|  |  |
| --- | --- |
| **文档名称：** | **软件设计说明书** |
| **项目名称：** | **失物招领** |

|  |  |
| --- | --- |
| **团队编写：** | **202002** |
| **团队组长：** | **焦旭超** |
| **团队成员：** | **穷吉、白玛次仁、罗松** |
| **日期：** | **2020年06月05日** |
| **院系：** | **计算机科学与工程学院** |
| **专业：** | **计算机科学与技术** |

目录

[1. 引言 3](#_Toc26168)

[1.1编写目的 3](#_Toc24810)

[1.2项目背景 3](#_Toc5001)

[2.概述 4](#_Toc30201)

[2.1目标 4](#_Toc30830)

[2.2运行环境 4](#_Toc27056)

[2.3条件与限制 4](#_Toc28261)

[3.总体设计 5](#_Toc11351)

[3.1系统结构 5](#_Toc19215)

[3.2系统整体功能模块 6](#_Toc28907)

[3.3管理员功能模块 7](#_Toc13748)

[3.4用户功能模块 7](#_Toc418)

[3.5游客功能模块 7](#_Toc2305)

[4.数据库逻辑结构 8](#_Toc12696)

[4.1 E-R图： 8](#_Toc13135)

[4.2 E-R 图向关系模型的转换 8](#_Toc27238)

[5.软件重用方案 9](#_Toc17500)

[5.1需求重用方案 9](#_Toc23800)

[5.2架构设计重用方案 10](#_Toc10930)

[5.3代码重用方案 10](#_Toc7777)

[5.4项目组织结构重用方案 10](#_Toc27793)

[6.设计关键类的重点服务 11](#_Toc8249)

[6.1用户接口 11](#_Toc7373)

[6.2外部接口 11](#_Toc18910)

[6.3内部接口 11](#_Toc24051)

# 引言

## 1.1编写目的

我们团队在编写《失物招领》软件设计说明书之前对同类型产品的市场进行了大量前期调查，在与多位软件设计者和使用者进行了探讨和分析，之后由软件项目小组共同协商提出了这份软件设计说明书。这份软件设计说明书对失物招领系统软件进行了全面细致的用户需求分析，明确所要开发的软件应具有的功能、界面以及实现效果。

## 1.2项目背景

目前，失主丢失物品之后，往往苦于寻找，要不就是到处粘贴寻物启事。同时，拾主拾到物品若想要归还，也是重重麻烦。现在我国正在建设和谐社会，提高公民素质，学校也应该全面贯彻这一精神，发扬拾金不昧的传统美德。为了帮助我校失主更加方便快捷地寻回丢失物品，同时减轻失物招领处的管理员的工作负担，我们开发了的“校园失物招领管理系统”，为校师生服务，帮助大家找回失物，减少由此带来的经济和精神损失。

要求系统有效、快速、安全、可靠和无错误的完成需要的操作，并要求客户机的界面要简单明了，易于操作。

# 2.概述

## 2.1目标

本设计要实现的是失物招领系统，在设计该系统时，应尽可能的贴近用户,便于用户操作。系统在实现上应该具有如下功能：

1.系统要求用户必须输入正确的用户名和密码才能进入系统。

2.系统应该提供用户发布失物信息功能。

3.系统应提供用户发布招领信息功能。

4.系统应提供用户发布表扬信息的功能。

5.系统应提供分类查找功能，以方便用户对失物信息的查询(要实现按多种条件的查询)。

6.系统应提供增加、删除、修改用户帐户的功能。

## 2.2运行环境

|  |  |
| --- | --- |
| CPU | Pentium Ⅳ |
| 内存 | 512MB |
| 硬盘空间 | 以上硬盘剩余空间 |
| 输入设备 键盘/鼠标 |  |
| 操作系统 | Windows XP Server以上 |
| 数据库 | MySql |
| 开发环境 | Eclipse |
| 数据库工具 | Navicat |

## 2.3条件与限制

由于目前所掌握的知识有限，并且时间比较短，本失物招领系统并没有提供用户定位功能。并且对信息的保护手段简单，安全性能有待进一步完善。由于系统开发人员较少，后期的维护工作比较难以进行，希望进一步完善。

