四、概要设计说明书

1．引言... 2

1.1编写目的... 2

1.2项目背景... 2

1.3定义... 2

1.4参考资料... 2

2．任务概述... 3

2.1目标... 3

2.2运行环境... 3

2.3需求概述... 3

2.4条件与限制... 3

3．总体设计... 3

3.1处理流程... 3

3.2总体结构和模块外部设计... 3

3.3功能分配... 3

4．接口设计... 3

4.1外部接口... 3

4.2内部接口... 3

5．数据结构设计... 3

5.1逻辑结构设计... 3

5.2物理结构设计... 3

5.3数据结构与程序的关系... 3

6．运行设计... 3

6.1运行模块的组合... 3

6.2运行控制... 3

6.3运行时间... 3

7．出错处理设计... 3

7.1出错输出信息... 3

7.2出错处理对策... 3

8.安全保密设计... 3

9.维护设计... 3

# 1．引言

## 1.1编写目的

为了对本宝石鉴定中心业务管理系统进行规范化的分析设计，实现用户方与软件开发者双方对本系统包括硬件、功能、性能、输入输出、接口需求、警示信息、保密安全、数据与数据库、文档与法规等的要求有一个共同的理解，故撰写本文档。

本文档供项目经理、设计人员、开发人员参考。

## 1.2项目背景

本次项目为受委托单位（国人传奇宝石鉴定中心）委托，为其开发一个用于线上信息录入、管理的专属网页。该中心主要从事宝石鉴定业务，在网页功能方面需要实现主页信息总汇、鉴定收样情况显示、样品检测情况显示、任务管理、证书管理、报表功能、会员管理、系统管理和日志管理等功能。希望通过该系统对公司鉴定工作进行规范化管理和信息精准录入，达到提高工作效率的最终目的。详情如下：

项目的委托单位:国人传奇宝石鉴定中心

开发单位: 北京林业大学信息学院计创18班徐杨组

主管部门: 北京林业大学信息学院

项目名称: 国人传奇宝石鉴定中心业务管理系统

项目用户:国人传奇宝石鉴定中心

## 1.3定义

**E-R图**。也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。

**系统流程图**。是概括的描绘系统物理模型的传统工具。它的基本思想是用图形符号以黑盒子形式描绘系统里面的每个具体部件(程序、文件、数据库、表格、人工过程等），表达数据在系统各个部件之间流动的情况。

## 1.4参考资料

《软件工程导论》，张海藩（第五版），清华大学出版社；

《软件工程及软件开发应用》，施一萍，清华大学出版社；

《面向对象分析与设计（UML）》，陈志泊  清华大学出版社；

# 2．任务概述

## 2.1目标

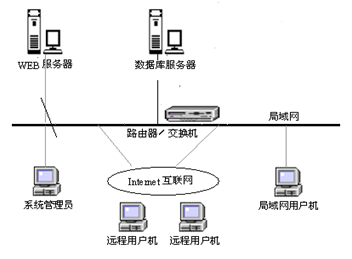
**系统开发的意图**：加强宝石鉴定业务的数字化办公，从而使得业务办理、鉴定流程等效率更高、内容更准确。

**应用目标**：实现宝石鉴赏业务数字化，加快鉴定效率，保障数据准确。

**作用范围**：为国人传奇宝石鉴定公司所用，可以分布式为分店所用。

## 2.2运行环境

**2.2.1 网络结构图**



其中的局域网用户机主要是公司内部的人员可以使用的机器，远程用户机主要是指通过互联网登录系统的人员使用的机器，可以是公司内部的人。

**2.2.2 硬件环境**

本系统的硬件环境如下：

* 客户机：普通PC
  + CPU：Pentium 133M 或更高
  + 内存：256MB以上
  + 能够运行IE5.0以上或者目前主流浏览器稳定版版本的机器
  + 分辨率：推荐使用1024\*768像素
* WEB服务器
  + CPU：Pentium 900M (推荐 Pentium 4 2.1G)
  + 内存：至少 256M（推荐 1G）
  + 硬盘：80G以上
  + 网卡：千兆
* 数据库服务器
  + CPU：P4 2.0GHz
  + 内存：1G以上
  + 硬盘：80G以上

