

西北師範大學

背包問題知識社區系統

系統設計說明書

版本：1.3

團隊名稱： 待宰的高羊

團隊成員： 何晨澤

高楊

阿麗米拉

謝家俊

指導教師： 代祖華

完成時間： 2022年6月19日

修改记录

* A-新增 D-删除 M-修改

修改号	日期	影响的范围	A/D/M *	概要描述
001	2022-5-13	全文	A	编撰版本 文档 1.0
002	2022-5-14	1.2	M	修改错误 内容
003	2022-6-8	2.1	A	增加系统 总体结构 图
004	2022-6-8	3.2	M	修改为数 据库逻辑 结构图
005	2022-6-8	2.7、2.8	A	增加重用 方案、重 点服务
006	2022-6-19	2	A、M	增加类说 明、调整 目录结构
007	2022-6-19	2.10	A	增加接口 说明

目录

一、引言	4
1.1 目的	4
1.2 文档约定	4
1.3 预期的读者	4
1.4 产品的范围	4
1.5 系统概述	5
1.6 引用文件	5
二、结构概述	6
2.1 功能概述	6
2.1.1 功能概述表	6
2.1.2 软件系统总体结构	6
2.2 技术概述	7
2.3 功能分析	8
2.3.1 注册登录	8
2.3.2 后台管理	8
2.3.3 资源、文档、社区、OJ	9
2.3.4 系统数据处理	10
2.4 异常处理	10
2.5 设计模式	11
2.6 安全设计	11
2.7 重用方案	12
2.8 系统类说明	12
2.9 关键类的重点服务	13
2.10 接口设计	13
2.10.1 用户接口	13
2.10.2 软件接口	14
三、数据库逻辑结构	15
3.1 数据字典	15
3.2 数据库逻辑结构	17
3.3 数据库安全	18

一、引言

1.1 目的

本文档在需求规格说明的基础上,详细描述背包问题知识社区系统着重描述各个功能模块的业务逻辑、数据库设计、实现逻辑,作为本项目的项目管理人员、系统设计人员、编码和测试人员以及与本项目相关的其他人员开展工作的基础和依据,同时为系统的测试提供参考。

1.2 文档约定

本系统的系统角色分为用户和管理员。

1.3 预期的读者

本文档的读者包括:

- 参与本项目的软件工程课程教师、助教
- 系统设计人员
- 编码和测试人员
- 用户
- 市场人员

上述读者可以通过阅读本文档对将要开发的背包问题知识社区系统有一个全面、详细的了解和认识。

1.4 产品的范围

本系统将实现用户注册、登录;设置管理员来管理及审核;资源上传、检索、审核与管理;论坛的发布、回复等基础功能;包含背包问题相关资源(典型算法源码、开源数据集、相关文献等)。

1.5 系统概述

项目名称：背包问题知识社区系统

项目开发者：何晨泽、高杨、阿丽米拉、谢家俊

项目用户：计算机类专业高校学生、背包问题学习者、算法爱好者

项目特性：本系统是一个一站式平台，用户只需访问该平台就可解决关于背包问题的所有需求；作为算法相关的平台，以后还可推广至其他的算法问题，具有较好的复用性；用户使用成本、门槛较低。

1.6 引用文件

[1] 张中峰, 李秋丹. 社区问答系统研究综述 [J]. 计算机科学, 2010, 37(11):19-23+54.

[2] 田烽楠, 王于. 求解 0-1 背包问题算法综述 [J]. 软件导刊, 2009, 8(01):59-61.

[3] 张海藩, 牟永敏. 软件工程导论 [M]. 第 6 版. 北京:清华大学出版社, 2013.

