# 《二手交易平台系统》设计说明书

**1 引言**

**1.1 编写目的**

本文档旨在呈现《校园二手交易平台系统》的设计成果，涵盖系统结构设计与构件设计两大部分。系统设计部分详细介绍了系统的基本处理流程、程序系统的组织结构、模块划分、功能分配、运行设计及出错处理设计等内容，为后续的详细设计工作奠定了坚实的基础。此外，本文档也将作为产品实现、特性要求和控制的重要依据。构件设计部分则聚焦于类设计和交互设计，为编码和测试提供了具体的指导。本文档的目标读者包括但不限于：

1. 对相关业务技术及总体方案作出决策的管理人员和质量管理人员；
2. 参与本说明书评审和确认的相关业务、技术人员；
3. 参与数据库设计、测试设计阶段工作的全体设计人员；
4. 参与编码工作的全体人员。

**1.2 背景**

* **系统名称**：校园二手交易平台系统
* **任务提出者**：闫礼贤
* **开发者**：闫礼贤、王凯歌、游耀朋
* **用户群体**：全体校园内的学生、教职工及校友
* **运行环境**：Windows系统

**1.3 定义**

* **MYSQL**：数据库管理软件
* **WINDOWS11**：开发环境
* **IDEA**：开发所需的编辑器
* **Java**：开发所需的语言
* **Windows**：Windows操作系统

**1.4 参考资料**

1. 孙钰坤等. 大学生闲置物品交易与校园可持续公益网络平台研究[J]. 老字号品牌营销, 2022, 12(06): 180-182.
2. 赵天齐, 袁毅贤. 探讨如何构建循环经济视角下大学校园旧物回收利用模式[J]. 商讯, 2021, 39(29): 163-165.
3. 严晓君. 循环经济视角下大学校园旧物回收利用模式构建研究[J]. 营销界, 2020, 23(43): 80-81.
4. 张雪青霞. 废旧物品升级利用再设计的探索与实践[D]. 四川美术学院硕士学位论文, 2020: 5-20.
5. 吴子沛. 面向垃圾分类的瓶罐识别分类系统[D]. 河北工程大学硕士学位论文, 2020: 5-18.
6. Wisakha Phoochinda, Saraporn Kriyapak. Electronic Waste Recycling Business: Solution, Choice, Survival[J]. IJSDP, 2021, 16(4): 4-20.
7. 费悦. 循环经济原则下的商品包装法律问题研究[D]. 中国地质大学(北京) 硕士学位论文, 2020: 3-15.
8. Castro Camila Goncalves, Trevisan Adriana Hofmann, Pigosso Daniela C. A., Mascarenhas Janaina. The rebound effect of circular economy: Definitions, mechanisms and a research agenda[J]. Journal of Cleaner Production, 2022, 345(26): 1-18.
9. 隋海利, 孟丹丹. 二手交易平台“闲鱼”发展现状及问题分析[J]. 营销界, 2020, 23(22): 13-21.
10. 刘肖. 基于JSP+MySQL的供电管理系统的设计与实现[D]. 电子科技大学硕士学位论文, 2021: 2-15.
11. 王一竹. 基于MySQL的图书馆信息管理系统设计[J]. 电脑编程技巧与维护, 2022, 29(01): 93-116.
12. 徐彩云主编. MySQL数据库实用教程[M]. 武汉: 华中科技大学出版社, 2021: 67-68.
13. 张经伟. 基于SpringBoot的高校毕业生离校系统设计与实现[D]. 西安电子科技大学硕士学位论文, 2020: 7-15.
14. 钟林森, 罗剑. Spring Boot企业级项目开发从入门到精通[M]. 武汉: 华中科技大学出版社, 2020: 55-57.
15. 欧超权. 基于SpringBoot技术的M公司协同办公平台的设计与实现[D]. 电子科技大学硕士学位论文, 2020: 7-10.
16. 赖佳星. 基于Bootstrap和JavaScript技术设计与实现简易Web计算器[J]. 科技资讯, 2022, 20(01): 13-16.

