

**<<“微服务”校园服务信息平台>>**

**综述报告**



北京航空航天大学

2016-12

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V1.1 |  | 陈益民 | 武洋阳 | 初次撰写版本 |
| V1.2 |  | 陈益民 | 武洋阳 | 自评修改版本 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

项目组成员信息

|  |  |  |
| --- | --- | --- |
| 小组名称 | 北航软件开发小组10组 | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 14231052 | 陈益民 | 文档主写人（80%） |
| 14231020 | 武洋阳 | 文档审核人（20%） |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目 录

1. 范围 1
   1. 标识 1
   2. 系统概述 1
   3. 文档概述 2
   4. 基线 2
2. 引用文档 3
3. 需求概述 4
   1. 功能需求 4
   2. 非功能需求 5
      1. 性能 5
      2. 并行性 5
      3. 可扩展性 6
      4. 易用性 6
      5. 安全性 6
   3. 条件和限制 6
4. 总体设计 6
   1. 硬件结构设计 6
   2. 软件结构设计 7
   3. 数据结构设计 13
      1. 公共数据结构设计 13
      2. 数据库设计 14
5. 管理员信息表 14
6. 用户信息表 15
7. 店铺信息表 15
8. 任务信息表 16
9. 交易记录 16
10. 任务类别 17
11. 服务信息表 18
    1. 接口设计 18
       1. 内部接口设计 18
12. 业务逻辑层——表现层 18
13. 数据操作层——业务逻辑层 19
    * 1. 外部接口设计 19
14. 硬件接口 19
15. 软件接口 19
    * 1. 用户界面设计 20
16. 关键技术及解决方案 20
17. 详细设计与实现 22
    1. 个人信息程序Message（）设计说明 22
       1. 程序描述 22
       2. 功能 23
       3. 流程逻辑 23
    2. 查询程序search()设计说明 24
       1. 程序描述 24
       2. 功能 24
       3. 流程逻辑 25
    3. 发布程序publish()设计说明 25
       1. 程序描述 25
       2. 功能 26
       3. 流程逻辑 26
    4. 接取任务程序receive（）设计说明 27
       1. 程序描述 27
       2. 功能 28
       3. 流程逻辑 28
    5. 支付程序pay（）设计说明 29
       1. 程序描述 29
       2. 功能 29
       3. 流程逻辑 30
    6. 审核程序examine ()设计说明 30
       1. 程序描述 30
       2. 功能 31
       3. 流程逻辑 31
18. 系统测试与运行效果 32
    1. 系统测试 32
    2. 系统运行效果 32
19. 其它 32
20. 范围
    1. 标识

文档标识号：A2016-12-28-00-09

文档标题：“微服务”校园服务信息平台需求规格说明书

项目/产品中文全称：“微服务”校园服务信息平台

项目/产品英文全称："Do Me a Favor" campus service information platform

项目/产品英文简称：DMF\_1.0

项目/产品编码：201611080947

* 1. 系统概述

本文档适用于的系统是“微服务”校园服务信息平台。“微服务”校园信息平台是一个以服务为商品进行交易的校园平台，用户可以通过此平台发布自己需要的服务，寻找可以提供此服务的人，双方通过“微服务”信息平台完成服务交易。

本项目主要承担“微服务”校园服务信息平台项目中系统的设计、开发任务。该系统主要由发布需求、查询、接取任务、在线支付、审核、申请服务店铺等6个功能模块组成。系统在开发过程中，我们调查了身边同学的一些生活中实际的需求，完善了系统的功能，结合实际的情况，对系统进行了修改和补充。然后在考虑到了支付环节的安全性后，选择了让用户自行选择支付方式，或者提供合理的通道，避免资金停留在平台，导致用户的财产安全收到侵害。另外，在开发过程中，考虑到多个用户对系统后台数据库的同时操作，我们补充了用户对与数据库并发操作的功能，提高了系统的可靠性。

本系统的投资方以及开发方都是本组的全体成员，由本组同学进行前期服务器和编程设备的投资开发。需求方则是全国高校的所有学生，这个系统可应用于全国大学的学生，为他们提供生活中的便利，广大的高校都是项目潜在的需求方。我们计划先将北航这个特殊的群体作为需求方的典型来设计实现系统。本系统的用户应是每一个在校园中的学生，他们即是系统的用户，也是系统的提供帮助的成员。

