

深圳大学实验报告

课程名称: Database System

实验项目名称: "数据库设计 Database Design"

学院: Computer Science and Software Engineering

专业: 2023 计算机科学与技术（创新班）02

指导教师: Basker George

报告人: 林宪亮 学号: 2022150130 班级: 国际班

实验时间: 11th November, 2024

实验报告提交时间: 30th November, 2024

实验目的：

- 1、掌握数据库设计方法；
- 2、了解概念模型、逻辑数据模型和物理数据模型之间的关系和不同；
- 3、掌握使用高级语言访问、操作数据库，加深对前后台数据交互的理解。

实验内容：

- 1、E-R 图至少包括 8 个实体和 7 个联系；
- 2、需要考虑关系完整性约束：主键约束、外键约束、空值约束；
- 3、设计至少 1 个视图、1 个索引（非主键、非外键索引）、1 个触发器、1 个存储过程或存储函数（视图、索引、触发器、存储过程或存储函数可以直接在 DBMS 中创建，不一定要在 pdm 图的时候创建）；
- 4、数据库使用 MySQL；
- 5、编程语言不限；
- 6、考察重点是数据库设计、前后台数据交互。图形界面是必须的，以简单明了为上。

实验过程：

选择的题目：宠物认领平台

1. E-R 图设计：

1.1 实体联系设计

设计思路：

宠物认领平台的设计旨在为用户和管理员提供方便高效的宠物认领管理服务，实现宠物信息管理、认领申请处理和领养跟踪记录等功能。

以用户为中心，平台支持两类用户：管理员和普通用户。提供差异化的功能权限，如用户认领申请、管理员的审核与管理。模块化与扩展性，实体划分清晰，确保数据存储和操作简单、易扩展。设计合理的表间关系，避免数据冗余，同时提高查询效率。完整性和安全性，数据库设计中考虑主键、外键和空值约束，确保数据的完整性和一致性。增加管理员日志记录，提供操作追踪能力。用户体验友好性，提供用户登录、宠物浏览和认领等直观功能界面。支持宠物照片展示和领养回访功能，增强用户与平台的互动。

平台功能模块划分：

用户管理模块，用户注册、登录和角色区分（普通用户/管理员）。普通用户可浏览宠物信息和提交认领申请。管理员可冻结用户账号、查看和审核申请。宠物管理模块，管理员可添加、修改、删除宠物信息。支持上传宠物照片并分类显示（按品种、年龄等）。认领管理模块，用户提交认领申请并查看申请状态。管理员审核申请，并记录审批状态及备注。领养跟踪模块，管理员记录成功认领的宠物跟踪信息，如回访时间和反馈。系统日志模块，记录管理员的操作历史，便于日后追踪和管理。

完成的内容：

实体设计：

通过分析平台需求，设计了8个实体，分别覆盖用户、宠物、申请、照片、品种等核心数据。同时利用外键定义了关键的实体间关系（如用户与申请、宠物与照片）。

数据关系设计：

数据库设计遵循第三范式，确保数据无冗余，并采用主外键约束，明确实体间的多对一或一对多关系。

视图与索引：

设计一个展示宠物认领状态的视图，为管理员提供快速查询支持。创建非主键索引优化常用查询（如按宠物品种筛选）。

触发器与存储过程：

使用触发器确保用户在提交认领申请时宠物状态自动更新；存储过程实现定期统计平台关键数据（如已认领宠物数量、用户申请总数等）。

完成了实验的设计要求！！！！

设计细节：

实体和属性设计：

1. 用户（User）

设计思路：用户是平台的核心实体之一，分为管理员和普通用户。普通用户用于认领宠物，管理员管理宠物信息、审批申请等。

主要属性：user_id（主键）：唯一标识用户。

username：用户名。

password：用户密码。

role：用户角色（普通用户/管理员）。

name：用户姓名。

gender：性别。

age：年龄。

address：用户地址。

email：邮箱地址。

phone：联系电话。

status：账号状态（正常/冻结）。

2. 宠物（Pet）

设计思路：宠物是平台的核心实体之一，用于存储可认领的宠物信息。

主要属性：pet_id（主键）：唯一标识宠物。

nickname：宠物昵称。

birthday：生日。

age：年龄。

gender：性别。

breed：品种。

color：颜色。

character：性格描述。

status：状态（待认领/已认领）。

3. 认领申请 (AdoptionApplication)