**2 总体设计**

**2.1 系统环境架构**

**该系统的参与者主要包括全体校园内的学生和教职工。为了实现高效的信息管理和交易流程，二手交易平台系统集成了多个子系统和模块，具体如下：**

* **用户管理子系统：负责用户信息的有效管理，包括注册、登录及个人信息的修改等操作。**
* **物品管理子系统：支持物品信息的全面管理，涵盖物品的发布、浏览、搜索及购买等功能。**
* **系统设置模块：提供系统运行所需的参数和配置，便于调整运行规则或参数。**
* **新闻公告模块：用于发布系统相关的新闻和公告，确保用户能够及时了解最新动态。**

**系统环境架构如图2-1所示，展示了各子系统和模块之间的交互方式。**

图示

描述已自动生成

**图2-1 系统环境架构**

**2.2 系统结构原型**

**该系统的核心要素及其相互关系如下：**

* **管理员：拥有最高权限，可执行注册、登录、查询等管理操作。**
* **用户信息：包含用户的个人信息，例如姓名、电话号码等。**
* **二手交易平台系统：作为系统核心，支持用户上传和购买二手物品。**
* **物品信息：包括物品的名称、描述、价格等详细信息。**
* **日志信息：记录系统运行过程中的关键操作，如用户的活动记录等。**

**管理员通过用户信息模块管理用户，并通过二手交易平台系统管理交易。用户信息模块和物品信息模块共同支撑二手交易平台系统的功能实现，而日志信息模块则负责监控和分析系统状态。系统结构原型如图2-2所示。**

图示

描述已自动生成

**图2-2 系统结构原型**

**2.3 系统构件关系**

**系统主要由以下构件组成，各构件之间通过特定接口进行通信，确保系统的正常运行：**

* **用户交互构件：负责用户界面的展示，包括商品列表和详细信息的显示。**
* **商品浏览构件：提供商品的浏览功能，支持用户查看商品详情。**
* **商品评价构件：允许用户对已购买的商品进行评价。**
* **商品购买构件：处理商品的购买流程，包括支付、订单生成等。**
* **数据库操作构件：负责与数据库的交互，实现数据的读取和写入。**
* **管理员交互构件：提供管理员界面，支持商品管理、用户管理及数据维护。**
* **商品举报构件：接收并处理用户的商品举报请求。**
* **商品查询构件：支持用户根据关键词或类别查找商品。**

**各构件之间的关系如图2-3所示，清晰地展示了它们的协作机制。**

图示

描述已自动生成

**图2-3 构件关系图**

**2.4 功能结构设计**

**二手交易系统的功能结构设计如下，分别展示了用户和管理员的功能模块：**

图示, 箱线图

描述已自动生成

**图2-4 用户结构图**

图示

描述已自动生成

**图2-5 管理员结构图**

**2.5 功能需求与系统模块的关系**

**下表列出了功能需求与系统模块之间的对应关系，帮助理解每个模块的具体职责：**

| **需求** | **功能模块** |
| --- | --- |
| **登录** | **用户管理模块、权限管理模块、物品管理模块** |
| **退出** | **用户管理模块** |
| **找回密码** | **用户管理模块、权限管理模块** |
| **用户添加** | **用户管理模块、权限管理模块** |
| **用户查询** | **用户管理模块、权限管理模块** |
| **用户修改** | **用户管理模块、权限管理模块** |
| **用户删除** | **用户管理模块、权限管理模块** |
| **权限添加** | **权限管理模块** |
| **权限查询** | **权限管理模块** |
| **权限修改** | **权限管理模块** |
| **权限删除** | **权限管理模块** |
| **物品信息添加** | **物品管理模块** |
| **物品信息查询** | **物品管理模块** |
| **物品信息修改** | **物品管理模块** |
| **物品信息删除** | **物品管理模块** |
| **物品记录添加** | **物品管理模块、用户管理模块** |
| **物品记录查询** | **物品管理模块、用户管理模块** |
| **物品记录修改** | **物品管理模块、用户管理模块** |
| **物品记录删除** | **物品管理模块、用户管理模块** |