二、结构概述

2.1 功能概述

2.1.1 功能概述表

本系统的功能概述如表 2-1 所示。

表 2-1 系统功能概述

功能	具体描述	权限
注册登录	首次使用本系统的用户可注册账户；已拥有账户的管理员和用户在通过系统认证后可进入系统。	用户、管理员
后台管理	管理员登陆后进入，可管理用户信息及各类资源、文档、博文等。	管理员
资源	可在此上传、浏览、下载各类资源。	用户、管理员
文档	可在此上传、浏览、下载各类文档。	用户、管理员
社区	可在此浏览所有发表的博文。	用户、管理员
OJ	可在此选择题目进行浏览、练习。	用户、管理员
消息	可在此接收系统消息、与其他用户进行交流。	用户、管理员
设置	可在此修改个人信息。	用户、管理员
撰写博文	可在此撰写、保存、发布博文。	用户、管理员

2.1.2 软件系统总体结构

采用 MVC 设计模式，设计软件系统总体结构，如图 2-1 所示。

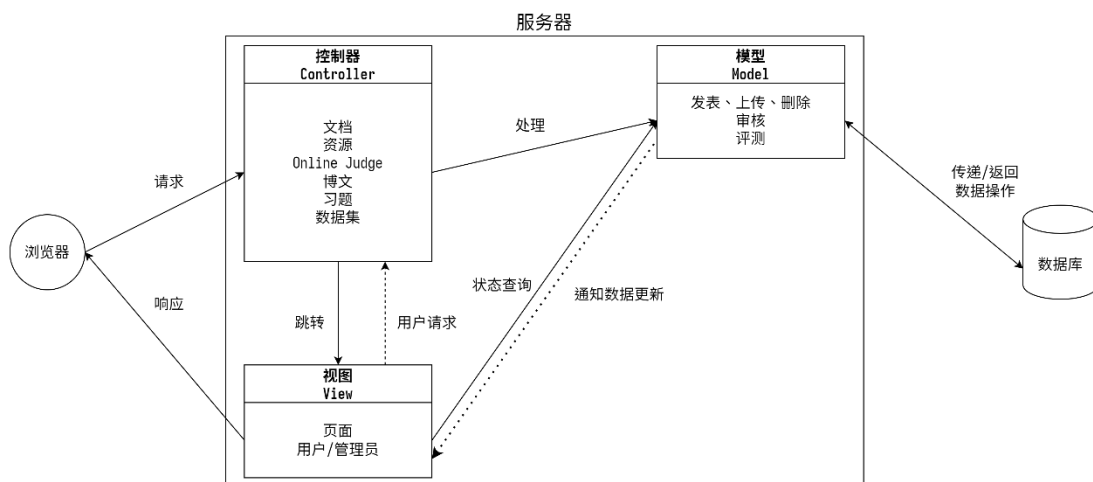


图 2-1 系统总体结构

2.2 技术概述

系统将主要使用 Java 开发，采取前后端分离方式。后端以 Spring Boot、MyBatis 等框架为基础，前端采用 Layui 或 Flarum 框架为基础。