**2.2.3 软件环境**

本系统的的软件环境如下：

* 操作系统：Unix/Linux/Windows2000或以上版本
* 数据库：SQLite Version 3.33.0
* 开发工具包：Django Version 3.1.3
* 开发环境：PyCharm Version 2020.2.3
* Web服务器：bandwagon
* 浏览器：Internet Explorer 6.0及以上 ，配置 TCP/IP 协议

（1）数据库:

对于核心数据库来说，选择一个合适的数据库系统对我们的系统运行是很重要的，选择数据库的关键因素是要考虑预计会有多少人同时访问数据库；正常工作时间的级别；用来访问数据库的应用程序的类型；运行数据库的服务器的硬件和操作系统类型；以及管理人员的专业技术水平。考虑到价格因素、易用性，我们使用SQLite作为系统后台数据库系统，服务器操作系统采用Windows 10。SQLite适用于中小规模流量的网站。SQLite可以处理多少网站流量在于网站的数据库有多大的压力. 通常来说, 如果一个网站的点击率少于100000次/天的话, SQLite是可以正常运行的。而我们的服务基于web，且访问量较少，SQLite是一个较好的选择。

（2）开发工具包：

Django是一个python web框架，主要用于进行高效率的web软件开发而衍生出来一个python组件。其功能完善、要素齐全，自带大量常用工具和框架，无须自定义、组合、增删及修改。拥有强大的数据库访问组件：Django的Model层自带数据库ORM组件；自带后台管理系统admin：只需要通过简单的配置和代码就可以实现一个完整的后台数据管理控制平台。完整的错误信息提示：在开发调试过程中如果出现运行错误或者异常，Django可以提供非常完整的错误信息帮助定位问题。因此我们选择Django作为开发工具包。

（3）开发环境：

PyCharm是使用最广泛的一种集成开发环境适用于Python编程语言。目前许多大企业都选择使用该开发环境进行开发，同时最重要是其支持Django，有了自带的HTML，CSS和 JavaScript编辑器 ，用户可以更快速的通过Django框架进行Web开发。除此之外，该开发环境还支持编码协助：其提供了一个带编码补全，代码片段，支持代码折叠和分割窗口的智能、可配置的编辑器，可帮助用户更快更轻松的完成编码工作。项目代码导航：可帮助用户即时从一个文件导航至另一个，从一个方法至其申明或者用法甚至可以穿过类的层次。若用户学会使用其提供的快捷键的话甚至能更快。代码分析：用户可使用其编码语法，错误高亮，智能检测以及一键式代码快速补全建议，使得编码更优化。Python重构：有了该功能，用户便能在项目范围内轻松进行重命名，提取方法/超类，导入域/变量/常量，移动和前推/后退重构。

## 2.3需求概述

本次项目为受委托单位（国人传奇宝石鉴定中心）委托，为其开发一个用于线上信息录入、管理的专属网页。该中心主要从事宝石鉴定业务，在网页功能方面需要实现主页信息总汇、鉴定收样情况显示、样品检测情况显示、任务管理、证书管理、报表功能、会员管理、系统管理和日志管理等功能。希望通过该系统对公司鉴定工作进行规范化管理和信息精准录入，达到提高工作效率的最终目的。

## 2.4条件与限制

1. 应具备条件：应提供相应的硬件配置，相应的软件及一定数量的技术人员和管理人员等，以及相应的开发环境。

2. 用户及合同承包者承担的工作及完成期限：由用户提出项目开发的具体要求，提供上述软硬件支持和开发环境支持，并进行管理操作，以及保证资金的到位。合同承包者提供技术人员，完成该项目的开发，并进行后续的维护工作。

3. 完成期限：本系统在三周内完成，包括最终成品和各种开发文档。

# 3．总体设计

## 3.1处理流程

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能名称 | 输入 | 处理 | 输出 | 数据来源 |
| 1. 鉴定收样 | 检测室、样品、收样人等信息 | 将收样信息添加到数据库 | 鉴定收样单 | 管理员输入和电子秤读取 |
| 1. 样品检测 | 样品检测值、项目图片、备注信息和审核结果 | 将样品监测信息添加到数据库 | 样品检测信息与结果 | 管理员输入 |
| 1. 任务管理 | 任务分配信息 | 将任务管理信息添加到数据库 | 分配后的任务管理信息 | 管理员输入 |
| 1. 证书管理 | 证书筛选条件、对证书所执行的操作 | 对证书进行查询或销毁 | 生成证书或销毁证书 | 管理员输入 |
| 1. 报表功能 | 报表查询条件 | 根据查询条件将结果上传到网站 | 上传到网站的信息 | 管理员输入 |
| 1. 会员管理 | 操作类型与对应的数据信息 | 对会员信息进行对应操作 | 更新后的会员信息 | 管理员输入 |
| 1. 系统管理 | 设置类型与对应参数信息 | 对系统设置继续对应操作 | 更新后的系统配置 | 管理员输入 |