系统计划设计成为手机应用和网页平台，两个端口共享数据，能够同时使用。用户既可以在手机上下载使用这个应用，也能通过电脑，在平台的网站上发布或者提供帮助。

* 1. 文档概述

本条应概述本文档的用途和内容，并描述与其使用有关的保密性或私密性要求。

1. **文档用途：**

本文档的目的是给我们这个系统整体情况进行整合介绍，包含了前面的开发文档，需求分析文档和系统设计文档的一些内容，本文档用于对整个系统的整合与最终的系统实现。

1. **文档内容：**

本文档主要介绍了本软件从设计到实现到测试的所有过程。并且对系统的详细设计进行了讲解，以及系统的各个子模块实现的流程图。具体的系统的实现尚未完成，所以暂时不介绍系统的测试的测试结果与运行案例。

1. **文档保密性要求**

本文档的预期读者仅限于项目开发人员，项目测试人员。

* 1. 基线

所依据的需求基线是在需求分析说明书之后建立的。在SRS《需求规格分析说明书》撰写完成后确定了客户需求和产品需求，在项目后续步骤中严格遵循需求基线的步骤和要求。

所依据的设计基线是根据整个系统的设计过程，完成了《详细规格设计规格》文档后建立的。在该文档中详细说明了数据库、功能设计等设计要求。后续文档全部根据系统的设计路线完成。本文档中详细介绍了系统详细设计的内容，即各个模块的流程图以及实现的算法。

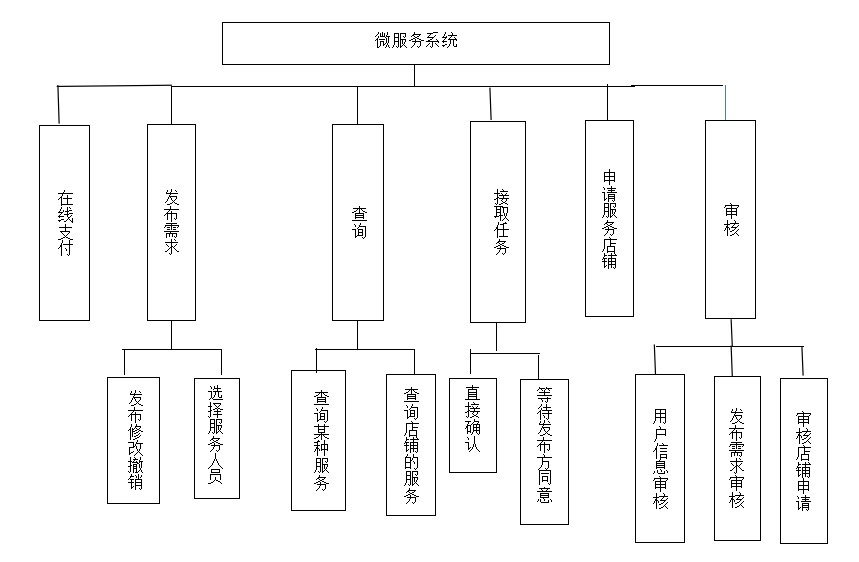
测试基线由《软件测试文档》建立。

1. 引用文档

[1]Team10. A2016-10-06-00-01.SDP-软件开发计划书

[2]Team10. A2016-11-18-13-47.SRS需求规格说明书

1. 需求概述
   1. 功能需求



简要阐述本系统各个功能的主要内容：

（1）发布需求

用户在平台上发布需求服务信息，填写任务属性（在选择服务人员之前可对信息进行修改、撤销），服务人员上前接受信息，用户从接收人中选择为用户提供服务的人员，服务完成后，用户确认受到服务，并对服务进行评价。