作为成熟的体系框架，Spring Boot 能够极大的优化开发流程，简化环境配置过程。如图 2-2 所示。

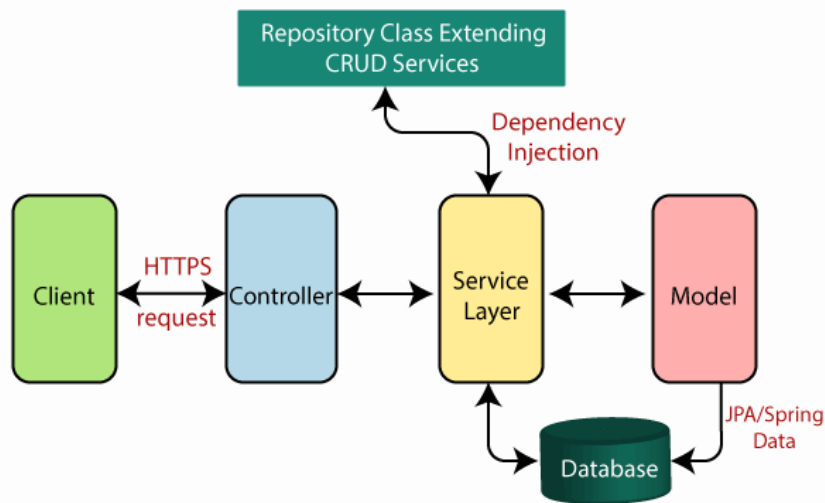


图 2-2 Spring Boot 体系结构

同时，前端采用成熟的 Layui 及 Flarum 框架，具有轻量、高效的特点，还可更贴近用户使用需求。其内容丰富，包含众多组件，从核心代码到 API 都非

常适合界面的快速开发。并且更多面向于后端开发者，拥有自己的模式。

2.3 功能分析

2.3.1 注册登录

登录：用户可选择账户密码登录、手机验证码登录以及二维码扫描登陆的登录方式；

注册：注册时，用户需要提供手机号、收到的验证码、密码，并同意用户协议。

在登陆时，若为普通用户，页面加载至系统首页；若为管理员，页面加载至管理页面。注册功能的数据流向如图 2-3 所示。

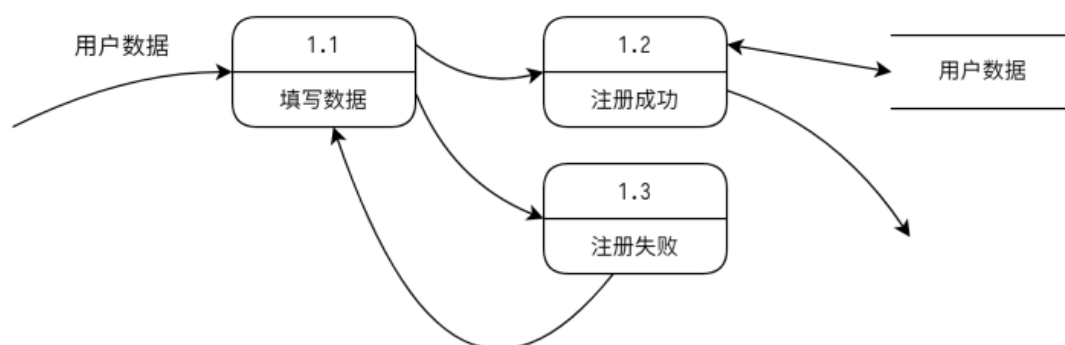


图 2-3 注册功能数据流向

2.3.2 后台管理

管理员在后台管理可对用户信息、文档、资源、习题等各类信息进行管理、审核、删除等操作。如图 2-4 所示。

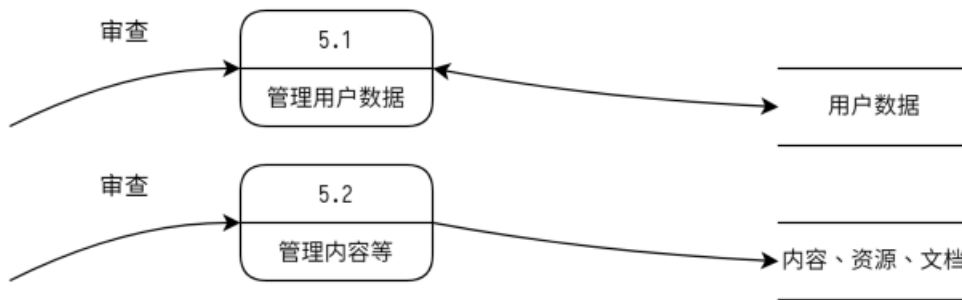


图 2-4 后台管理数据流向

2.3.3 资源、文档、社区、OJ

资源：该功能向用户提供资源搜索、推荐、下载，同时用户可在经管理员审核后上传资源；

文档：与资源功能类似，用户同样可以对文档进行搜索、下载等操作，对文档的上传同样需要经过管理员的审核；

社区：用户可在此阅读、回复其他用户发表的博文，同时也可查看自己的博文内容；

OJ：用户可在此进行练习，采用预置的数据集，对用户的代码进行测评。