**3 接口设计**

本系统所包含的接口如下表所示，每个接口对应不同的功能模块，实现具体的服务和数据库操作。

**3.1 服务接口**

图形用户界面, 文本, 应用程序

描述已自动生成

**3.1.1 DatabaseBakService (数据库备份管理)**

* **findList(PageBean<DatabaseBak> pageBean)**
  + 目标：分页查找数据库备份记录。
  + 参数：pageBean - 分页信息，包含当前页码和每页显示的数量。
  + 流程：根据pageBean创建Pageable对象, 使用databaseBakDao的findAll方法进行分页查询。将查询结果设置到pageBean中，并返回。
* **save(DatabaseBak databaseBak)**
  + 目标：添加或修改数据库备份记录。
  + 参数：databaseBak - 要保存或更新的数据库备份记录。
  + 流程：调用databaseBakDao的save方法保存或更新记录。返回保存或更新后的记录。
* **find(Long id)**
  + 目标：根据id查询数据库备份记录。
  + 参数：id - 数据库备份记录的ID。
  + 流程：调用databaseBakDao的find方法根据ID查询记录。返回查询到的记录。
* **delete(Long id)**
  + 目标：根据id删除数据库备份记录。
  + 参数：id - 要删除的数据库备份记录的ID。
  + 流程：调用databaseBakDao的deleteById方法删除记录。
* **backup()**
  + 目标：备份数据库。
  + 参数：无
  + 流程：检查备份目录是否存在，不存在则创建。构建备份命令并执行，将数据库备份到指定目录。创建一个新的DatabaseBak对象，记录备份信息。保存备份记录到数据库。记录日志和操作日志。
* **restore(Long id)**
  + 目标：还原数据库。
  + 参数：id - 要还原的数据库备份记录的ID。
  + 流程：根据ID查询备份记录。构建还原命令并执行，从备份文件中还原数据库。记录日志和操作日志。
* **total()**
  + 目标：获取备份总数。
  + 参数：无
  + 流程：调用databaseBakDao的count方法计算备份记录总数。返回总数。

**3.1.2 MenuService (菜单管理)**

* **save(Menu menu)**
  + 目标：菜单添加/编辑
  + 参数：menu - 菜单对象
  + 流程：调用menuDao的save方法，将传入的菜单对象进行保存或更新操作，并返回保存后的结果。
* **findAll()**
  + 目标：获取所有的菜单列表
  + 参数：无
  + 流程：调用menuDao的findAll方法，查询数据库中的所有菜单记录，并将结果以列表形式返回。
* **find(Long id)**
  + 目标：根据id查询菜单
  + 参数：id - 菜单ID
  + 流程：调用menuDao的find方法，根据传入的菜单ID查询数据库中的菜单记录，并将结果返回。
* **delete(Long id)**
  + 目标：根据id删除一条记录
  + 参数：id - 菜单ID
  + 流程：调用menuDao的deleteById方法，根据传入的菜单ID删除数据库中的菜单记录。

**3.1.3 OperaterLogService (后台操作管理)**

* **save(OperaterLog operaterLog)**
  + 目标：添加/修改操作日志
  + 参数：operaterLog - 操作日志对象
  + 流程：调用operaterLogDao的save方法，将传入的操作日志对象进行保存或更新操作，并返回保存后的结果。
* **findLastestLog(int size)**
  + 目标：获取指定条数的操作日志列表
  + 参数：size - 指定的条数
  + 流程：调用operaterLogDao的findLastestLog方法，根据传入的条数查询数据库中的操作日志记录，并将结果以列表形式返回。
* **findById(Long id)**
  + 目标：根据id查询单条数据
  + 参数：id - 操作日志ID
  + 流程：调用operaterLogDao的find方法，根据传入的操作日志ID查询数据库中的操作日志记录，并将结果返回。
* **findAll()**
  + 目标：返回所有的记录
  + 参数：无
  + 流程：调用operaterLogDao的findAll方法，查询数据库中的所有操作日志记录，并将结果以列表形式返回。
* **delete(Long id)**
  + 目标：删除单条记录
  + 参数：id - 操作日志ID
  + 流程：调用operaterLogDao的deleteById方法，根据传入的操作日志ID删除数据库中的操作日志记录。
* **deleteAll()**
  + 目标：清空整张表
  + 参数：无
  + 流程：调用operaterLogDao的deleteAll方法，清空数据库中的操作日志表。
* **add(String operater, String content)**
  + 目标：操作日志添加
  + 参数：operater - 操作者；content - 操作内容
  + 流程：创建一个新的操作日志对象，设置操作者和操作内容，然后调用save方法保存该对象。
* **add(String content)**
  + 目标：操作日志添加
  + 参数：content - 操作内容
  + 流程：调用add方法，传入操作内容和当前登录用户的用户名（如果获取失败，则使用"未知(获取登录用户失败)"作为操作者）。
* **findList(OperaterLog operaterLog, PageBean<OperaterLog> pageBean)**
  + 目标：分页查找日志
  + 参数：operaterLog - 操作日志对象；pageBean - 分页信息对象
  + 流程：创建一个ExampleMatcher对象，设置匹配规则，然后创建一个Example对象，将操作日志对象和ExampleMatcher对象传入。接着创建一个Pageable对象，设置分页信息。最后调用operaterLogDao的findAll方法，根据Example和Pageable对象查询操作日志记录，并将结果存入pageBean对象中。
* **total()**
  + 目标：操作日志总数
  + 参数：无
  + 流程：调用operaterLogDao的count方法，查询数据库中的操作日志总数，并返回结果。