（2）查询

用户输入检索关键词，获得匹配条件的任务列表或服务店铺列表。

（3）接取任务

用户查询或浏览任务信息后，如果用户接取某个“请求”，系统提示给请求发布者；发布者用户选择该用户提供服务后，系统提示接受者可以提供开始服务，否则宣布接受任务失败。

（4）在线支付

发布“寻求服务”信息时，用户将悬赏金暂存于第三方平台，当确认服务完成后点击确认支付，第三方平台便会将赏金发到提供服务者的账户上。

（5）审核信息

管理员审核举报列表中的任务信息、注册资料、店铺申请，并对相应数据进行处理调停。

（6）申请服务店铺

用户满足条件可以提出成立服务店铺的申请。审核通过后该店铺加入服务店铺列表。店铺服务允许用户主动提供服务，供其他用户选择。

* 1. 非功能需求
     1. 性能

本系统作为多用户服务系统，需要支持对大规模数据的索引，因此需要具有很高的效率。这主要体现在两个方面：

1. 能高效实现数据增删改查

如果服务发布信息或者店铺申请信息等数据入库不能做到实时性，势必会影响用户使用。因此程序需要高效完成大规模数据的索引入库过程。

2. 能迅速响应查询条件并返回结果

程序需要能够在很短的时间内响应查询请求并返回检索结果，以不超过5s为例。

* + 1. 并行性

该系统需要对于任务并行添加或者领取，对任务内容需要并行读写。在并行过程中需要利用事务恢复技术以及封锁技术保证整个系统的完整性约束以及数据的正确性。

* + 1. 可扩展性

该系统可扩展至任何供需关系的管理之中，在扩展的同时只需要进行对于用户以及任务的属性的一些额外的约束即可。

* + 1. 易用性

本系统以网页为基本平台，支持在各种终端上使用，界面交互尽量人性化。系统设计考虑到了各种用例以及实际情况，能够保证系统在各种情况下满足用户们的使用要求。

* + 1. 安全性

1.确保数据库、用户信息等安全性；

2.借助第三方支付平台（例：微信、支付宝）的支付接口来确保资金支付的安全性。

* 1. 条件和限制

暂无。

1. 总体设计
   1. 硬件结构设计

数据库体系架构：采用三层体系架构，将程序与数据分离开来；用户通过界面层操作，向Web服务器请求对数据库数据进行某种操作；服务器收到请求之后通过调用数据库管理系统对数据库中的数据进行读写等操作：

界面层：“微服务”系统网站运行界面及相关程序。

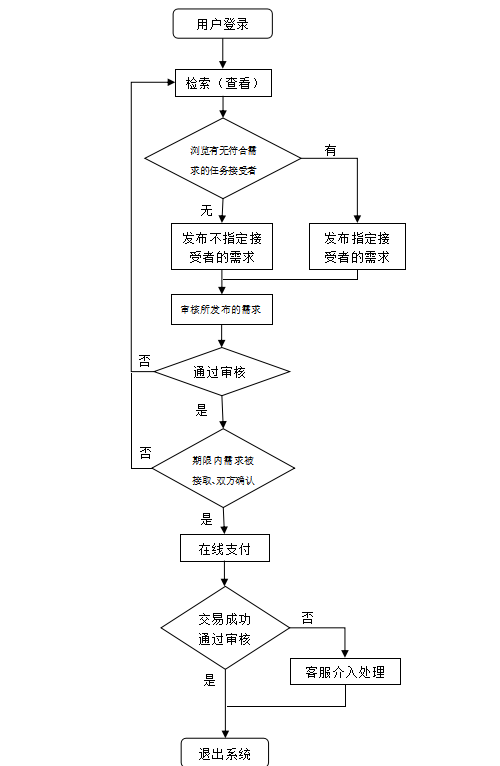
业务逻辑层：Web服务器

数据访问层：数据库及数据库管理软件（MySQL）

* 1. 软件结构设计

系统的软件由三个部分组成，分别是任务发布者子系统，服务提供者子系统，管理员子系统。下面利用流程图来说明各个层次的设计思路，并列举出各个子系统的内部模块以及功能说明。

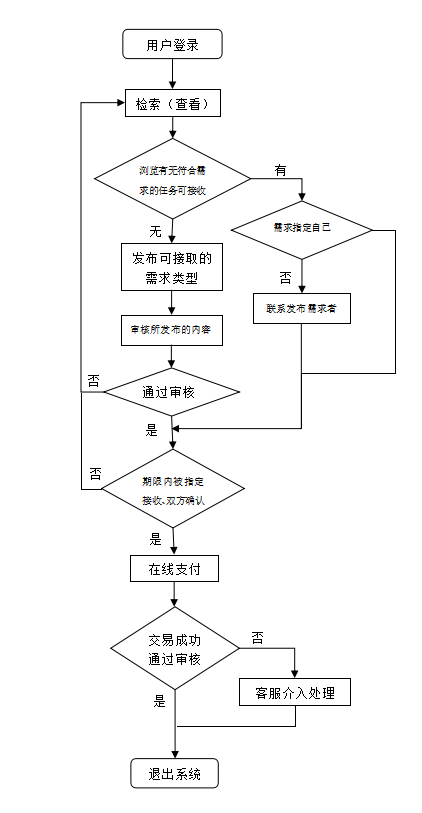
1. **任务发布者子系统**



任务发布者行为流程图

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | 模块名称 | 功能需求 | 程序ID |
| 1 | 个人信息模块 | 创建个人基本信息  修改个人基本信息  查看个人基本信息 | UserManage\_\* |
| 2 | 任务接收者查询 | 按照查询条件进行符合需求的任务接受者查询 | SenderSearch\_\* |
| 3 | 任务发布 | 发布指定或不指定接受者的任务需求 | SendWork\_\* |
| 4 | 用户/管理员交流 | 与其他用户互动：  聊天咨询或确认订单  向管理员举报/咨询等  给用户/管理员留言  订单提交 | UserTalk\_\* |
| 5 | 在线支付 | 在线支付订单 | SenderPay\_\* |