如图 2-5、图 2-6、图 2-7 所示，为上述四个功能的数据流向。

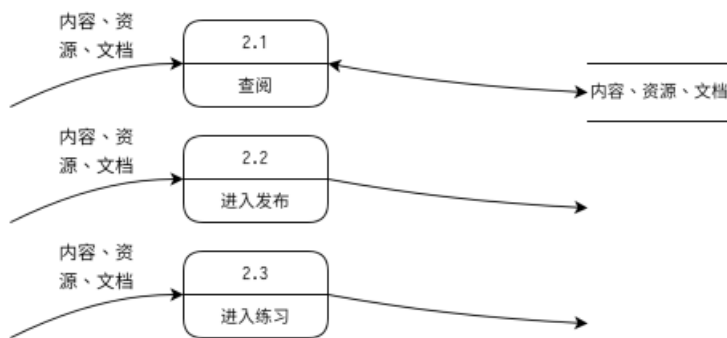


图 2-5 资源、文档、社区、OJ 总体数据流向

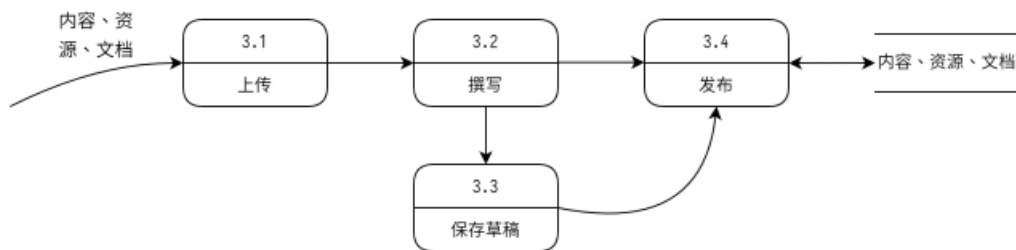


图 2-6 发布资源、文档、博文数据流向

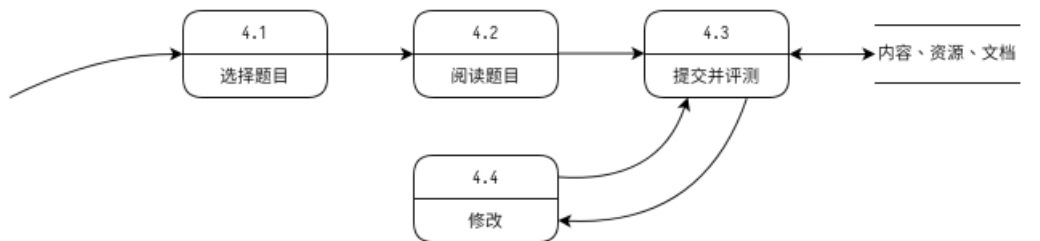


图 2-7 OJ 数据流向

2.3.4 系统数据处理

各功能间数据流向如图 2-8 所示。

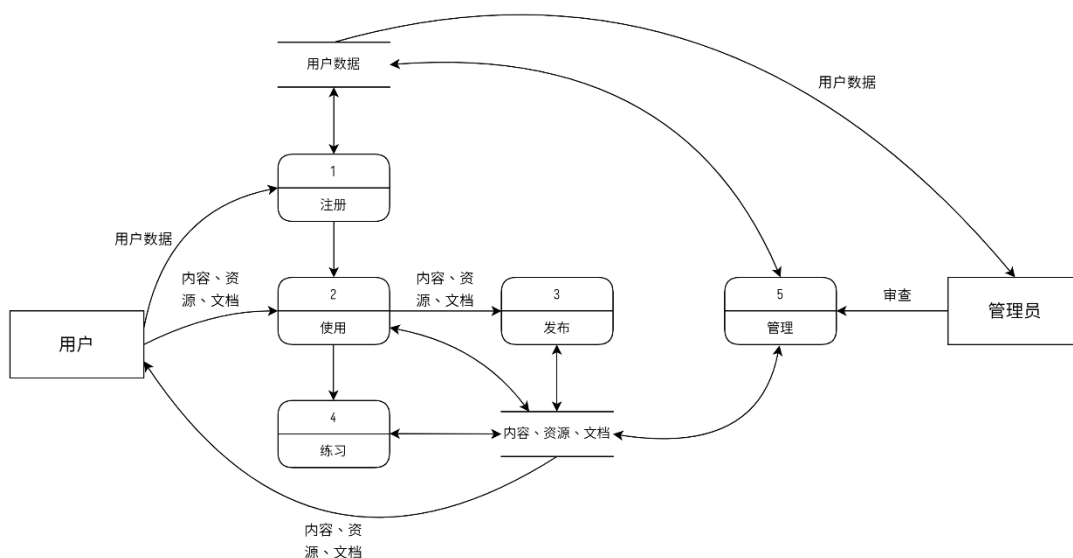


图 2-8 总体数据流向

2.4 异常处理

异常处理（又称为错误处理）是编程语言或计算机硬件里的一种机制，用于处理软件或信息系统中出现的异常状况（即超出程序正常执行流程的某些特殊条件）。其功能提供了处理程序运行时出现的任何意外或异常情况的方法。本系统的异常处理机制如下：

- (1) 在应用程序遇到异常情况（如被零除情况或内存不足警告）时，就会

产生异常。发生异常时，控制流立即跳转到关联的异常处理程序（如果存在）。

(2) 如果给定异常没有异常处理程序，则程序将停止执行，并显示一条错误信息。

(3) 可能导致异常的操作通过 try 关键字来执行。

(4) 程序可以使用 throw 关键字显式地引发异常。

(5) 异常对象包含有关错误的详细信息，其中包括调用堆栈的状态以及有关错误的文本说明。

(6) 即使引发了异常，finally 块中的代码也会执行，从而使程序可以释放资源。

(7) 异常处理使用 try、catch 和 finally 关键字来尝试可能未成功的操作，处理失败，以及在事后清理资源。异常处理通常是防止未知错误产生所采取的处理措施。