**3.1.4 RoleService (后台角色操作管理)**

* **save(Role role)**
  + 目标：角色添加/编辑
  + 参数：role - 角色对象
  + 流程：调用roleDao的save方法，将传入的角色对象进行保存或更新操作，并返回保存后的结果。
* **findAll()**
  + 目标：获取所有的角色列表
  + 参数：无
  + 流程：调用roleDao的findAll方法，查询数据库中的所有角色记录，并返回结果。
* **findByName(Role role, PageBean pageBean)**
  + 目标：分页按角色名称搜索角色列表
  + 参数：role - 角色对象；pageBean - 分页信息对象
  + 流程：创建一个ExampleMatcher对象，设置匹配规则，然后创建一个Example对象，将角色对象和ExampleMatcher对象传入。接着创建一个Pageable对象，设置分页信息。最后调用roleDao的findAll方法，根据Example和Pageable对象查询角色记录，并将结果存入pageBean对象中。
* **find(Long id)**
  + 目标：根据id查询角色
  + 参数：id - 角色ID
  + 流程：调用roleDao的find方法，根据传入的角色ID查询对应的角色记录，并返回结果。
* **delete(Long id)**
  + 目标：根据id删除一条记录
  + 参数：id - 角色ID
  + 流程：调用roleDao的deleteById方法，根据传入的角色ID删除对应的角色记录。

**3.1.5 UserService (用户管理)**

* **find(Long id)**
  + 目标：根据用户id查询用户信息
  + 参数：id - 用户ID
  + 流程：调用userDao的find方法，根据传入的用户ID查询对应的用户记录，并返回结果。
* **findByUsername(String username)**
  + 目标：按照用户名查找用户
  + 参数：username - 用户名
  + 流程：调用userDao的findByUsername方法，根据传入的用户名查询对应的用户记录，并返回结果。
* **save(User user)**
  + 目标：用户添加/编辑操作
  + 参数：user - 用户对象
  + 流程：调用userDao的save方法，将传入的用户对象进行保存或更新操作，并返回保存后的结果。
* **findList(User user, PageBean pageBean)**
  + 目标：分页查询用户列表
  + 参数：user - 用户对象；pageBean - 分页信息对象
  + 流程：创建一个ExampleMatcher对象，设置匹配规则，然后创建一个Example对象，将用户对象和ExampleMatcher对象传入。接着创建一个Pageable对象，设置分页信息。最后调用userDao的findAll方法，根据Example和Pageable对象查询用户记录，并将结果存入pageBean对象中。
* **isExistUsername(String username, Long id)**
  + 目标：判断用户名是否存在，添加和编辑均可判断
  + 参数：username - 用户名；id - 用户ID
  + 流程：调用userDao的findByUsername方法，根据传入的用户名查询对应的用户记录。如果存在且不是编辑用户的本身，则返回true，否则返回false。
* **delete(Long id)**
  + 目标：按照用户id删除
  + 参数：id - 用户ID
  + 流程：调用userDao的deleteById方法，根据传入的用户ID删除对应的用户记录。
* **total()**
  + 目标：返回用户总数
  + 参数：无
  + 流程：调用userDao的count方法，查询数据库中的用户总数，并返回结果。