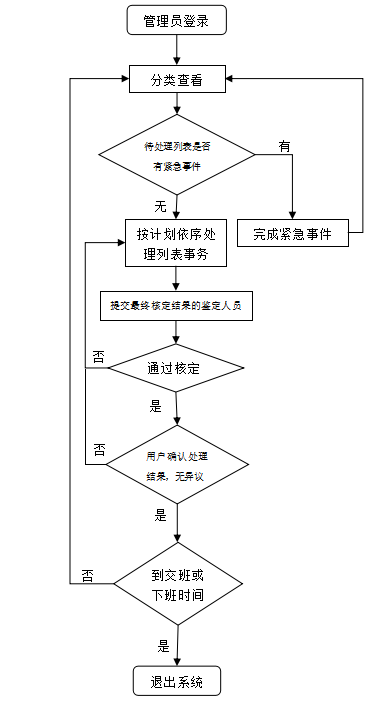
1. **服务提供者子系统**



服务提供者行为流程图

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | 模块名称 | 功能需求 | 程序ID |
| 1 | 个人信息模块 | 创建个人基本信息  修改个人基本信息  查看个人基本信息 | UserManage\_\* |
| 2 | 服务提供者查询 | 按照查询条件进行符合需求的可接收任务查询 | AccepterSearch\_\* |
| 3 | 店铺提供服务发布 | 申请店铺后或直接在店铺发布可提供服务 | AcceptWork\_\* |
| 4 | 用户/管理员交流 | 与其他用户互动：  聊天咨询或确认订单  向管理员举报/咨询等  给用户/管理员留言  订单提交 | UserTalk\_\* |
| 5 | 在线支付 | 退款/修改订单等 | AccepterPay\_\* |

1. **管理员子系统**



管理员行为流程图

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | 模块名称 | 功能需求 | 程序ID |
| 1 | 个人信息模块 | 创建个人基本信息  修改个人基本信息  查看个人基本信息 | UserManage\_\* |
| 2 | 待处理列表管理 | 添加紧急待处理事务  删除重复待处理事务  修改待处理事务类别 | ManageList\_\* |
| 3 | 待处理事务管理 | 处理待处理事务 | ManageWork\_\* |
| 4 | 待处理事务核定 | 负责鉴定的管理人员进行处理后事务的结果核定 | AssureWork\_\* |
| 5 | 用户/管理员交流 | 与用户/其他管理员互动交流 | ManageTalk\_\* |

* 1. 数据结构设计
     1. 公共数据结构设计

对程序当中所定义的全局变量、常量、全局数据结构或类当中Public数据结构进行定义和说明。

* + 1. 数据库设计

1. 管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 1 | 表名 | Manager | |
| 表描述 | 记录管理平台的管理员 | | | |
| 字段 | 类型 | | | 备注 |
| **id** | varchar(36) not null primary key | | | 管理员ID |
| **name** | varchar(10) not null unique | | | 管理员名称 |
| **password** | varchar(50) not null | | | 管理员密码 |

1. 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 2 | 表名 | User | |
| 表描述 | 记录服务平台的用户 | | | |
| 字段 | 类型 | | | 备注 |
| **id** | int primary key auto\_increment | | | 用户编码 |
| **idcard** | varchar(36) not null primary key | | | 身份证号 |
| **schoolid** | char(18) | | | 学号 |
| **name** | varchar(10) | | | 用户名 |
| **turename** | varchar(8) | | | 真实姓名 |
| **sex** | char(2) | | | 性别 |
| **addr** | nvarchar(40) not null | | | 住址 |
| **email** | varchar(40) not null | | | 用户Email地址 |
| **password** | varchar(50) not null | | | 用户密码 |
| **postalcode** | char(6) not null定长 | | | 用户头像地址 |
| **createTime** | datetime not null | | | 添加用户时间 |
| **remark** | nvarchar(100) | | | 备注 |
|  |  |  |  |  |

1. 店铺信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 3 | 表名 | Shop\_Info | |
| 表描述 | 记录注册到平台的店铺信息 | | | |
| 字段 | 类型 | | | 备注 |
| **id** | varchar(36) not null primary key | | | 店铺ID |
| **name** | varchar(255) not null unique | | | 店铺名称 |
| **url** | varchar(200) not null | | | 店铺地址 |
| **des** | nvarchar(100) | | | 店铺描述 |
| **createtime** | datetime not null | | | 店铺注册时间 |
| **accountID** | varchar(36) | | | 店铺所有者ID |
| **expireTime** | datetime null | | | 店铺的失效时间 |
|  |  |  |  |  |

