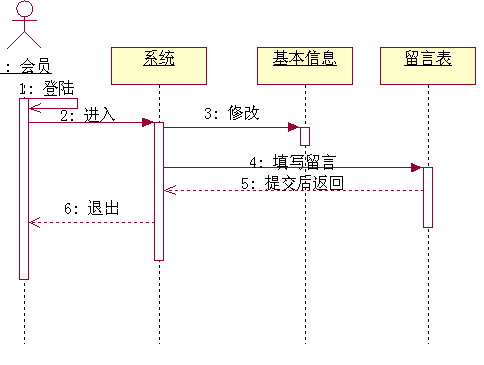
修改成功

是否修改信息

进入用户信息界面

图5-2 用户信息修改流程逻辑

## 6．8顺序图

****

## 6．9接口



图5-3 用户信息修改接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为用户信息修改及完善。通过用户浏览本系统后，用户欲进入本在线系统对数据对象进行操作时开始了对本程序的调用。

本程序在数据库中主要和会员表发生联系。当用户注册后才能进行登录系统，进行用户操作， 如：用户信息的修改。

## 6．10存储分配

本程序存储主要是对用户信息（如用户ID、密码等等信息）的存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 6．11注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（作者姓名+8位日期）

\*

\* 修改标识：（作者姓名+8位日期）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 6．12限制条件

所有用户都具有修改个人信息的权限，在登录之后才有修改的权限。此模块是属于管理子系统下的子模块，登录后即可进入执行该模块。

## 6．13测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：选择男女、年龄等，进行分别的测试。

预期结果：输出符合条件的用户信息。

（2）输入测试数据：输入正确和错误的密码。

预期结果：密码连续输入两次且相同才能修改成功，否则提示密码出错。

此模块的测试安排在该模块完成，当该模块完成后即对些模块进行测试，保证单个部分的正确运行，当整体完成后再回归测试。

## 6．14尚未解决的问题

目前，日前系统处于初始阶段，只设了一项二级密码保护措施，建议用户通过邮箱或手机验证进行修改个人信息。对于怎样解决用户的密码保护问题，我们希望用密保卡来解决，但怎样设计好密保卡目前还是一项难题。

# 7 程序5 用户删除（deleteUser）设计说明

## 7．1程序描述

本模块识用于对用户的删除处理。主要是基于用户在网上交易非法物品、欺骗其他用户等等，一经确认，管理员可以对用户进行删除，情节严重者予以追究其法律责任。本程序模块属于管理子系统，设计该程序模块目的在于删除用户，能够实现重用，处理方式为并行处理。

## 7．2功能

为用户进行删除处理。下图为此模块的IPO图：

输入 处理 输出

登录成功，进入后台界面

显示该用户的信息

确认删除该用户

1. 验证管理员身份

2. 在数据库中找到对应用户信息

3.在数据库中更新用户信息

1. .管理员登录

2.欲删除的用户ID

3删除该用户

6-1 用户删除IPO图

## 7．3性能

程序要求有超差的容错能力、超高的精度和超强的灵活性，能够即时响应请求。

对用户进行删除时，系统响应时间要求小于3s;

## 7．4输入项

用户ID（UserID），通过输入用户ID来确定用户并对进行删除处理。

表7-1 用户管理输入项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 用户ID | User ID | char | 8位数字 | 系统自动识别 | 1 | 高 | 用户名称对应 | 无 |

## 7．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。

表6-2 用户管理输出项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **数量** | **频度** | **输出说明** | **安全保密条件** |
| 用户详细信息 | User Information | 字符型 | 显示用户ID、用户名、性别以及年龄等详细信息 | 图文并茂 | >=0 | 较高 | 输出用户的详细信息以及用户头像 | 登录时配以密码使用 |
| 用户信息更新 | Update | 字符型 | 无 | 对话框 | >=0 | 较高 | 更新是否成功以及详细说明 | 无 |

## 7．6算法

此模块中只包括用户删除，通过设置多个Text文本输入框和Combox选择框来组合用户信息。

本模块中的用户删除算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高查询速度。在线注册用户信息的算法如下：

（1）首先程序要输入用户ID;

（2）再获得用户ID；

（3）将这个参数传入函数Delete（）

void Delete (string UserID)

{

Cin>>UserID;

Get ClothID;

Delete User-Information;

Write Back to Database;

}；

（3）通过用户ID来实现对用户的删除；

## 7．7流程逻辑

管理员登陆

登录成功

进入后台界面

输入用户ID

更新用户信息

退出系统

是否找到该用户

否

是否删除用户

找到该用户

是

图6-2 用户删除流程逻辑

## 7．8接口



图7-3 用户删除接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为删除用户。同样，本程序通过将用户ID作为参数传递完成用户的删除。