## 3.2总体结构和模块外部设计

系统的总体结构设计遵循如下原则：

1. 系统应具有良好的适应性：能适应用户对系统的软件环境、管理内容、模式和界面的要求；
2. 系统应具有可靠性：采用成熟的技术方法和软件开发平台，以保证在以后的实际应用中安全、可靠；
3. 系统应具有较好的安全性：应提高完善的安全机制和用户权限限制机制，确保数据的受限访问；
4. 系统应具有良好的可维护性：系统应易于维护、安装；
5. 系统应具有良好的可扩展性：系统应适应未来信息化建设的要求，能方便得进行功能扩展，以建立完善的信息集成管理体系。

本系统采用struts体系结构，Struts 是一个基于模型 (Model) －视图 (View) －控制器 (Controller)(MVC) 模式的应用架构的开源框架。

|  |  |
| --- | --- |
| 功能模块 | 实现功能 |
| 鉴定收样模块 | 新建收样单，添加检验室信息、收样人信息、样品信息和内部信息 |
| 样品检测模块 | 添加样品检测值，编辑检测项目图片及备注信息，最后填写审核结果 |
| 任务管理模块 | 将宝石鉴定的订单指派给不同的检测员，可以实现单个分派操作和批量分派操作，同时收样员可以在本页选择以及成功领件的任务，使其变成任务完成状态。 |
| 证书管理模块 | 按照一定的筛选条件查询证书信息, 使证书作废, 打印输出证书信息。 |
| 报表功能模块 | 根据报表的不同状态进行查询，同时可以将查询结果上传至网站。 |
| 会员管理模块 | 添加新会员信息, 更新或删除已有的会员信息, 选中会员点击生成订单转至鉴定收样页创建新订单。 |
| 系统管理模块 | 设置管理员分组，并授予不同分组不同的管理权限。进行全局系统设置。设定不同会员等级。 |

## 3.3功能分配

数据库总领整个系统，宝石信息录入功能负责对其进行建立、增加、插入、删除、修改、合并的操作，各模块之间相互协作，功能完备，结构严谨。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 模块1 | 模块2 | 模块3 | 模块4 | 模块5 | 模块6 | 模块7 | 模块8 | 模块9 |
| 功能1：总体概况 | √ |  |  |  |  | √ |  |  |  |
| 功能2：鉴定收样 |  | √ |  |  |  |  |  |  |  |
| 功能3：样品检测 |  | √ | √ |  |  |  |  |  |  |
| 功能4：任务管理 |  | √ |  | √ |  |  |  |  |  |
| 功能5：证书管理 |  | √ | √ |  | √ |  |  |  |  |
| 功能6：报表功能 |  | √ | √ | √ | √ | √ |  |  |  |
| 功能7：会员管理 |  | √ |  |  |  |  | √ | √ |  |
| 功能8：系统管理 |  | √ |  |  |  |  | √ | √ |  |
| 功能9：日志管理 |  |  |  |  |  |  |  |  | √ |

# 4．接口设计

## 4.1外部接口

宝石鉴定中心业务管理系统是B/S结构的，它提供友好的浏览页面，利用HTTP协议交互与用户之间的操作指令、数据信息。用户各种操作都是通过浏览器来实现的，数据格式是以字符串的形式传到服务器。同时还需要USB接口与电子秤进行数据传输。