# 3.总体设计

## 3.1系统结构

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。

　　通常模型对象负责在数据库中存取数据。

View（视图）是应用程序中处理数据显示的部分。

　　通常视图是依据模型数据创建的。

Controller（控制器）是应用程序中处理用户交互的部分。

　　通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

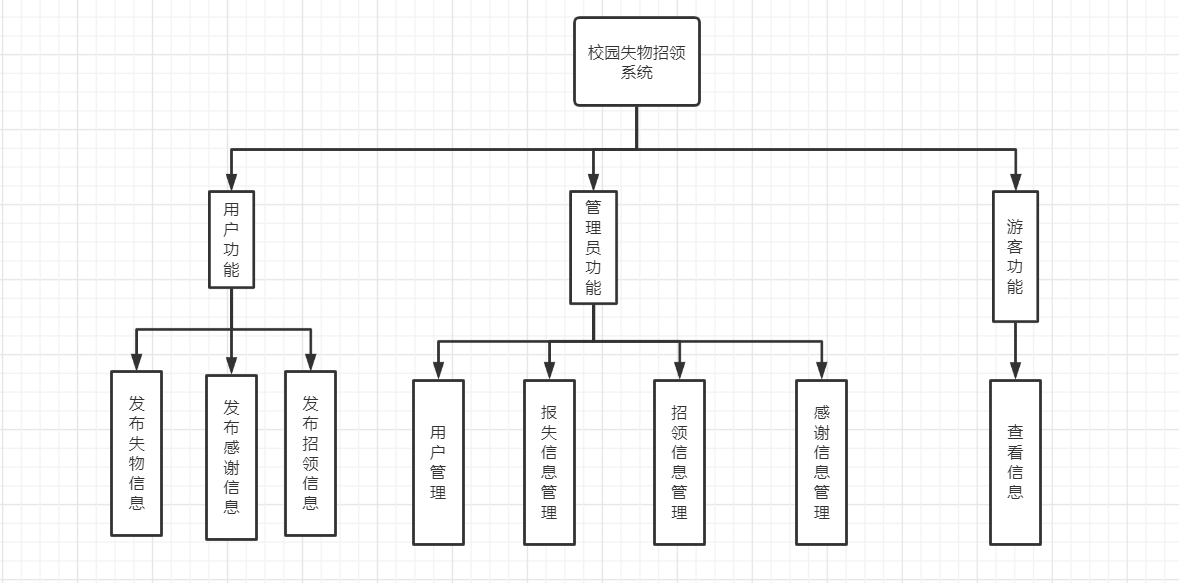
B/S结构（Browser/Server，浏览器/服务器模式），是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器，如Netscape Navigator或Internet Explorer，服务器安装SQL Server、Oracle、MYSQL等数据库。浏览器通过Web Server 同数据库进行数据交互。

第一层是浏览器，即客户端，只有简单的输入输出功能，处理极少部分的事务逻辑。由于客户不需要安装客户端，只要有浏览器就能上网浏览，所以它面向的是大范围的用户，所以界面设计得比较简单，通用。

第二层是WEB服务器，扮演着信息传送的角色。当用户想要访问数据库时，就会首先向WEB服务器发送请求，WEB服务器统一请求后会向数据库服务器发送访问数据库的请求，这个请求是以SQL语句实现的。

第三层是数据库服务器，他扮演着重要的角色，因为它存放着大量的数据。当数据库服务器收到了WEB服务器的请求后，会对SQL语句进行处理，并将返回的结果发送给WEB服务器，接下来，WEB服务器将收到的数据结果转换为HTML文本形式发送给浏览器，也就是我们打开浏览器看到的界面。

在 B/S 结构下的该失物招领系统大致可以划分为以下模块：

****

在整体设计中，我们将宿舍管理系统分为三个大的模块：用户功能模块、管理员功能模块、游客功能模块。每个模块下面又细分了很多功能来实现不同的功能需求。下面将具体进行介绍。

## 3.2系统整体功能模块

系统管理模块包括：用户登录、用户管理、管理员管理信息、游客功能四部分。

1. 用户登录：实现普通用户的登录，可以发布失物信息、招领信息以及感谢信息等功能。
2. 用户管理：实现普通用户管理，可以进行失物招领基本信息管理，包括修改，删除等操作。
3. 管理员管理信息：实现所有信息的管理，包括删除、发布、更新各种信息的功能，还能实现对用户账号的管理。
4. 游客功能：免登陆查看失物招领信息，但不能进行发布、留言等功能。