## 7．9存储分配

本程序存储主要是对删除用户的时间、原因等记录的存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 7．10注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（作者姓名+8位日期）

\*

\* 修改标识：（作者姓名+8位日期）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 7．11限制条件

删除用户只有网站管理员才有权限。

## 7．12尚未解决的问题

目前，系统删除用户后为永久删除，无法恢复，同样，用户的ID他人也无法再用。

# 8 程序6 商品查询（findProudct） 设计说明

## 8．1程序描述

本模块只包括商品查询。在商品查询中，用户可以查看所有商品的详细信息。本程序模块属于管理子系统，设计该程序模块目的在于让用户更好地找到所需的二手商品；它是整个系统有机组成部分。本模块处理方式为并行处理。

## 8．2功能

为用户查询商品信息。下图为此模块的IPO图：

输入 处理 输出

1用户登录

2.商品名

用户登录后，进入系统主界面。

商品信息列表

1. 验证用户ID和密码。

2.通过商品名来查找对应的商品

图 1-1查询用户IPO

## 8．3性能

程序要求有较好的容错能力、精度和灵活性，能够即时响应用户请求。

查询、登录时，系统响应时间要求小于1s;

## 8．4输入项

商品名 (ProductName）和用户ID（UserID），根据系统规定的原则或协议来查询商品。

表1-1 用户查询输入项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 商品名称 | ProductName | varchar | 无限制 | 键盘输入 | 1 | 低 | 用户输入 | 无 |

## 8．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。

表1-2 用户管理输出项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **数量** | **频度** | **输出说明** | **安全保密条件** |
| 商品详细信息 | Commodity  Information | 字符型 | 显示商品名、数量、种类等详细信息 | 图文并茂 | >=0 | 较高 | 输出商品的详细信息以及用户头像 | 登录时配以密码使用 |

## 8．6算法

此模块中只包括商品信息查询，通过设置多个Text文本输入框和Combox选择框来组合用户信息，使用ListView或DataGridview来显示商品信息，通过用户输入商品名来查询对应的商品信息。

本模块中的商品信息查询算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高查询速度。商品信息查询的算法如下：

（1）首先程序要输入商品名;

（2）再获得商品名；

（3）将两个参数传入函数Query（）

Bool Query (string Commodity Name)

{

Cin>> Commodity Name;

Get Commodity Name;

Write Back to Database;

}；

（3）通过点击查询按钮显示商品信息并返回给用户；

## 8.7 流程逻辑

登录成功

用户登录

退出系统

返回

显示商品信息列表

进入系统主界面

输入商品名

查询成功

查询商品

图1-2 查询商品流程逻辑

## 8．8顺序图

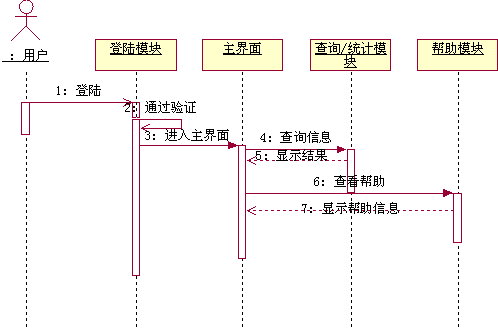


图1-3 查询商品的顺序图

## 8．9接口



图1-3 查询用户接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为查询商品信息。通过用户进入本系统后，用户欲进入本在线系统对数据对象进行操作时开始了对本程序的调用。同样，本程序通过将商品名作为参数传递完成对商品信息的查询，进入该系统。

## 8．10存储分配

本程序存储主要是对商品信息（商品名、种类、数量等等信息）的存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 8．11注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（商品名）

\*

\* 修改标识：（商品名）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 8．12限制条件

只有用户登录后才能进行商品查询。此模块是属于管理子系统下的子模块，登录后即可进入执行该模块。

## 8．13测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：选择商品哆嗦，进行分别的测试。

预期结果：输出符合条件的商品信息。

（2）输入测试数据：输入无效的信息。

预期结果：程序显示输入信息，要求重新输入。

此模块的测试安排在该模块完成，当该模块完成后即对些模块进行测试，保证单个部分的正确运行，当整体完成后再回归测试。

# 9 程序7 商品发布（addProduct）设计说明

## 9．1程序描述

本模块只包括用户发布商品。本程序模块属于管理子系统，设计该程序模块目的在于让用户能方便地发布自己不用的二手商品；它是整个系统有机组成部分。本模块处理方式为并行处理