异常由公共语言运行库(CLR)、第三方库或使用 throw 关键字的应用程序代码生成，也可以用自定义的异常处理类来扩展 php 内置的异常处理类。

2.5 设计模式

本系统中使用 MVC 的设计模式，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。

2.6 安全设计

由于在网络环境下，任何用户对任何资源包括硬件和软件资源的共享，所以必须通过制定相应的安全策略来防止非法访问者访问数据资源，对数据资源的存储以及传输进行安全性保护。在背包问题知识社区系统中，参考 OSI 的七层协议，从网络级安全、传输级安全、系统级安全和应用级安全等几方面进行考虑，主要遵循下面的设计原则：

(1) 标识与确认：任何用户访问系统资源，必须得到系统的身份认证以及身份标识，如用户的数据证书、用户号码、密码。当用户信息与确认信息一致时，才能获准访问系统。

(2) 授权：对系统资源，包括程序、数据文件、数据库等，根据其特性定义其保护等级；对不同的用户，规定不同的访问资源权限，系统将根据用户权限，授予其不同等级的系统资源的权限。

(3) 日志：为了保护数据资源的安全，在系统中对所保护的资源进行任何存取操作，都做相应的记录，形成日志存档，完成基本的审计功能。

(4) 加密：为了保护数据资源的安全，在系统中对在网络中传输的信息必须经过高强度的加密处理来保证数据的安全性。

2.7 重用方案

软件重用方案主要是在源代码级，通过类的继承、调用来实现。其实可重用的范围是很大的，如设计重用，测试用例的重用，可运行的代码的重用等。将来会扩大重用的粒度，在框架基础上，进一步根据我们项目本身的软硬件环境定制出一个适用于我们系统的框架，将可以极大地提高软件的开发和维护的效率。