4．系统物理构成与配置

本系统架构图如下图所示。

图示

描述已自动生成

**5．系统数据结构设计**

**5.1概念结构设计**

根据需求分析的系统类信息，可得到ER模型，如下图。

图示

描述已自动生成

图示

描述已自动生成

**5.2 逻辑结构设计**

本节详细描述了系统中各个表的逻辑结构设计，包括表名、字段名、数据类型、数据长度及字段说明。这些表涵盖了管理员信息、角色信息、用户信息、菜单信息、物品信息、求购物品信息、举报物品信息、物品评论信息、后台操作日志记录、网站设置信息、新闻信息及友情链接等。

**5.2.1 管理员基本信息 (ylrc\_user)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 管理员id(主键) | bigint |  | 管理员信息属性，存储基本管理员信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| email | 邮箱 | varchar | 32 |  |
| head\_pic | 头像 | varchar | 128 |  |
| mobile | 电话 | varchar | 12 |  |
| password | 密码 | varchar | 32 |  |
| sex | 性别 | int |  |  |
| status | 用户状态 | int |  |  |
| username | 用户名 | varchar | 18 |  |
| role\_id | 角色id | bigint |  |  |

**5.2.2 角色信息 (ylrc\_role)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 角色id(主键) | bigint |  | 管理员角色信息属性，存储基本管理员角色信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| name | 角色名 | varchar | 18 |  |
| remark | 角色备注 | varchar | 128 |  |
| status | 角色状态 | int |  |  |

**5.2.3 角色权限 (ylrc\_role\_authorities)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| role\_id | 角色id(主键) | bigint |  | 角色所拥有权限 |
| authorities\_id | 权限id | bigint |  |  |

**5.2.4 用户基本信息 (ylrc\_student)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 用户id(主键) | int |  | 用户信息属性，存储基本用户信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| academy | 所属学院 | varchar | 18 |  |
| grade | 所属年级 | varchar | 18 |  |
| mobile | 电话 | varchar | 18 |  |
| nickname | 昵称 | varchar | 32 |  |
| qq | QQ号 | varchar | 18 |  |
| school | 所属学校 | varchar | 18 |  |
| sn | 学生学号 | varchar | 18 |  |
| head\_pic | 头像 | varchar | 128 |  |
| status | 学生状态 | int |  |  |
| password | 密码 | varchar | 18 |  |

**5.2.5 菜单信息 (ylrc\_menu)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 菜单id(主键) | bigint |  | 菜单信息属性，存储基本菜单信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| name | 菜单名称 | varchar | 18 |  |
| url | 菜单路径 | varchar | 128 |  |
| icon | 菜单图标 | varchar | 32 |  |
| sort | 菜单顺序 | int |  |  |
| parent\_id | 菜单父分类 | bigint |  |  |
| is\_button | 是否是按钮 | bit | 1 |  |
| is\_show | 是否显示 | bit | 1 |  |

**5.2.6 物品信息 (ylrc\_goods)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 物品id(主键) | bigint |  | 物品信息属性，存储基本物品信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| buy\_price | 购买价格 | float |  |  |
| content | 物品描述 | varchar | 1024 |  |
| flag | 是否擦亮 | int |  |  |
| name | 物品名称 | varchar | 32 |  |
| photo | 物品图片 | varchar | 128 |  |
| recommend | 是否推荐 | int |  |  |
| sell\_price | 出售价格 | float |  |  |
| status | 物品状态 | int |  |  |
| goods\_category\_id | 物品分类 | bigint | 18 |  |
| student\_id | 学生id | bigint |  |  |
| view\_number | 物品浏览量 | int |  |  |

**5.2.7 物品分类 (ylrc\_goods\_category)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 分类id(主键) | bigint |  | 物品分类信息属性，存储基本物品分类信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| icon | 分类图标 | varchar | 32 |  |
| name | 分类名称 | varchar | 18 |  |
| sort | 分类顺序 | int |  |  |
| parent\_id | 分类父分类 | bigint |  |  |