## 9．2功能

为用户发布商品。下图为此模块的IPO图：

输入 处理 输出

用户登录后，进入发布商品界面

该商品信息表

商品信息更新

1. 验证管理员ID和密码。

2.验证商品信息的真假

3.在数据库中更新商品信息

1用户登录

2.商品的详细信息

3.商品信息更新

图 9-1查询用户IPO

## 9．3性能

程序要求有较好的容错能力、精度和灵活性，能够即时响应用户请求。

发布商品及更新维护时，系统响应时间要求小于1s;

## 9．4输入项

商品（Product Information），根据系统规定的原则或协议来进行发布商品。

表9-1 用户查询输入项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 商品详细信息 | Commodity  Information | 字符型 | 显示商品名、数量、种类等详细信息 | 图文并茂 | >=0 | 较高 | 商品的详细信息 | 登录时配以密码使用 |

## 9．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。

表2-2 用户管理输出项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **数量** | **频度** | **输出说明** | **安全保密条件** |
| 商品详细成功信息 | Product  Information | 字符型 | 显示商品名、数量、种类等详细信息 | 图文并茂 | >=0 | 较高 | 商品的详细信息 | 登录时配以密码使用 |

## 9．6算法

此模块中只包括商品发布及更新，通过设置多个Text文本输入框和Combox选择框来组合商品信息，使用ListView或DataGridview来显示商品信息，通过用户来发布商品信息及更新。

本模块中的商品发布和更新算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高发岂有此理速度。

## 9.7 流程逻辑

登录成功

用户登录

退出系统

更新成功

进入商品发布界面

更新商品信息

输入商品信息

否

是

发布成功？

发布商品

图9-2 查询商品流程逻辑

## 9．8顺序图

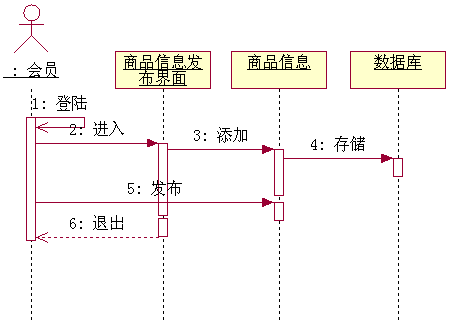


图2-3 查询用户的顺序图

## 9．9接口



图2-3 查询用户接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为发布商品信息及更新。通过用户进入本系统后用户欲进入本在线系统对数据对象进行操作时开始了对本程序的调用。

## 9．10存储分配

本程序存储主要是商品信息（如商品名、种类、数量等等信息）的存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 9．11注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（商品信息）

\*

\* 修改标识：（商品信息）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 9．12限制条件

只有本网站用户拥有发布商品的权限，且是有偿发布。此模块是属于管理子系统下的子模块，登录后即可进入执行该模块。

## 9．13测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：选择商品名或大小等，进行分别的测试。

预期结果：输出符合条件的商品信息。

（2）输入测试数据：输入无效的信息。

预期结果：程序显示输入信息，要求重新输入。

此模块的测试安排在该模块完成，当该模块完成后即对些模块进行测试，保证单个部分的正确运行，当整体完成后再回归测试。

## 9．14尚未解决的问题

目前，由于发布商品为有偿发布，支付方式只支持网上银行支付，但这一块还没有做好。

# 10 程序10 商品修改(updateproduct) 设计说明

## 10．1程序描述

本模块只包括商品修改。在商品修改中，用户可以修改商品的详细信息。本程序模块属于管理子系统，设计该程序模块目的在于及时跟新产品信息，方便用户了解；它使整个系统更加合理，进一步提升了系统的功能。本模块处理方式为并行处理。

## 10．2功能

输入:用户输入要修改的商品

处理:系统更新商品信息

输出:显示更新后的商品信息。

## 10．3性能

程序要求有较好的容错能力、精度和灵活性，能够即时响应管理员请求。

查询、登录时，系统响应时间要求小于1s;

## 10．4输入项

商品名 (ProductName）和用户ID（UserID），根据系统规定的原则或协议来修改商品。

表1-1 用户修改商品项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 商品名称 | ProductName | varchar | 无限制 | 鼠标点击或键盘输入查询得出 | 1 | 低 | 用户操作 | 无 |

## 10．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。输出信息显示在屏幕上，它包括用户更改的信息和没有更改的信息。

表1-2 用户管理输出项

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **频度** | **输出说明** | **安全保密条件** |
| 商品详细信息 | Commodity  Information | 字符型 | 显示商品名、数量、种类等详细信息 | 图文并茂 | 较高 | 输出商品的详细信息 | 登录时配以密码使用 |

## 10．6算法

此模块中只包括商品信息修改，通过设置多个Text文本输入框和Combox选择框来组合用户信息，使用ListView或DataGridview来显示商品信息，通过用户输入商品名或点击一个商品来修改对应的商品信息。