1. 页面：

页面均在一个基础的框架增加内容，编写时首先调用框架，进行重用。

2. 接口：

Spring Boot 框架中，所提供接口均可进行重用；对上传、删除等操作将其单独封装，进行重用。

3. 控制器：

控制器负责调用各项功能，在进行操作时，均需对其进行重用。

2.8 系统类说明

UML (Unified Modeling Language)，统一建模语言，是一种为面向对象系统的产品进行说明、可视化和编制文档的一种标准语言，是非专利的第三代建模和规约语言。

我们使用 UML 建立问题域对象模型，以类图标识系统中类的关系，如图 2-9

所示。

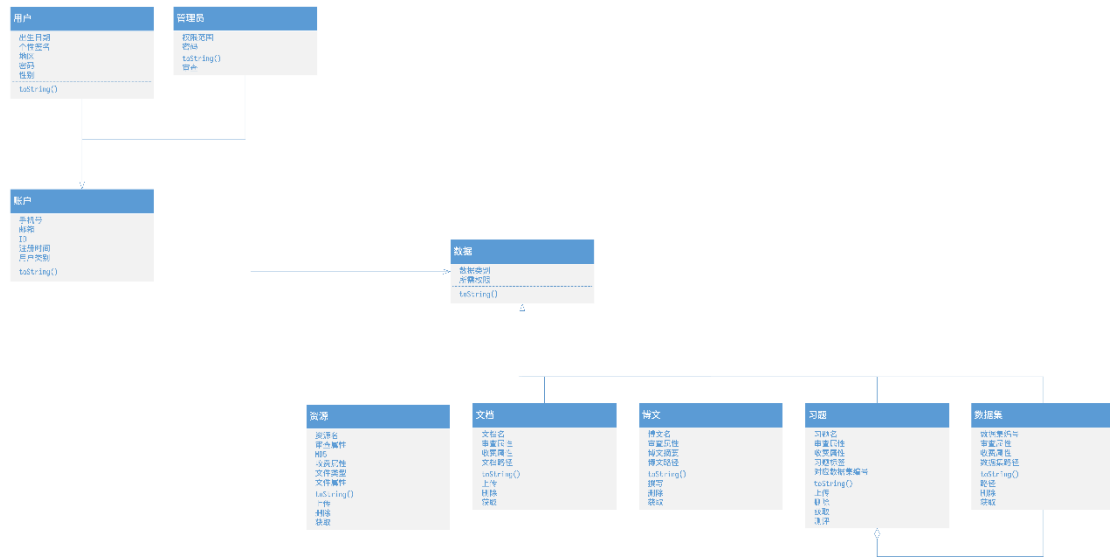


图 2-9 类图

2.9 关键类的重点服务

1. 账户类：

可分为用户及管理员，主要包含了账户的主要信息，并且继承该类的用户类和管理员类分别包含了其相应角色的信息，如权限范围等。

2. 控制器类：

该类负责对操作进行转发，对数据进行访问、查询、修改等操作。

3. 数据类：

该类负责了系统的主要操作，如文档、资源的上传或删除等，用户的操作通过控制器转发后，由该类进行操作和具体的转发。并且继承该类的资源类、文档类等类负责了其对应模块的实际操作。

2.10 接口设计

2.10.1 用户接口

本项目产品通过 PC 机进行运行、操作，对单条数据导入、多条数据导入、

数据分析、操作记录通过按钮进行单击即可。因为本项目中默认情况下，所有的用户默认不需要登录，所有的数据进行公开，因此，我们直接访问、进行操作即可。

本系统中采用 MVC 的设计架构，因此，采用 mapper 层、service 层、controller 层三层，三层之间通过接口调用，这是采用了 IOC 的设计理念，同时，采用注解的方式进行对 Bean 的注册。本系统中采用了 MyBatisPlus 的设计，因此，mapper 层和 service 层只需要实现接口就可以了，不需要定义任何的方法和接口函数。而在 controller 层，主要处理两个方面的内容：一个方面是路由转发，另外一个方面是数据应答。