1. 任务信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 4 | 表名 | Task | |
| 表描述 | 记录平台中的任务 | | | |
| 字段 | 类型 | | | 备注 |
| **id** | int identity primary key | | | 任务标识 |
| **uid** | int | | | 发布者id |
| **name** | varchar(255) not null | | | 任务内容 |
| **categoryId** | int | | | 种类id |
| **money** | double | | | 奖励金额 |

1. 交易记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 5 | 表名 | Order | |
| 表描述 | 记录平台中的任务 | | | |
| 字段 | 类型 | | | 备注 |
| **pid** | int | | | 发布人id |
| **rid** | int | | | 接收人id |
| **risshop** | int | | | 表示接收人是否是店铺 |
| **money** | double | | | 悬赏金额 |
| **status** | int | | | 标记订单是否完成/废弃 |

1. 任务类别

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 6 | 表名 | Category | |
| 表描述 | 记录平台中的任务 | | | |
| 字段 | 类型 | | | 备注 |
| **id** | int | | | id |
| **pid** | int | | | 种类父节点id |
| **isleaf** | double | | | 是否为叶节点 |
| **name** | varchar(255) | | | 名称 |
| **grade** | int | | | 级数 |

1. 服务信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 6 | 表名 | Server | |
| 表描述 | 记录平台中的任务 | | | |
| 字段 | 类型 | | | 备注 |
| **id** | int | | | id |
| **categoryid** | int | | | 种类id |
| **shopid** | int | | | 所属服务店铺id |
| **name** | varchar(255) | | | 名称 |

* 1. 接口设计

给出系统内、外部接口的设计，可分为两小节。内部接口指系统内部模块之间的调用接口，外部接口包括系统对外提供的接口以及外部用户界面的设计。可细分为三小节。

* + 1. 内部接口设计

1. 业务逻辑层——表现层

本应用的数据库采用分布式方式，所有的数据分布在各个数据中心，方便各个模块分开管理。登陆接口和账号数据库连接，通过session方式保密的传递账号密码，然后对应后段对比数据库后再穿回登陆成功失败信息。

注册接口和账号数据库连接，通过session方式保密的传递账号密码，然后用sql语句操作数据库，设置了严密的触发器来为新用户开通各种权限。

用户进行浏览服务列表、求助列表时与数据库进行连接，通过jsp传信息至前端页面。该页面对用户信息反馈给数据库。

1. 数据操作层——业务逻辑层

数据库基本操作（增删改查）存放在数据库操作接口DataBase中，供外部静态调用来执行SQL语句；对每一个表单的操作方法在相应接口中声明，比方说操作User表的所有方法声明在接口UserDAO中，当数据库发生变化时（从Mysql转变为Oracle），只需修改对应实现该接口的java类，不会影响到上层逻辑操作层。

* + 1. 外部接口设计

1. 硬件接口

服务器：服务器配置（基本要求）：双核CPU、内存2G以上、硬盘80G

网卡：10M/1000M Ethernet

客户机：

CPU：Inter Pentium E 2200或AMD Athlon 5000及以上

内存：512M以上

硬盘：8G以上

网卡：10M/1000M Ethernet

外围设备：打印机 扫描仪 绘图仪

通讯设备：传真机

1. 软件接口

服务器操作系统： WINDOWS2000/2003 Server IE4.0 SP1/NT SP4/2000/2003

SERVER SP4

数据库服务器： MySQL Server 5

WEB服务器软件：IIS6.0+

* + 1. 用户界面设计

用户主界面只要分为发布需求、查询与接受需求、我的服务记录、申请店铺、自服务、私信这几个主要的功能。

每个功能对应不同的界面。发布需求界面中利用网页的表格，用户能够输入自身的基本信息，寻求服务的内容以及提供的悬赏（在线支付功能可能需要单独的界面来支持）。

查询与接受需求中应该能够按照地点、发布者、发布时间、悬赏金额、任务关键字这些内容对任务进行搜索。点击不同任务应该能够进入其中查看信息。

“我的服务记录”栏目中能够查看本用户接受的所有任务的信息，以及本用户发布的需求的接受情况以及进行情况。

申请店铺中应该能够填入申请店铺的名称、提供的主要服务等信息，并且能够提交至服务器等待审核。

私信界面中能够与那些与自己有服务或者被服务关系的用户进行私信交流。

1. 关键技术及解决方案

哈希查找算法：

1. **基本原理**

我们使用一个下标范围比较大的数组来存储元素。可以设计一个函数（哈希函数， 也叫做散列函数），使得每个元素的关键字都与一个函数值（即数组下标）相对应，于是用这个数组单元来存储这个元素；也可以简单的理解为，按照关键字为每一个元素"分类"，然后将这个元素存储在相应"类"所对应的地方。