本模块中的商品信息查询算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高查询速度。商品信息查询的算法如下：

（1）首先程序要输入商品名或通过点击确定商品名;

（2）将两个参数传入函数Update（）

Bool Update (string Commodity Name)

{

Cin>> Commodity Name;

Get Commodity Name;

Write Back to Database;

}；

（3）通过点击确定修改按钮将信息写入数据库保存；

## 10.7 流程逻辑

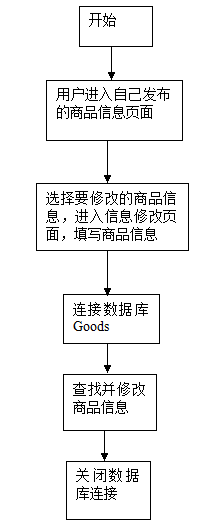


图1-2 修改商品流程逻辑

## 10．8顺序图

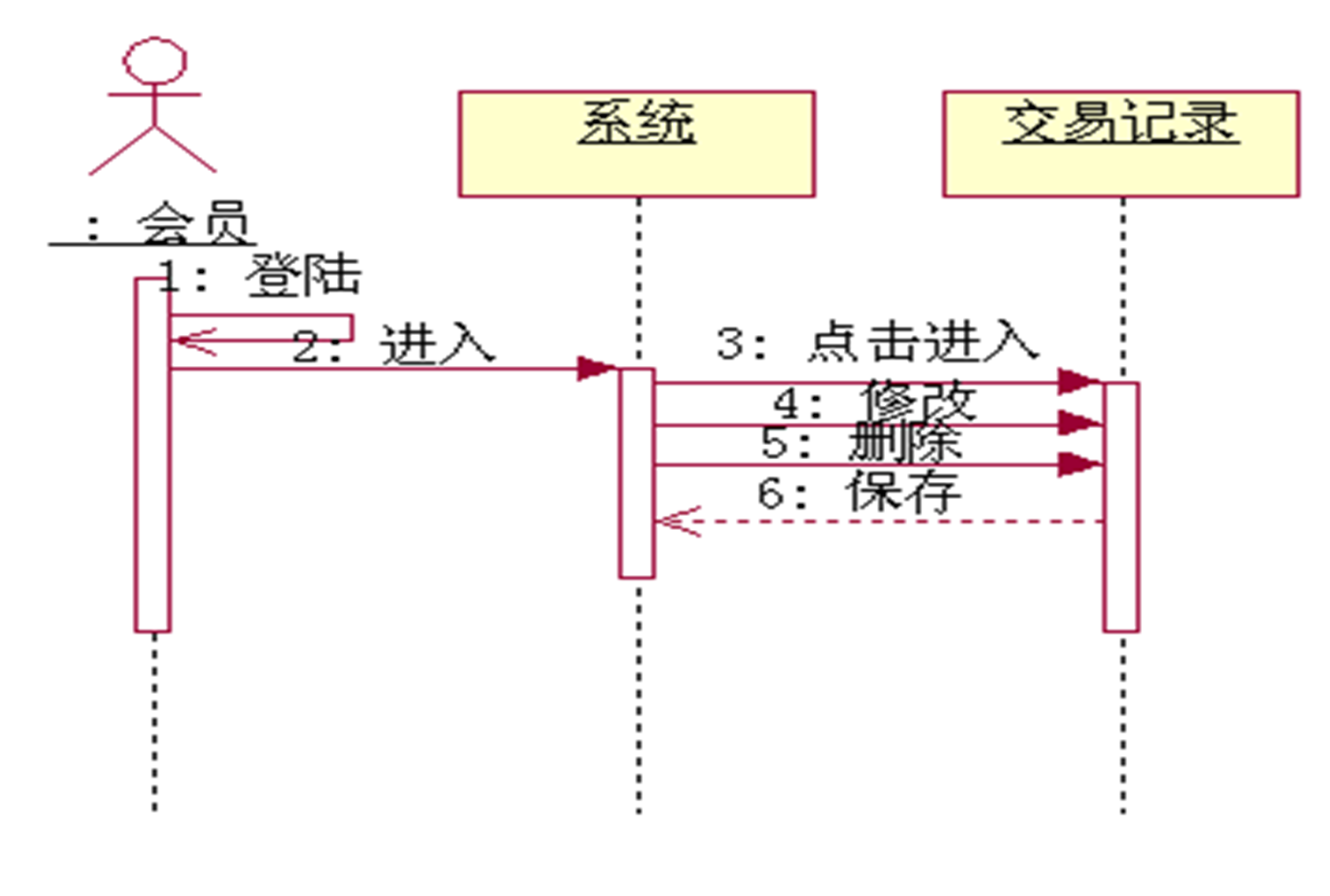


图1-3 修改商品的顺序图

## 10．9接口



图1-3 修改商品接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为商品修改信息。通过管理员进入本系统后，用户欲进入本在线系统对数据对象进行操作时开始了对本程序的调用。同样，本程序通过将商品名作为参数传递完成对商品信息的修改，进入该系统。