## 3.3管理员功能模块

管理员系统模块包括，用户基本信息查看及修改功能，发布信息查看修改功能，感谢信编辑修改功能。

1. 管理用户基本信息功能：查看用户基本信息，如果需要修改则进行修改，删除等操作，还可以进行高级查询。
2. 管理发布失物招领信息功能：查看修改或删除不良或不恰当的发布信息，实现平台的健康安全管理。
3. 管理好人好事功能：可官方发布感谢信息，激励大家多多拾“金”不昧。

## 3.4用户功能模块

用户功能包括发布各种信息，删除自己发布的已过期信息等操作。

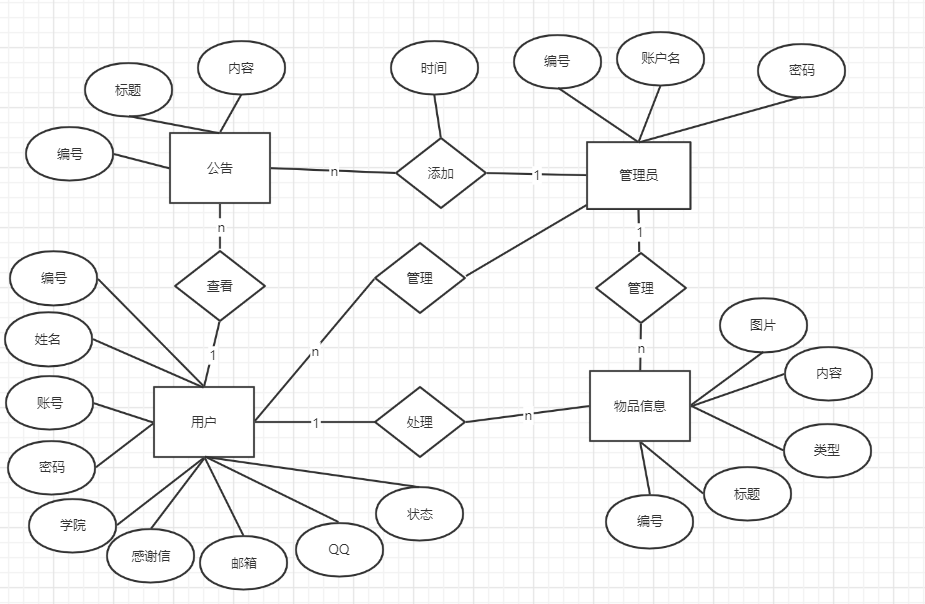
1. 查看基本信息：用户登录后可以查看自己的基本信息。
2. 发布信息：可以发布招领信息，失物信息，酬金启示信息，好人好事信息。
3. 删除过期信息：只能删除自己发布的信息。
4. 意见反馈：用户可以进行意见反馈，通过留言的形式给管理员后台留言提意见。

## 3.5游客功能模块

游客功能主要是浏览各种公开信息，具体发布或者其它操作还需注册。

# 4.数据库逻辑结构

## 4.1 E-R图：



## 4.2 E-R 图向关系模型的转换

失物表：编号、用户ID、类别、名称、大致地点、详细描述、联系方式                      主键：编号  
  
招领表：编号、用户ID、类别、名称、地点、状态、详细描述、联系方式                             主键：编号  
  
酬金表：编号、用户ID、类别、名称、大致地点、状态、详细描述、联系方式                            
主键：编号            
  
感谢信表：编号、用户ID、感谢信息                                                                    主键：编号  
  
用户表：用户ID、用户密码、用户昵称、注册时间、权限  
主键：用户ID  
  
管理员表：用户ID、用户密码、用户昵称、注册时间、权限、操作记录  
主键：用户ID

# 5.软件重用方案

实现软件复用应考虑的主要问题：

1)需求的符合性：软件项目最主要的目标是满足客户需求，在充分考虑用户共性需求和个性化需求基础土建立的系统架构，使实现软件复用的前提。

2)业务流程的可调整性

3)业务信息的可调整性

4)源代码的可扩充性

5)源代码的可移植性

## 5.1需求重用方案

失物招领系统核心业务的识别与抽取 在需求分析的总体规划指导下，我们经过深入的分析调研，发现失物招领的业务应用中都包含了：在校的注册用户，管理员类以及游客子系统，涵盖了大部分的业务流程。