但是，不能够保证每个元素的关键字与函数值是一一对应的，因此极有可能出现对于不同的元素，却计算出了相同的函数值，这样就产生了"冲突"，换句话说，就是把不同的元素分在了相同的"类"之中。后面我们将看到一种解决"冲突"的简便做法。

总的来说，"直接定址"与"解决冲突"是哈希表的两大特点。

1. **函数构造**

构造函数的常用方法（下面为了叙述简洁，设 h(k) 表示关键字为 k 的元素所对应的函数值）：

a) 除余法：

选择一个适当的正整数 p ，令 h(k ) = k mod p

　　这里， p 如果选取的是比较大的素数，效果比较好。而且此法非常容易实现，因此是最常用的方法。

b) 数字选择法：

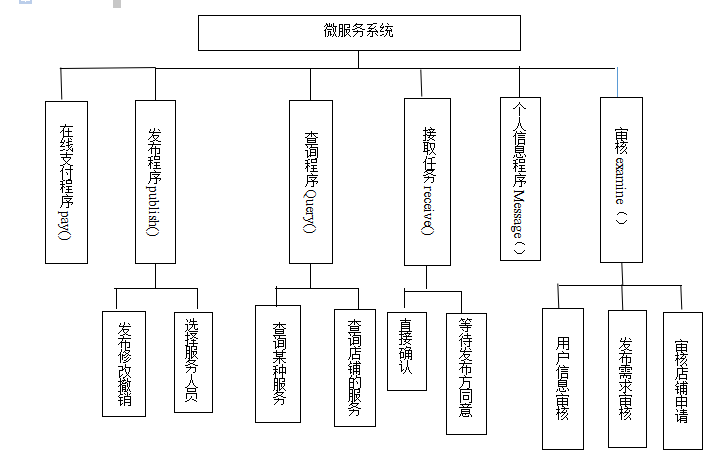
如果关键字的位数比较多，超过长整型范围而无法直接运算，可以选择其中数字分布比较均匀的若干位，所组成的新的值作为关键字或者直接作为函数值。

1. **冲突处理**

线性重新散列技术易于实现且可以较好的达到目的。令数组元素个数为 S ，则当 h(k) 已经存储了元素的时候，依次探查 (h(k)+i) mod S , i=1,2,3…… ，直到找到空的存储单元为止（或者从头到尾扫描一圈仍未发现空单元，这就是哈希表已经满了，发生了错误。当然这是可以通过扩大数组范围避免的）。

1. 详细设计与实现

系统核心业务的类图如下：



* 1. 个人信息程序Message（）设计说明
     1. 程序描述

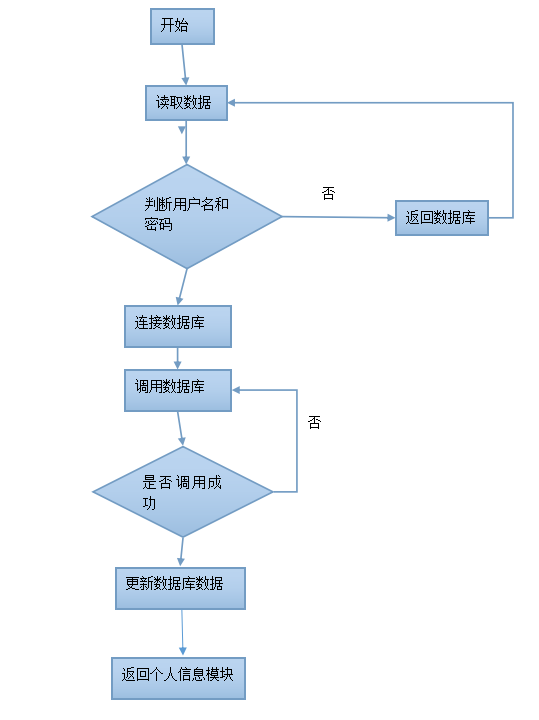
个人信息程序包括用户资料、用户隐私设置，用户安全设置，用户快捷设置等。

* + 1. 功能

|  |  |
| --- | --- |
| 功能名称 | 功能描述 |
| 修改个人资料 | 可修改用户上传的个人资料（用户名和用户身份除外） |
| 修改密码 | 需要填写正确的旧密码以及新密码和新密码确认 |
| 用户隐私设置 | 可设置性别年龄真实姓名为隐私项 |
| 用户安全设置 | 可设置手机，邮箱，微博绑定密保问题设置界面锁等 |
| 用户快捷设置 | 可设置一键查看在线任务，委托任务等。 |