**用户界面：**

本系统仅有管理端程序。管理端主要实现鉴定收样、样品检测、任务管理、证书管理、报表功能、会员管理和系统管理功能。

主要界面设计如下：

* 鉴定收样模块
  + 新建收样单页面，添加检验室信息、收样人信息、样品信息和内部信息的集成页面
* 样品检测模块
  + 包括添加样品检测值页面，编辑检测项目图片及备注信息，最后填写审核结果页面
* 任务管理模块
  + 以列表形式展示出当前任务数据
* 证书管理模块
  + 按照一定的筛选条件查询证书信息, 使证书作废, 打印输出证书信息。
* 报表功能模块
  + 根据报表的不同状态进行查询的页面
* 会员管理模块
  + 添加新会员信息, 更新或删除已有的会员信息页面, 选中会员点击生成订单转至鉴定收样页创建新订单页面。
* 系统管理模块
  + 设置管理员分组，并授予不同分组不同的管理权限页面。进行全局系统设置。设定不同会员等级页面。

## 4.2内部接口

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传递。具体参数的结构将在数据结构设计的内容中说明。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。

# 5．数据结构设计

## 5.1逻辑结构设计

部分数据结构如表所示

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 长度 | 可否为空 | 键引用 | 字段含义 |
| identification\_id | 检测批号 | varchar | 50 | no | 主键 | 该宝石的检测批号 |
| identification\_staff | 检测员 | varchar | 50 | no |  | 检测员姓名 |
| sample\_sender | 送样员 | varchar | 50 | no |  | 送样员姓名 |
| task\_status | 任务状态 | varchar | 50 | no |  | 宝石鉴定进度 |
| verification\_ code | 验证码 | varchar | 50 | no |  | 宝石的验证码 |
| gemstone\_id | 宝石编号 | varchar | 50 | no | 主键 | 该宝石本身的编号 |
| identification\_result | 鉴定结果 | varchar | 50 | yes |  | 宝石鉴定结果 |
| gemstone\_appearance | 外观 | varchar | 50 | yes |  | 宝石外观描述 |
| gemstone\_mass | 总质量 | float |  | yes |  | 宝石的总质量 |
| gemstone\_density | 密度 | float |  | yes |  | 宝石的密度 |
| refractive index | 折射率 | float |  | yes |  | 宝石的折射率 |
| light\_ characteristic | 光性特质 | varchar | 50 | yes |  | 宝石的光性特质 |
| observation \_result | 放大观察 | varchar | 50 | yes |  | 宝石放大观察结果 |
| heavy\_metal | 重金属检测 | varchar | 50 | yes |  | 宝石中重金属含量 |