2.10.2 软件接口

本系统中使用 MySQL，因此，需要使用 3306 端口；本系统中使用浏览器，使用 8080 端口和 80 端口，需使用较为完整支持 HTML5 标准的浏览器。

三、数据库逻辑结构

3.1 数据字典

数据项条目，用于标识实体。数据字典是数据库的重要部分，它存放有数据库所用的有关信息，对用户来说是一组只读的表。它是关于数据信息的集合。它是数据流图中所有要素严格定义的场所，这些要素包括数据流、数据流的组成、文件、加工小说明及其他应进入字典的一切数据，其中每个要素对应数据字典中的一项条目。各数据表项如下表所示。

(1) 用户表 (user)

表 3-1 用户表 (user)

字段名	数据类型	默认值	非空约束	自动递增	备注
user_id	VARCHAR(50)		NO		用户名
user_cnt	BIGINT		NO	YES	用户计数
user_type	TINYINT(1)	0	NO		0: 用户; 1: 管理员
user_phone	VARCHAR(11)		NO		手机号, 需经过正确性验证
user_gender	TINYINT(1)	NULL	YES		0: 男; 1: 女, 可为空
user_passw	VARCHAR(20)		NO		密码, 加密后存储
user_birth	DATA	NULL	YES		出生日期, 可为空
user_loac	VARCHAR(50)	NULL	YES		地区, 可为空
user_sign	VARCHAR(100)	NULL	YES		个性签名, 可为空

(2) 资源表 (res)

表 3-2 资源表 (res)

字段名	数据类型	默认值	非空约束	自动递增	备注
res_id	VARCHAR(50)		NO		资源名
res_cnt	BIGINT		NO	YES	资源计数
res_audit	TINYINT(1)	0	NO		0: 未审查; 1: 已审查
res_MD5	VARCHAR(32)		NO		MD5码
res_paid	TINYINT(1)	0	NO		0: 免费; 1: 收费
res_type	VARCHAR(10)		NO		文件类型
res_path	VARCHAR(100)		NO		文件路径

(3) 文档表 (doc)

表 3-3 文档表 (doc)

字段名	数据类型	默认值	非空约束	自动递增	备注
doc_id	VARCHAR(50)		NO		文档名
doc_cnt	BIGINT		NO	YES	文档计数
doc_audit	TINYINT(1)	0	NO		0: 未审查; 1: 已审查
doc_paid	TINYINT(1)	0	NO		0: 免费; 1: 收费
doc_path	VARCHAR(100)		NO		文档路径

(4) 习题表 (oj)

表 3-4 习题表 (oj)

字段名	数据类型	默认值	非空约束	自动递增	备注
oj_id	VARCHAR(50)		NO		习题名
oj_cnt	BIGINT		NO	YES	习题计数
oj_audit	TINYINT(1)	0	NO		0: 未审查; 1: 已审查
oj_paid	TINYINT(1)	0	NO		0: 免费; 1: 收费
oj_tag	VARCHAR(100)		NO		习题标签
oj_data	VARCHAR(10)		NO		对应数据集编号

(5) 数据集表 (data)

表 3-5 数据集表 (data)

字段名	数据类型	默认值	非空约束	自动递增	备注
data_id	VARCHAR(50)		NO		数据集编号
data_audit	TINYINT(1)	0	NO		0: 未审查; 1: 已审查
data_paid	TINYINT(1)	0	NO		0: 免费; 1: 收费
data_path	VARCHAR(100)		NO		数据集路径

(6) 博文表 (blog)

表 3-6 博文表 (blog)