* + 1. 流程逻辑

算法的流程图如下：



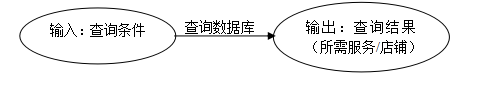
* 1. 查询程序search()设计说明
     1. 程序描述

查询程序包括查看服务分类列表、查看提供一种服务的各个店铺、查看一个店铺的各种服务、进入某个店铺发布需求等。

* + 1. 功能

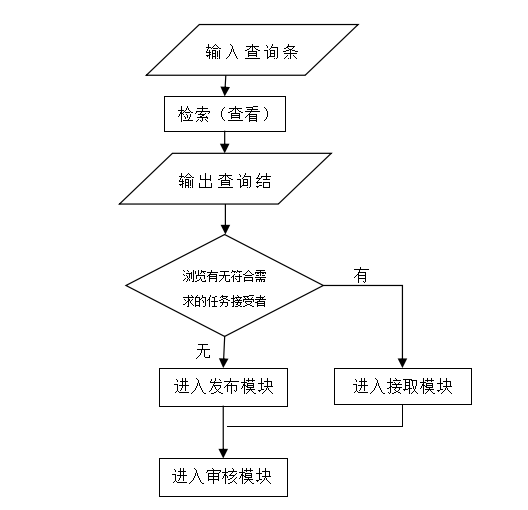
用户通过查询条件找到需要寻找的服务或店铺列表。

IPO图（即输入一处理一输出图）如下：



* + 1. 流程逻辑

查询程序的逻辑流程图如下。

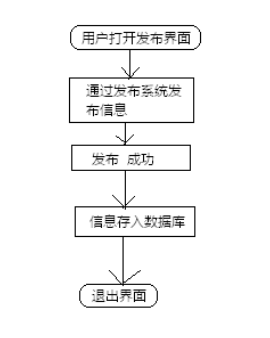


* 1. 发布程序publish()设计说明
     1. 程序描述

该程序主要是为了发布用户需求到网站上，通过发布系统，用户可以发布需求信息，此信息即可发布在查询界面上，等待服务者接收，也可发布在单个服务者的接收界面里，直接指定此服务者。