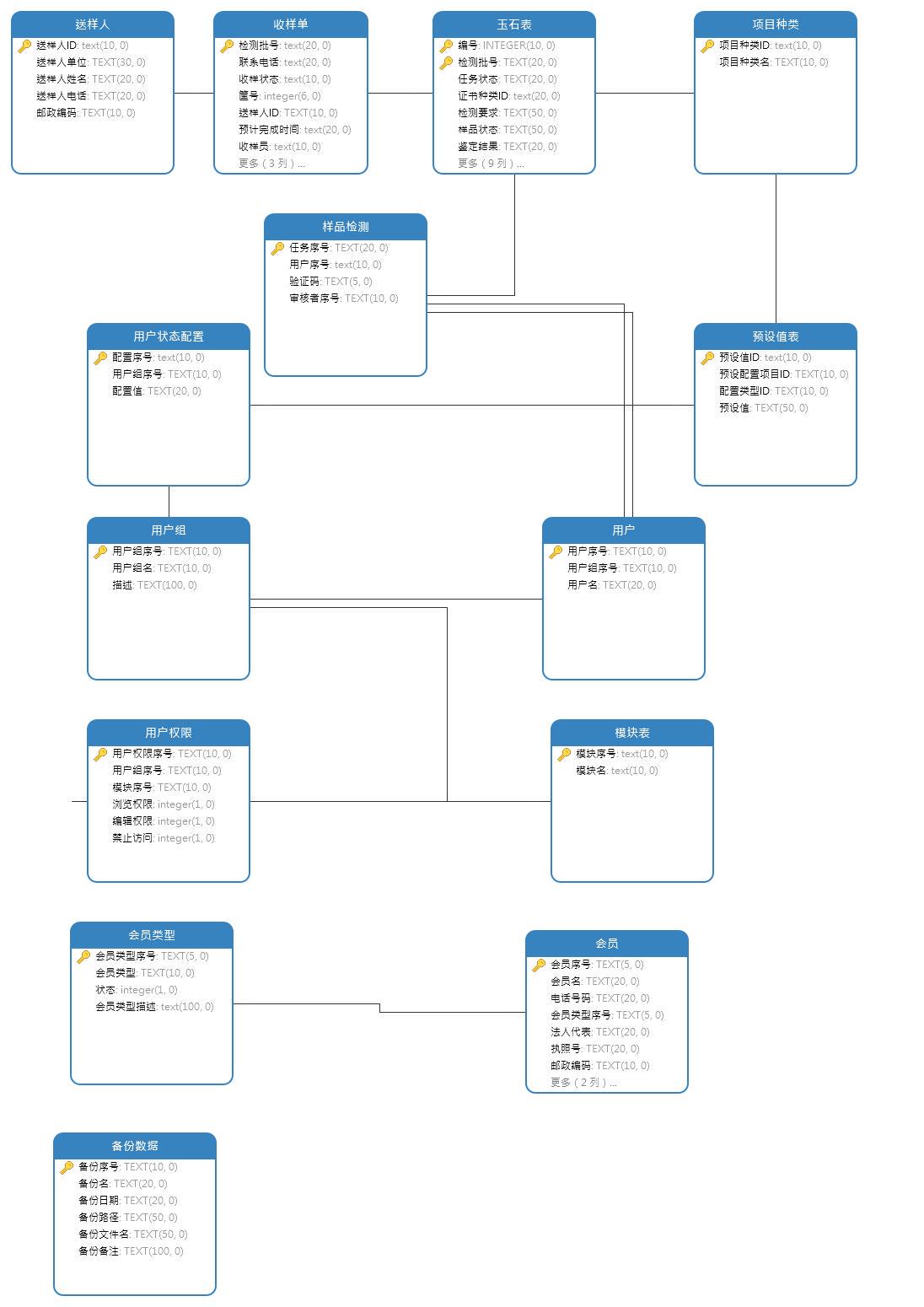
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 长度 | 可否为空 | 键引用 | 字段含义 |
| id | 序号 | long |  | 否 | 主键 | 证书在数据库中独一无二的序号。 |
| inspecter | 检测员 | varchar | 50 | 是 |  | 检测员姓名 |
| batch\_number | 检测批号 | varchar | 50 | 否 |  | 该宝石的检测批号 |
| status | 任务状态 | varchar | 50 | 否 |  | 宝石鉴定进度 |
| serial\_number | 编号 | varchar | 50 | 否 |  | 该宝石本身的编号 |
| cert\_type | 证书种类 | varchar | 50 | 否 |  | 宝石品种 |
| sample\_collectr | 收样员 | varchar | 50 | 否 |  | 收样员的姓名 |
| sample\_date | 收样日期 | Date |  | 否 |  | 收样员收到宝石的具体日期 |
| finish\_date | 预计完成日期 | Date |  | 否 |  | 收样员预计宝石完成鉴定的日期 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 长度 | 可否为空 | 键引用 | 字段含义 |
| group\_id | 用户组序号 | vchar | 5 | 否 | 主键 | 自动生成的用户组排序的序号 |
| group\_name | 用户组名 | varchar | 10 | 否 | 主键 | 用户组的名字 |
| group\_describe | 用户组介绍 | varchar | 100 | 是 |  | 用户组介绍 |
| user\_id | 用户序号 | vchar | 5 | 否 | 主键 | 自动生成的用户序号 |
| user\_name | 用户名 | varchar | 20 | 否 | 主键 | 用户的名字 |
| user\_email | 用户邮箱 | varchar | 50 | 否 |  | 用户的邮箱 |
| user\_tel | 用户电话 | vchar | 20 | 是 |  | 用户的电话 |
| user\_passwd | 用户密码 | varchar | 50 | 否 |  | 经过确认存储的用户输入密码 |
| module | 模块名 | varchar | 10 | 否 | 主键 | 各个模块的名字 |
| read\_power | 浏览权限 | bool |  | 否 |  |  |
| edit\_power | 编辑权限 | bool |  | 否 |  |  |
| foubidden\_power | 禁止访问 | bool |  | 否 |  |  |

## 5.2物理结构设计

物理结构设计过程中需要对时间效率、空间效率、维护代价和各种用户要求进行权衡，其结果可以产生多种方案，我们的具体设计方案与索引在逻辑设计方案中已有体现，是一个较优的方案作为数据库的物理结构。其中数据存储要求是数据通过SQLite存储到对应的服务器上。