**5.2.8 求购物品信息 (ylrc\_wanted\_goods)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 主键id | bigint |  | 求购物品信息属性，存储基本求购物品信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| content | 物品描述 | varchar | 1024 |  |
| name | 物品名称 | varchar | 32 |  |
| sell\_price | 期望价格 | float |  |  |
| trade\_place | 交易地点 | varchar | 128 |  |
| view\_number | 浏览量 | int |  |  |
| student\_id | 所属学生 | bigint |  |  |

**5.2.9 举报物品信息 (ylrc\_report\_goods)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 主键id | bigint |  | 举报物品信息属性，存储基本举报物品信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| content | 举报原因 | varchar | 1024 |  |
| goods\_id | 举报物品id | bigint |  |  |
| student\_id | 所属学生 | bigint |  |  |

**5.2.10 物品评论信息 (ylrc\_comment)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 主键id | bigint |  | 物品评论信息属性，存储基本物品评论信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| content | 评论内容 | varchar | 1024 |  |
| reply\_to | 回复者 | varchar | 64 |  |
| goods\_id | 评论物品id | bigint |  |  |
| student\_id | 所属学生 | bigint |  |  |

**5.2.11 后台操作日志记录 (ylrc\_operater\_log)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 主键id | bigint |  | 操作日志信息属性，存储基本操作日志信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| content | 操作内容 | varchar | 1024 |  |
| operater | 操作者 | varchar | 18 |  |

**5.2.12 网站设置信息 (ylrc\_site\_setting)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 菜单id | bigint |  | 网站设置信息属性，存储基本网站设置信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| all\_rights | 网站版权 | varchar | 256 |  |
| logo\_1 | 网站logo1 | varchar | 256 |  |
| logo\_2 | 网站logo2 | varchar | 256 |  |
| qrcode | 网站公众号二维码 | varchar | 256 |  |
| site\_name | 网站名称 | varchar | 128 |  |
| site\_url | 网站地址 | varchar | 256 |  |

**5.2.13 新闻信息 (ylrc\_news)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 主键id | bigint |  | 新闻信息属性，存储基本新闻信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| content | 新闻内容 | varchar | 10024 |  |
| sort | 分类顺序 | int |  |  |
| title | 新闻标题 | varchar | 1024 |  |
| view\_number | 浏览量 | int |  |  |

**5.2.14 友情链接 (ylrc\_friend\_link)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 主键id | bigint |  | 友情链接信息属性，存储基本友情链接信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| name | 链接名称 | varchar | 64 |  |
| sort | 显示顺序 | int |  |  |
| url | 链接地址 | varchar | 256 |  |

**5.2.15 数据库备份记录 (ylrc\_database\_bak)**

| **名称** | **数据名称** | **数据类型** | **数据长度** | **说明** |
| --- | --- | --- | --- | --- |
| id | 主键id | bigint |  | 数据库备份记录信息属性，存储基本数据库备份记录信息 |
| creat\_time | 创建时间 | datetime |  |  |
| update\_time | 更新时间 | datetime |  |  |
| filename | 文件名 | varchar | 32 |  |
| filepath | 文件路径 | varchar | 128 |  |

**1. 业务流程图 (Business Process Diagram, BPD)**

**1.1 管理员登录与管理流程**

**步骤:**

1. **管理员登录**
   * 管理员输入用户名和密码。
   * 系统验证用户名和密码。
   * 验证通过后，进入管理界面。
2. **管理员管理**
   * 管理员可以查看和管理用户信息、角色信息、物品信息等。
   * 管理员可以添加、修改或删除用户、角色、物品等信息。
3. **退出登录**
   * 管理员点击退出按钮，系统注销当前会话。

**关键节点:**

* 用户名和密码验证
* 添加/修改/删除操作
* 退出登录

**1.2 用户注册与登录流程**

**步骤:**

1. **用户注册**
   * 用户填写注册信息（用户名、密码、邮箱、电话等）。
   * 系统验证信息的有效性。
   * 注册成功后，发送确认邮件或短信。
2. **用户登录**
   * 用户输入用户名和密码。
   * 系统验证用户名和密码。
   * 验证通过后，进入用户界面。
3. **用户操作**
   * 用户可以发布物品、求购物品、评论物品等。
4. **退出登录**
   * 用户点击退出按钮，系统注销当前会话。