字段名	数据类型	默认值	非空约束	自动递增	备注
blog_id	VARCHAR(50)		NO		博文名
blog_cnt	BIGINT		NO	YES	博文计数
blog_audit	TINYINT(1)	0	NO		0: 未审查; 1: 已审查
blog_abst	VARCHAR(100)		YES		博文摘要, 可为空
blog_path	VARCHAR(100)		NO		博文路径

3.2 数据库逻辑结构

E-R 图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, 修改 E-R 图, 得出数据库逻辑结构。如图 3-1 所示。

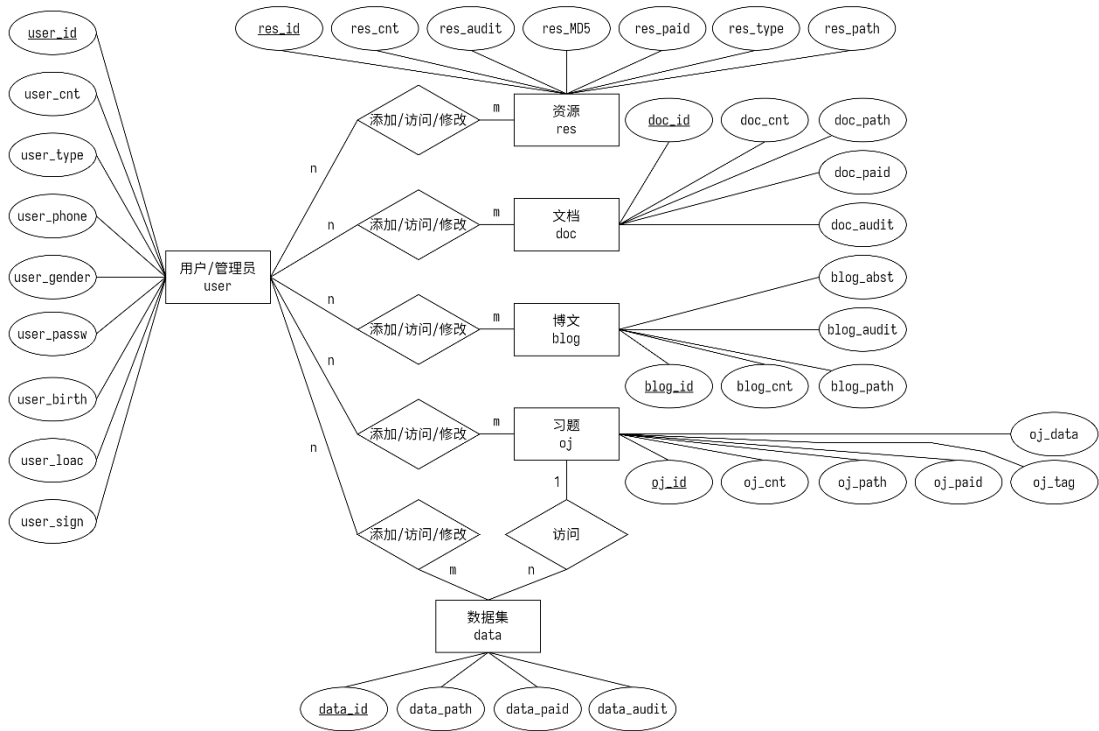


图 3-1 数据库逻辑结构图

3.3 数据库安全

数据库系统是整个系统的核心，是所有业务管理数据以及清算数据等数据存放的中心。数据库的安全直接关系到整个系统的安全。在本系统中对此考虑如下：

- (1) 数据库管理员 (SA) 的密码应由专人负责，密码应该定期变换。
- (2) 客户端程序连接数据库的用户绝对不能使用数据库管理员的超级用户身份。
- (3) 客户端程序连接数据库的用户在数据库中必须对其进行严格的权限管理，控制对数据库中每个对象的读写权限。
- (4) 利用数据库的审计功能，以对用户的某些操作进行记录。
- (5) 充分使用视图以及存储过程，保护基础数据表。
- (6) 对于不同的应用系统应建立不同的数据库用户，分配不同的权限。