## 10．10存储分配

本程序存储主要是对修改的商品信息（商品名、种类、数量等等信息）的存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 10．11注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（商品名）

\*

\* 修改标识：（商品名）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 10．12限制条件

只有用户登录后才能进行商品修改。此模块是属于管理子系统下的子模块，登录后即可进入执行该模块。

## 10．13测试计划

测试技术要求为专业测试。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性。

（1）输入测试数据：点击商品或输入要修改的商品名，进行分别的测试。

预期结果：输出用户更改过的完整商品信息。

（2）输入测试数据：输入无效的信息。

预期结果：程序显示输入信息，要求重新输入。

此模块的测试安排在该模块完成，当该模块完成后即对些模块进行测试，保证单个部分的正确运行，当整体完成后再回归测试。

# 11 程序10 商品删除(deleteproduct) 设计说明

## 11．1程序描述

本模块只包括商品删除。在商品删除中，管理员可以删除选择的商品。本程序模块属于管理子系统，设计该程序模块目的在于及时跟新该系统，使该系统减少冗余，优化系统；它使整个系统更加合理，进一步提升了系统的功能。本模块处理方式为并行处理。

## 11．2功能

输入:管理员输入要删除的商品

处理：系统删除商品信息

输出：系统显示删除商品成功与否；

## 11．3性能

程序要求有较好的容错能力、精度和灵活性，能够即时响应管理员请求。

查询、登录时，系统响应时间要求小于1s;

## 11．4输入项

商品名 (ProductName）和管理员ID（UserID），根据系统规定的原则或协议来删除商品。

表1-1 管理员删除商品项

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输入方式** | **数量** | **频度** | **来源** | **安全保密条件** |
| 商品名称 | ProductName | varchar | 无限制 | 鼠标点击或键盘输入查询得出 | 1 | 低 | 管理员操作 | 无 |

## 11．5输出项

未列出的输出数据类型及意义均与输入时的数据类型一致。输出信息显示在屏幕上，它包括删除商品是否成功的信息。

表1-2 用户管理输出项

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | **标识** | **数据类型** | **数据有效范围** | **输出形式** | **频度** | **输出说明** | **安全保密条件** |
| 商品详细信息 | Commodity  Information | 字符型 | 显示商品名、数量、种类等详细信息 | 对话框 | 较高 | 删除是否成功 | 登录时配以密码使用 |

## 11．6算法

此模块中只包括商品删除，通过管理员输入商品名或点击一个商品来删除对应的商品。

本模块中的商品删除算法均由数据库内部算法实现，不再单独写算法，只需在数据库中建立相关的索引来提高查询速度。商品信息查询的算法如下：

（1）首先程序要输入商品名或通过点击确定商品名;

（2）将两个参数传入函数Delete（）

Bool Delete (string Commodity Name)

{

Cin>> Commodity Name;

Get Commodity Name;

Write Back to Database;