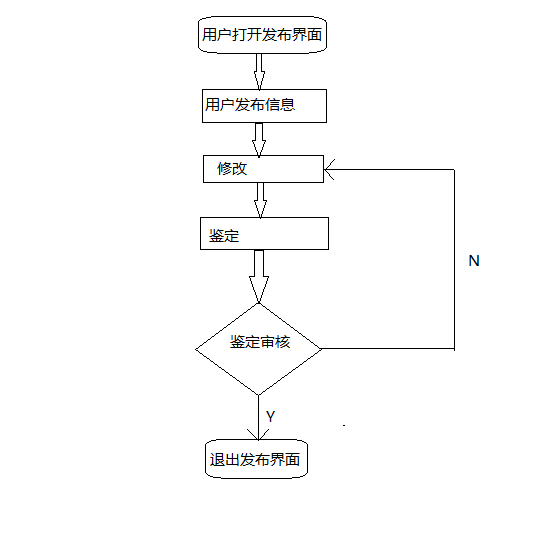
* + 1. 功能

说明该程序应具有的功能，可采用IPO图（即输入一处理一输出图）



* + 1. 流程逻辑

算法的逻辑流程图如下：



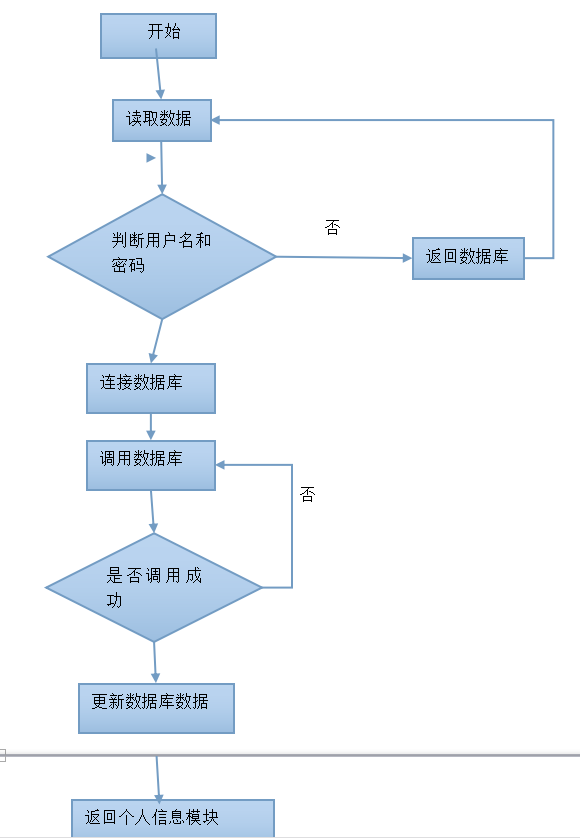
* 1. 接取任务程序receive（）设计说明
     1. 程序描述

接取任务模块涵盖了任务列表用户列表店铺列表，可通过查询结果挑选任务接取完成积累信用度和信用币。

* + 1. 功能

|  |  |
| --- | --- |
| 功能名称 | 功能描述 |
| 查看任务 | 可以通过关键字查询任务，可查看紧急任务 |
| 接受任务 | 有合适的任务可确定接受并执行 |
| 信用度管理 | 可按照用户执行任务更改用户信用 |

* + 1. 流程逻辑



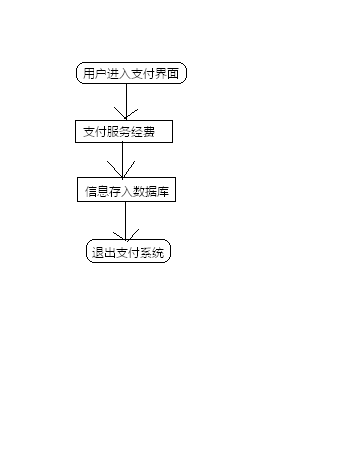
* 1. 支付程序pay（）设计说明
     1. 程序描述

在线支付程序：

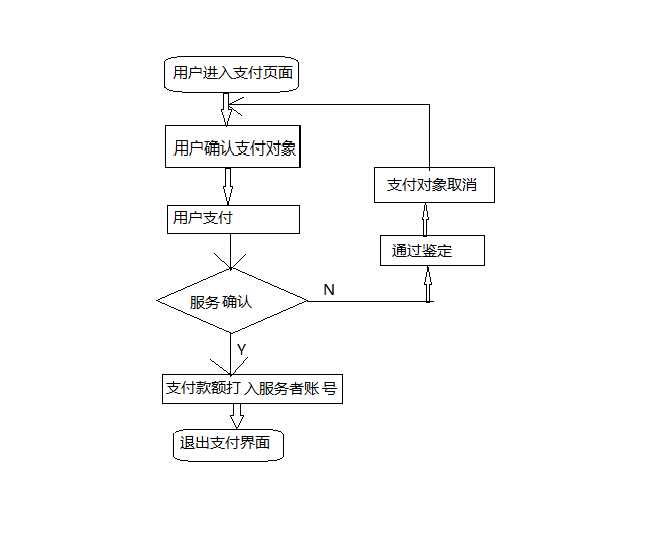
该程序主要是为了支付服务费用，创造安全支付环境。用户在指定的平台上支付服务费用，服务费用先打入第三方支付平台，在用户确认接收到服务时，费用由第三方平台打入服务者账号。

* + 1. 功能

说明该程序应具有的功能，可采用IPO图（即输入一处理一输出图）的形式。



* + 1. 流程逻辑



* 1. 审核程序examine ()设计说明
     1. 程序描述

审核包括发布需求时审核不良信息、注册时审核个人资料、店铺申请时审核信誉度，交易失败等意外情况发生时审核具体情况并调停，确保交易安全。以上多种类型的不同内容加入到待审核列表中。工作人员包括在微服鼯平台对待审核列表进行审核的审核人员、对具体情况进行调停处理的管理人员、最终核定结果的鉴定人员。

* + 1. 功能

验证系统操作员账号以及口令，处理用户提出成立服务店铺的申请，等待审核通过后加店铺加入服务店铺列表。

IPO图 3.6.2（即输入一处理一输出图）如下：

输出：审核结果

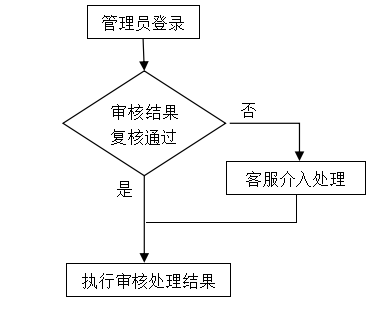
进行相应处理

进行处理/复核

输入：管理员ID以及密码、审核分类等

* + 1. 流程逻辑

用图表（例如流程图、判定表等）辅以必要的说明来表示本程序的逻辑流程。



1. 系统测试与运行效果
   1. 系统测试

暂无。

* 1. 系统运行效果

暂无。

1. 其它