在后续系统开发或者迭代的基础上，再进行需求调研时，我们可以有针对性的开展，并节约了流程梳理的时间，直接把精力集中在用户反馈等细节问题上，提高了需求调研的效率。

对于类似的使用对象来说，保持两个系统在风格上的一致性是非常重要的。它不但可以减少软件的维护、培训费用，缩短用户的适应期；而且还可以在软件开发时进行界面风格复用，减少软件开发费用。因此我们在后续开发时保证软件在风格上的一致性、操作方式上的一致性是至关重要的。

## 5.2架构设计重用方案

基于失物招领系统是招领者与寻物者信息沟通的桥梁，对实时的要求较高，因此，我们在系统设计时，采用了以统一的系统架构，我们采用MVC模式。

对于本系统，采用面向对象的方法，对于标准的重用，我们采用国家规定的软件开发规范。整体遵守代码规范，对每个人的编码都进行规范化，使得后期维护方便快捷，代码整体结构清晰。定义通用的接口以及方法，使用过程中直接调用即可。

## 5.3代码重用方案

另外，在编码过程中的一个重要复用是算法的复用。由于在函数设计时基本上每一个都提供了相似的功能，如新增、删除、修改、查询，而这些操作的算法基本上是一致的，差别只在于SQL语句的差别；所以在设计编码时，可以先设计一个共用的函数提供这些功能，在其他功能实现时可以调用这些函数，实现算法的重用。

在本系统中，在后期编码的过程中，使用调用库函数，实现代码重用，可以大大提高代码的效率。对于编写的源代码、用户界面的设计、数据等都可以重用，比如源代码的编写中，所要用到的一些前端框架，我们就可以结合自身本系统的需求进行分析，然后进行框架整合，这样就减少了我们的编码工作量，有助于加快项目进度。在UI界面中，可以重用之前的原型设计中设计好的界面，对应进行设计，因为已经对原型进行了调研和试用，所以重用原型设计来设计界面，可以保证我们最后研发出来的产品是满足用户审美的，是符合用户需求的。对于数据重用，比如数据库中表的设计，我们可以对逻辑分析中的E-R图进行加工，根据E-R图进行建表等工作，根据逻辑分析设计数据库。

## 5.4项目组织结构重用方案

在软件复用的过程中，仅仅有软件复用方法是不够的，还必须有复用的开发组织结构可以支持。后续开发、迭代或者合作中也可以采用复用的组织模式开展项目实施。基于模块独立性、层次性，保证项目开发成员工作之间的相对独立性，这样就可以把开发人员之间的通信、模块开发制约关系减到最少。同时模块独立性也比较利于配置管理工作的进行。 其实在软件复用过程当中，不仅仅通用函数可以复用，在业务层模块之间也可相互引用。但是在引用时，也应该尽量避免模块之间的交互，提高模块的内剧性、降低模块间的耦合性。在模块之间的引用协调也由协调组完成。

# 6.设计关键类的重点服务

本系统根据状态图和数据流图可筛选出本项目的关键类及相应重点服务如下：

管理员：根据用户发布的信息进行查看，通知以及删除不恰当失物招领信息。

招领用户：可以浏览失物信息，发布物品信息，等待失主认领。

失物用户：可以发布失物信息，发布感谢信，浏览招领信息，找回失物。

## 6.1用户接口

使用基于小程序的GUI，用户通过手机的点击和键盘的输入完成操作，编辑框用于用户的输入。

用浏览器登录后台，管理员以及用户通过不同的入口进行登录，只有用户名和密码对应正确，才可以登录进入。

## 6.2外部接口

通过JDBC-ODBC桥接口与My SQL数据库连接。在进行与服务器端口的连接，实现数据的交互。

## 6.3内部接口

系统基于MVC模式架构，数据库与系统之间通过JDBC进行连接，各个模块之间相互独立。