}；

1. 通过点击确定删除按钮将信息写入数据库保存；

## 11.7 流程逻辑

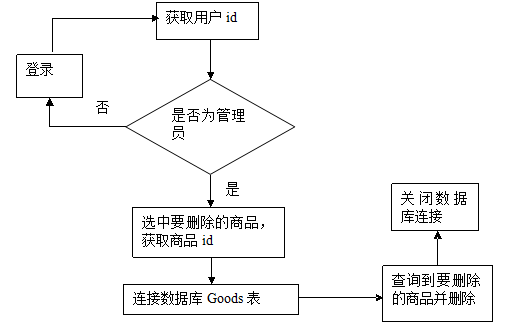


图1-2 删除商品流程逻辑

## 11．8顺序图

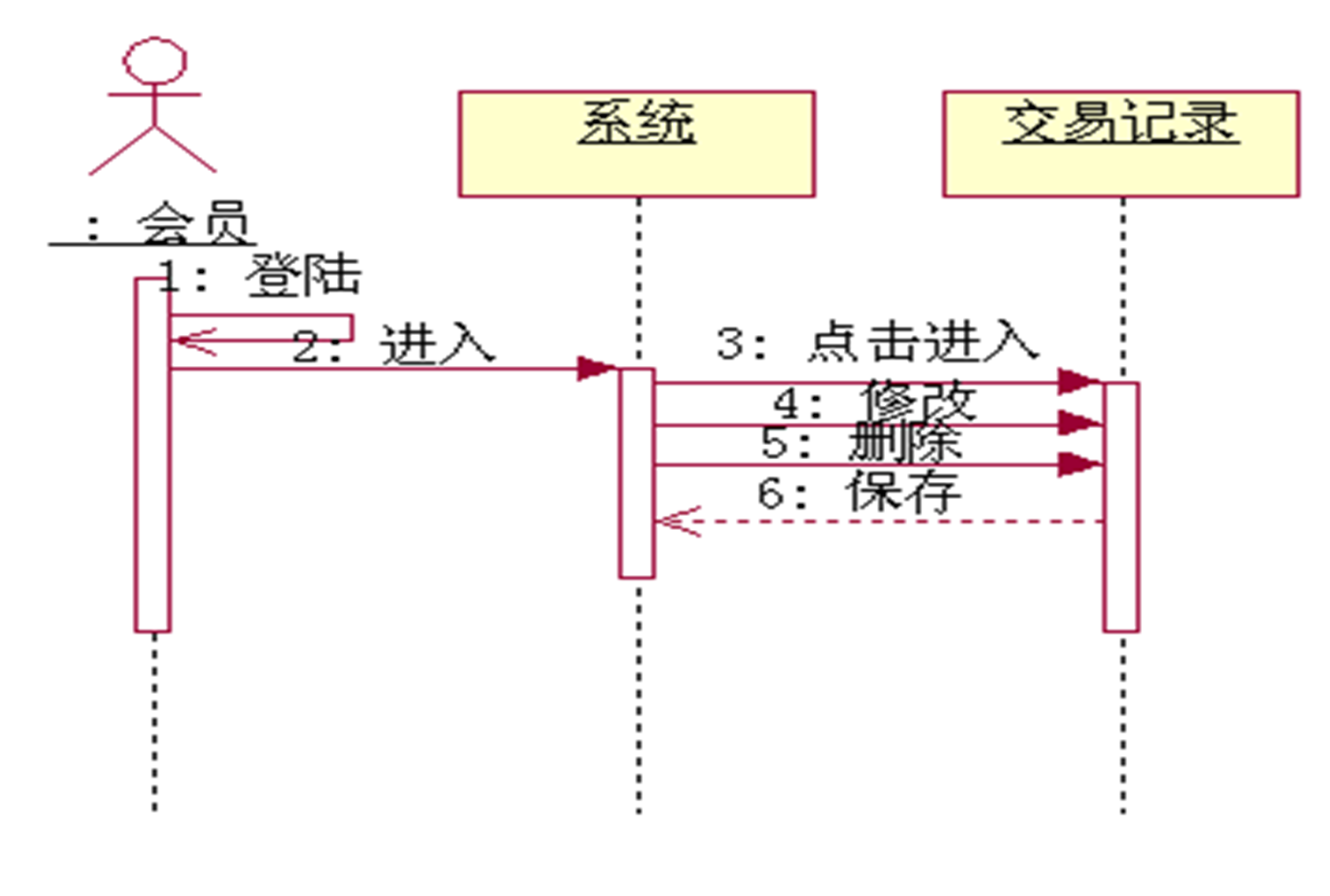


图1-3 删除商品的顺序图

## 11．9接口



图1-3 删除商品接口层次

调用本程序的上一层程序为管理子系统，本程序调用的下一层程序为商品删除。通过管理员进入本系统后，管理员欲进入本在线系统对数据对象进行操作时开始了对本程序的调用。同样，本程序通过将商品名作为参数传递完成对商品的删除，进入该系统。