## 5.3数据结构与程序的关系



# 6．运行设计

## 6.1运行模块的组合

web端在有输入时启动接收数据模块，通过各模块之间的调用，读入并对输入进行格式化。

在接收数据模块得到充分的数据时，将调用网络传输模块，将数据通过网络送到服务器，并等待接收服务器返回的信息。接收到返回信息后随即调用数据输出模块，对信息进行处理，产生相应的输出。

服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后，调用数据处理/查询模块对数据库进行访问，完成后调用网络发送模块，将信息返回浏览器。

## 6.2运行控制

运行控制将严格按照各模块间函数调用关系来实现。在各事务中心模块中，需对运行控制进行正确的判断，选择正确的运行控制路径。

在网络传送方面，浏览器在发送数据后，将等待服务器的确认收到反馈，收到后，再次等待服务器发送回答数据，然后对数据进行确认。服务器在接到数据后发送确认信号，在对数据处理、访问数据库后，将返回信息送回浏览器，并等待确认。

## 6.3运行时间

1. 响应时间：2秒以内。
2. 更新处理时间：2秒以内。
3. 数据转换与传输时间：10秒以内。
4. 查询时间：2秒以内。

# 7．出错处理设计

系统出错最严重的就是系统没有保证用户的用户名、密码、信息和资金的安全性，为了从根本上解决这个问题，首先在编写前台和后台代码时就要注意编写的严密性，不能让不怀好意的人利用保留的攻击手段侵入系统，盗用、毁坏用户的信息，造成用户的损失。

## 7.1出错输出信息

用出错归类的方式说明每一种可能出错的情况出现时，系统输出信息的形式、含义级处理的方法。

由于输入信息不符合规范（如：鉴定宝石数量为负数等），称之为软错误；由于硬件方面的错误（如：网络传输超时、硬件出错等），称之为软错误；对于一些关键的操作（如：删除操作），应该提供提示确认机制；对于数据、测试文档，都是要提供相应的保密措施设置。

同时，不同出错情况输出信息也应不同。数据库无法连接：请联系系统管理员；用户名错误：无此用户；用户密码错误：密码错误，请检查你的密码；管理员用户名错误：无此用户；管理员密码错误：密码错误，请检查你的密码；用户未登录：请您登陆后再进行操作。

## 7.2出错处理对策

|  |  |
| --- | --- |
| 运行时遇到不可恢复的系统错误 | 保证数据库完好无损 |
| 用户的输入有错误的情况 | 对于用户的输入错误应给出适当的改正提示。 |
| 对于重大的操作错误 | 采用适用故障处理界面直接操作数据库，进行处理 |

# 8.安全保密设计

对管理员的权限进行良好的管理, 确保系统安全性和防止被暴力破解。前台程序的用户数量多，地点比较分散，应限制非法用户对本系统的使用。拟采用WEB服务器端对网站安全性的实现手段。主要使用限制客户的IP地址的方法。后天管理程序的使用权控制通过页面认证表单来实现。也可以利用WEB服务器的安全认证来实现。如果对安全性的要求很高的话，可以通过在服务器、客服机中安装数字证书来提供安全性。本系统的后台权限验证功能目的是防止管理员的错误操作。

# 9.维护设计

1. 设计代码时：强调代码复用,多个组件可以很方便的以“插件”形式服务于整个系统，例如验证码的设置；
2. 使用 Django 框架，各个层次鲜明分割，专注于处理不同的事情, 各个插件能够自然地插拔；
3. URL 分派，使用正则表达式匹配 URL；
4. 使用 Django 的模板系统，利用模板系统自带的继承和包含方便修改和扩展；每个 Django App 的组织结构符合 Django 的 MTV 法则——Model（模型）+ Template（模板）+ View（视图)。
5. 表单的处理，将所有的表单抽象成类方便进行后台验证；
6. Django 在数据模型方面的设计堪称典范，列举一些闪光点：
7. 由于高度解耦的设计，可轻松切换各种关系型数据库（默认的 SQLite，可选 MySQL、PostgreSQL、Oracle 等等）
8. 强大的 ORM（Object Relation Mapping，对象关系映射）模块，使得用 Python 操作数据库非常轻松，免去了使用 SQL 的麻烦
9. 优秀的数据库迁移机制（Migration），修改数据模式（Schema）比较方便，能够适应不断变化的功能需求。