**关键节点:**

* 注册信息验证
* 用户名和密码验证
* 发布物品、求购物品、评论物品
* 退出登录

**2. 数据流图 (Data Flow Diagram, DFD)**

**2.1 一级数据流图 (Context Level DFD)**

**外部实体:**

* 管理员
* 用户

**处理过程:**

* 系统

**数据流:**

* 管理员 -> 系统 (登录、管理操作)
* 用户 -> 系统 (注册、登录、发布物品、求购物品、评论物品)

**2.2 二级数据流图 (Level 1 DFD)**

**处理过程:**

* 用户注册
* 用户登录
* 管理员登录
* 管理员管理
* 发布物品
* 求购物品
* 评论物品

**数据流:**

* 用户 -> 系统 (注册信息)
* 系统 -> 用户 (注册成功通知)
* 用户 -> 系统 (登录信息)
* 系统 -> 用户 (登录结果)
* 管理员 -> 系统 (登录信息)
* 系统 -> 管理员 (登录结果)
* 管理员 -> 系统 (管理操作)
* 系统 -> 管理员 (管理结果)
* 用户 -> 系统 (发布物品信息)
* 系统 -> 用户 (发布结果)
* 用户 -> 系统 (求购物品信息)
* 系统 -> 用户 (求购结果)
* 用户 -> 系统 (评论信息)
* 系统 -> 用户 (评论结果)

**3. 系统ER图 (Entity-Relationship Diagram, ERD)**

**实体 (Entities)**

1. **管理员 (ylrc\_user)**
2. **角色 (ylrc\_role)**
3. **用户 (ylrc\_student)**
4. **物品 (ylrc\_goods)**
5. **物品分类 (ylrc\_goods\_category)**
6. **求购物品 (ylrc\_wanted\_goods)**
7. **举报物品 (ylrc\_report\_goods)**
8. **物品评论 (ylrc\_comment)**
9. **菜单 (ylrc\_menu)**
10. **后台操作日志 (ylrc\_operater\_log)**
11. **网站设置 (ylrc\_site\_setting)**
12. **新闻 (ylrc\_news)**
13. **友情链接 (ylrc\_friend\_link)**
14. **数据库备份记录 (ylrc\_database\_bak)**

**关系 (Relationships)**

1. **管理员与角色 (ylrc\_user - ylrc\_role)**
   * 一个管理员可以有多个角色
   * 一个角色可以被多个管理员拥有
2. **用户与物品 (ylrc\_student - ylrc\_goods)**
   * 一个用户可以发布多个物品
   * 一个物品属于一个用户
3. **用户与求购物品 (ylrc\_student - ylrc\_wanted\_goods)**
   * 一个用户可以求购多个物品
   * 一个求购物品属于一个用户
4. **用户与举报物品 (ylrc\_student - ylrc\_report\_goods)**
   * 一个用户可以举报多个物品
   * 一个举报物品属于一个用户
5. **用户与物品评论 (ylrc\_student - ylrc\_comment)**
   * 一个用户可以评论多个物品
   * 一个评论属于一个用户
6. **物品与物品分类 (ylrc\_goods - ylrc\_goods\_category)**
   * 一个物品属于一个分类
   * 一个分类可以包含多个物品
7. **角色与权限 (ylrc\_role - ylrc\_role\_authorities)**
   * 一个角色可以有多个权限
   * 一个权限可以被多个角色拥有
8. **菜单与子菜单 (ylrc\_menu - ylrc\_menu)**
   * 一个菜单可以有多个子菜单
   * 一个子菜单属于一个菜单
9. **新闻与分类 (ylrc\_news - ylrc\_news\_category)**
   * 一个新闻属于一个分类
   * 一个分类可以包含多个新闻
10. **友情链接与网站设置 (ylrc\_friend\_link - ylrc\_site\_setting)**
    * 一个网站设置可以包含多个友情链接
    * 一个友情链接属于一个网站设置
11. **后台操作日志与管理员 (ylrc\_operater\_log - ylrc\_user)**
    * 一个管理员可以有多个操作日志
    * 一个操作日志属于一个管理员
12. **数据库备份记录与系统 (ylrc\_database\_bak - 系统)**
    * 系统可以有多个数据库备份记录
    * 一个数据库备份记录属于系统