## 11．10存储分配

本程序存储主要是对商品删除，删除存储在数据库上的商品信息。并保存删除日志到一个日志文件中。删除日志中记录该商品的信息。

## 11．11注释设计

　　a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（商品名）

\*

\* 修改标识：（商品名）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 11．12限制条件

只有管理员登录后才能进行商品删除。此模块是属于管理子系统下的子模块，登录后即可进入执行该模块。

## 11．13测试计划

测试技术要求为专业测试。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性。

（1）输入测试数据：点击商品或输入要删除的商品名，进行测试。

预期结果：删除的商品已不在该系统中。管理员可通过查询日志文件找到该商品信息，用户不能在目睹该商品信息。

（2）输入测试数据：输入无效的商品。

预期结果：程序显示该商品无效，要求重新输入。

此模块的测试安排在该模块完成，当该模块完成后即对些模块进行测试，保证单个部分的正确运行，当整体完成后再回归测试。

# 12 程序10 下订单（addOrder）设计说明

## 12．1程序描述

下订单：该模块是购物系统最核心的流程，用户必须要下订单之后才能得到卖家信息，然后与卖家进行面对面交易。

## 12．2功能

用户购买商品的必要步骤，提交商品信息和买家信息。

## 12．3性能

程序有较高的准确性，能够即时返回用户下订单是否成功以及相关信息。

## 12．4输入项

输入订单信息，具体信息如下表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **数据长度** | **精度** | **主/外键** | **能否为空** | **描述** |
| 会员编号 | MemberNO | Char | 10 | 无 | PK/FK | 否 | 每张订单都有一个会员 |
| 订购日期 | IndentDate | Datetime | 无 | 无 | 无 | 否 | 用户下订单的时间 |
| 订购数量 | Orderamount | Int | 无 | 0 | 无 | 是 | 用户订购商品数量 |
| 订购金额 | TotalAmount | Float | 无 | 0 | 无 | 是 | 记录用户此次订购金额总和 |

## 12．5输出项

返回用户下订单是否成功，成功则返回订单信息，否则返回下订单失败以及原因。

## 12．6算法

本模块只包括用户下订单，用户点击确定之后，系统自动获取购物车中的商品信息和用户的信息，自动生成订单编号，生成订单，发送到服务器。后台处理后，成功则返回订单信息，否则返回下订单失败以及失败原因。

## 12．7流程逻辑

开始

用户选择要买的商品，并确认

用户登录

连接数据库

否

查询是否存在用户

是

生成订单

是

是否继续购买

否

将商品信息存入数据库中

关闭数据连接

用户下订单流程图

## 12．8接口



## 12.9存储分配

本程序存储主要是商品信息和订单信息进行存储，所有数据都存储服务器上，对每一块数据区域建立相应的数据表。详见《概要设计说明书》的数据库设计部分。

## 12．10注释设计

a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（订单信息）

\*

\* 修改标识：（订单信息）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 12．11限制条件

只有注册并登陆的会员才能下订单。

## 12．12测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：输入订单信息

预期结果：返回下订单成功以及订单信息

此模块的测试安排在所有模块完成之后，因为此模块正常运行需要其他很多模块的支持。

## 12．13尚未解决的问题

由于该系统采取的是当面交易，用户下订单之后的和卖家的交易处理起来比较麻烦。

# 13 程序11 订单查询（findOrder）设计说明

## 13．1程序描述

用户有时候会查询以前自己所下过的订单信息，也可以对所下的订单作修改，修改订单和删除订单都必须先查询订单。

## 13．2功能

用户查询自己以前所下过的订单信息

## 13．3性能

程序有较高的准确性，能够即时返回用户所查询的订单信息。

## 13．4输入项

输入订单编号，具体信息如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **数据长度** | **精度** |
| 订单编号 | OrderNO | Char | 10 | 无 |

## 13．5输出项

成功则返回用户订单信息，否则返回错误信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **数据长度** | **精度** | **能否为空** | **描述** |
| 订单编号 | OrderNO | Char | 10 | 无 | 否 | 为了便于系统管理，每张订单都有唯一的编号 |
| 会员编号 | MemberNO | Char | 10 | 无 | 否 | 每张订单都有一个会员 |
| 订购日期 | IndentDate | Datetime | 无 | 无 | 否 | 用户下订单的时间 |
| 订购数量 | Orderamount | Int | 无 | 0 | 是 | 用户订购商品数量 |
| 订购金额 | TotalAmount | Float | 无 | 0 | 是 | 记录用户此次订购金额总和 |

## 13．6算法

本模块只包括用户查询订单，用户输入要查询的订单编号，确定后，发送至后台，后台处理后，查询成功则返回所查询到的订单信息，失败则返回错误信息。

## 13．7流程逻辑

开始

关闭数据库连接

是否登陆

否

是

登陆

是否存在该订单编号

否

是

输入要查询的订单编号

查询到输入的订单编号OrderNO

连接数据库SalesOrder表

查询订单流程图

## 13．8接口



## 13．9存储分配

本程序无相关数据的存储。

## 13．10注释设计

a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（订单信息）

\*

\* 修改标识：（订单信息）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 13．11限制条件

只有注册并登陆的会员才能下订单。

## 13．12测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：输入订单编号（存在的）

预期结果：返回下订单成功以及订单信息

（1）输入测试数据：输入订单编号（不存在的）

预期结果：返回错误信息：你输入的订单编号不存在

此模块的测试安排在该模块完成之后。

## 13．13尚未解决的问题

考虑管理员有没有权限去修改和删除订单。

# 14 程序12 修改订单（changeOrder）设计说明

## 14．1程序描述

用户有时候由于输入错误或者其他原因需要修改自己尚未交易的订单。

## 14．2功能

用户修改自己所下的尚未交易的订单。

## 14．3性能

程序有较高的准确性，能够即时返回用户所查询的订单信息。

## 14．4输入项

输入要修改的新订单信息，具体信息如下表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **数据长度** | **精度** | **能否为空** | **描述** |
| 订单编号 | OrderNO | Char | 10 | 无 | 否 | 为了便于系统管理，每张订单都有唯一的编号 |
| 会员编号 | MemberNO | Char | 10 | 无 | 否 | 每张订单都有一个会员 |
| 订购日期 | IndentDate | Datetime | 无 | 无 | 否 | 用户下订单的时间 |
| 订购数量 | Orderamount | Int | 无 | 0 | 是 | 用户订购商品数量 |
| 订购金额 | TotalAmount | Float | 无 | 0 | 是 | 记录用户此次订购金额总和 |

## 14．5输出项

成功则修改后的订单信息，否则返回错误信息。

## 14．6算法

本模块只包括用户修改订单，用户查询到订单信息后，输入新的订单信息，确定后，发送至后台，后台处理后，修改成功则返回修改成功和修改后的新订单信息，失败则返回错误信息。

## 14．7流程逻辑

开始

返回修改订单成功和修改后的订单信息

获得要修改的订单信息之后，进入订单修改页面

修改数据库中的相应订单信息

输入新的订单信息

否

修改订单流程图

## 14．8接口



**14．9存储分配**

本程序只作了数据库中订单的修改操作，修改数据库SalesOrder中某条订单的具体信息

**14．10注释设计**

a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（订单信息）

\*

\* 修改标识：（订单信息）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 14．11限制条件

只有注册并登陆了的会员才能作该操作，且只能修改自己的所下的。

## 14．12测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

（1）输入测试数据：输入订单信息

预期结果：返回修改订单成功以及修改之后的新订单信息

此模块的测试安排在该模块完成之后。

## 14．13尚未解决的问题

考虑管理员有没有权限去修改。

# 15 程序13 删除订单（deleteOrder）设计说明

## 15．1程序描述

用户有时候由于某种原因需要删除自己尚未交易的订单。或者交易执行完毕后需要删除交易完毕的订单

## 15．2功能

用户删除自己所下的尚未交易的订单。用户确认交易之后删除该交易的订单。

## 15．3性能

程序有较高的灵活性，能够很快的删除用户要删除的订单，特别是在用户确认交易后要自动删除该交易的订单。

## 15．4输入项

输入要删除的订单编号，具体信息如下表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **数据长度** | **精度** | **能否为空** | **描述** |
| 订单编号 | OrderNO | Char | 10 | 无 | 否 | 为了便于系统管理，每张订单都有唯一的编号 |

## 15．5输出项

用户手动删除时返回删除成功。

当用户确认交易时系统自动删除订单，无返回信息。

## 15．6算法

本模块包括用户删除订单和系统自动删除订单。用户删除订单：用户查询到订单信息后，输入新的订单信息，确定后，发送至后台，后台处理后，修改成功则返回修改成功和修改后的新订单信息，失败则返回错误信息。系统自动删除订单：用户确认交易后。系统自动获取该交易的订单编号，并删除该订单，无返回信息。

## 15．7流程逻辑

开始

返回删除订单成功信息

修改数据库中的相应订单信息

获得要删除的订单信息之后，进入订单删除页面，点击删除订单

用户删除订单流程图

## 15．8接口



## 15．9存储分配

本程序只作了数据库中订单的删除操作，删除数据库SalesOrder中某条订单的所有信息

## 15．10注释设计

a． 加在模块首部的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名称：

\* 文件功能描述：

\*

\* 创建标识：（订单信息）

\*

\* 修改标识：（订单信息）

\* 修改描述：

\*

\* 修改标识：

\* 修改描述：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b．加在各函数处的注释：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*函数名：

\*函数功能：

\*参数说明：

\*返回值：

\*作者：

\*时间：

\*备注：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 15．11限制条件

只有注册并登陆了的会员才能作该操作，且只能删除自己的所下的。

## 15．12测试计划

测试技术要求为专业和非专业。专业测试由专门的测试人员对代码进行分析，使用不同的测试数据进行测试，通过白盒和黑盒测试来保证此模块的正确性；非专业为用户进行操作上的测试，主要针对操作习惯，视觉感受进行测试。

1. 输入测试数据：删除查询到的订单
2. 预期结果：返回删除订单成功

此模块的测试安排在该模块完成之后。

## 15．13尚未解决的问题

考虑管理员有没有权限去删除。