设计思路：用于存储用户认领宠物的申请信息，便于管理员审核。

主要属性：application_id (主键)：唯一标识申请。

user_id (外键)：申请人 ID。

pet_id (外键)：申请的宠物 ID。

application_date：申请日期。

status：申请状态（待审核/批准/驳回）。

remark：管理员审核备注。

4. 领养跟踪 (AdoptionFollowUp)

设计思路：用于存储管理员对成功认领宠物的回访记录。

主要属性：followup_id (主键)：唯一标识回访记录。

adoption_id (外键)：对应认领申请 ID。

followup_date：回访日期。

feedback：回访反馈。

5. 宠物照片 (PetPhoto)

设计思路：宠物的照片可以方便用户选择认领。

主要属性：photo_id (主键)：唯一标识照片。

pet_id (外键)：对应宠物 ID。

photo_url：照片链接。

6. 宠物品种 (Breed)

设计思路：独立存储宠物的品种信息，用于规范化品种数据。

主要属性：breed_id (主键)：唯一标识品种。

name：品种名称。

description：品种描述。

7. 管理员日志 (AdminLog)

设计思路：记录管理员操作日志，用于管理平台的操作记录。

主要属性：log_id (主键)：唯一标识日志。

admin_id (外键)：管理员 ID。

operation：操作描述。

operation_date：操作日期。

8. 地址 (Address)

设计思路：存储用户的详细地址，便于后续的联系。

主要属性：address_id (主键)：唯一标识地址。

user_id (外键)：对应用户 ID。

province：省。

city：市。

district：区。

detail：详细地址。

实体关系设计:

用户-认领申请 一个用户可以提交多个认领申请，每个认领申请属于一个用户（1 对多关系）。

宠物-认领申请 一个宠物可以对应多个认领申请，但每个认领申请只针对一个宠物（1对多关系）。

认领申请-领养跟踪 一个认领申请可能有多次回访记录,每次回访属于一个认领申请(1对多关系)。

宠物-照片 每个宠物可以有多张照片，每张照片属于一个宠物（1 对多关系）。

宠物-品种 每个宠物属于一种品种，但每种品种可以有多个宠物（多对一关系）。

用户-地址 一个用户可以有多个地址，每个地址属于一个用户（1 对多关系）。

管理员-操作日志 每个管理员可以有多个操作日志，每条日志对应一个管理员（1 对多关系）。

下面是使用 power-designer 设计的 CDM:

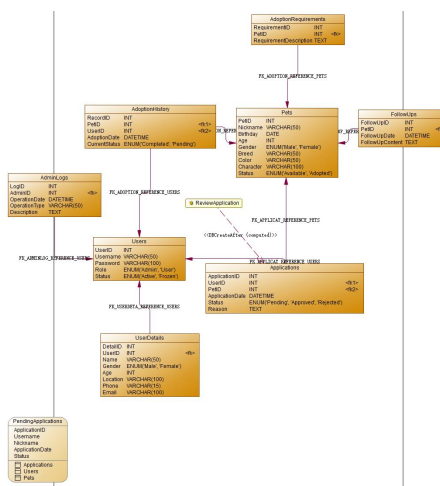


图 1 CDM 模型

1.2 转成逻辑模型

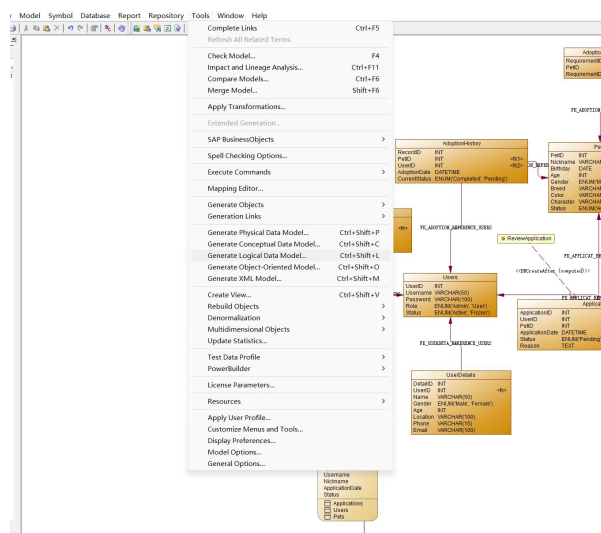


图 2 转换

点击图中的 Generate Logic Data Model 转换成逻辑模型。

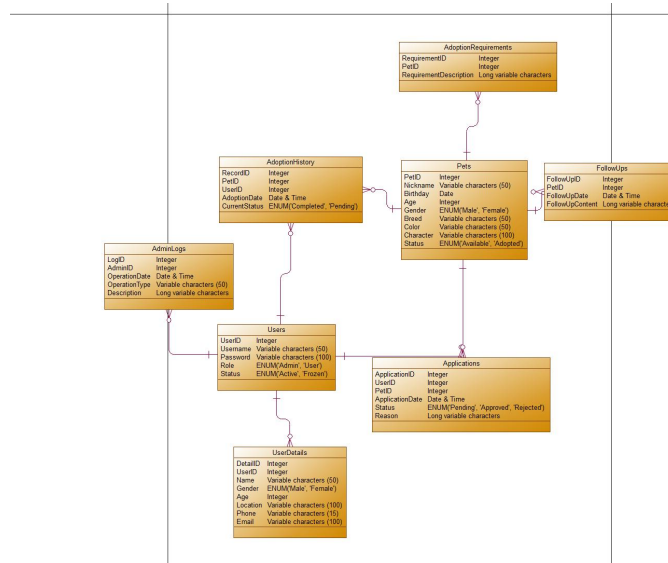


图 3 逻辑模型

如图 3，为 LDM 图。

1.3 转成物理模型

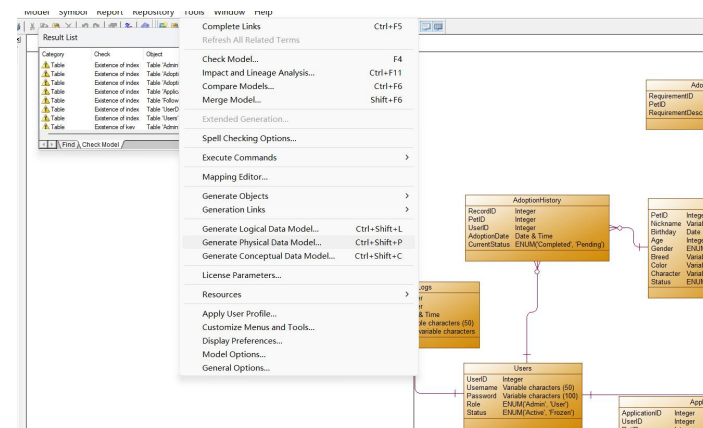


图 4 转换

如图，点击 Generate Physical Data Model 转换成 PDM，如下图所示。

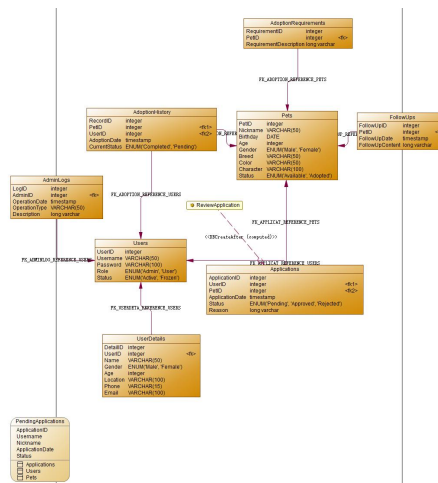


图 5 PDM

视图:

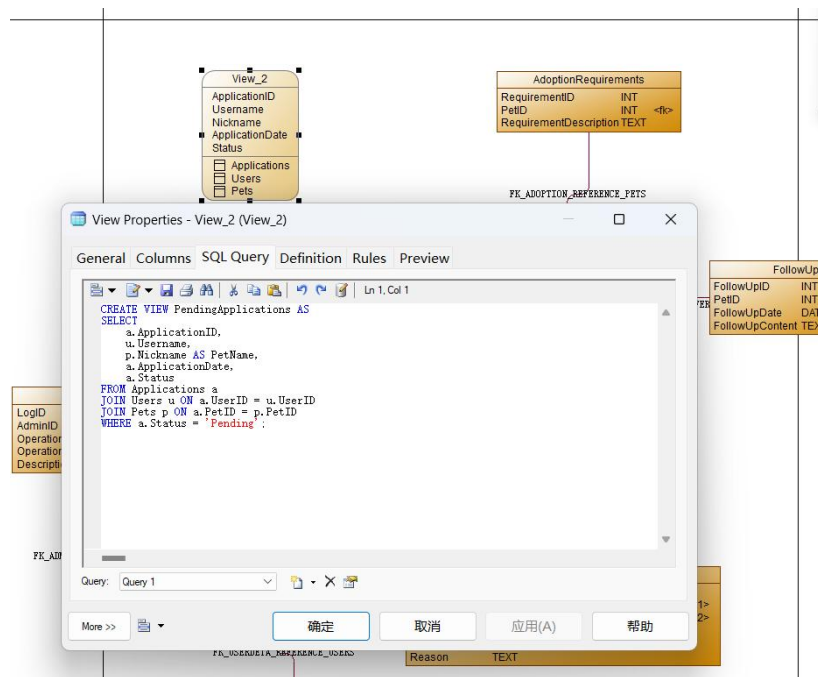


图 6 视图

我创建了一个视图，显示所有“待审批”的申请详情。

索引:

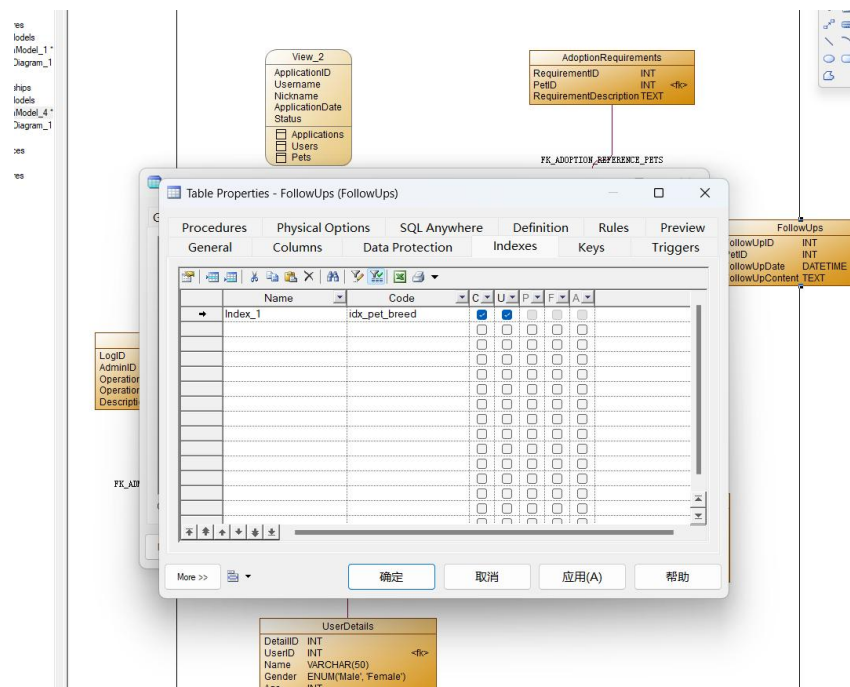


图 7 索引

我为宠物表的品种列创建一个索引。

触发器:

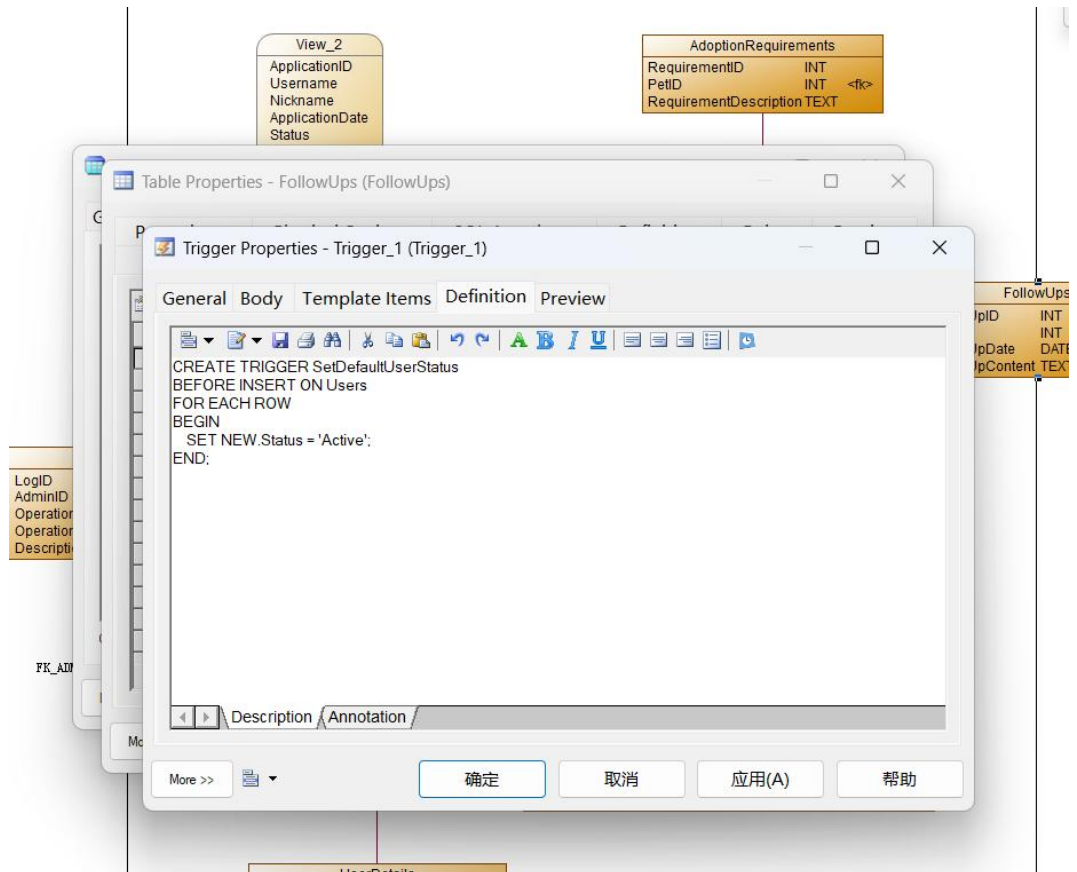


图 8 触发器

在用户表中插入新用户时，默认将用户设置为“未冻结”状态。

存储过程:

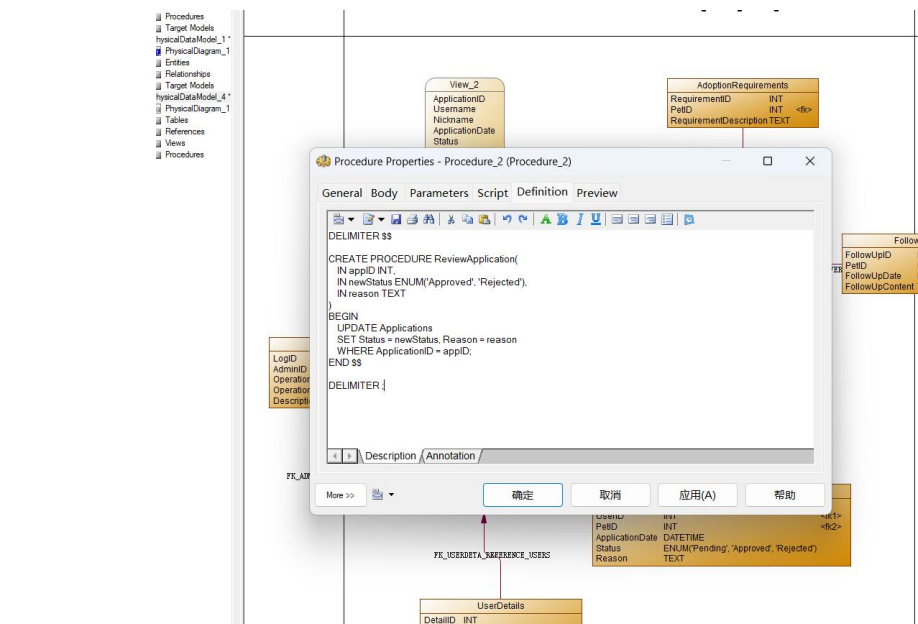


图 9 存储过程

我实现一个存储过程，用于管理员审核领养申请。

3. 导出的 SQL 文件

按照教程的方法，可以将 SQL 文件导出，以下为导出的 SQL 文件：

```
        Description TEXT,
        FOREIGN KEY (AdminID) REFERENCES Users(UserID)
    );

-- 创建领养历史表
CREATE TABLE AdoptionHistory (
    RecordID INT AUTO INCREMENT PRIMARY KEY,
    PetID INT NOT NULL,
    UserID INT NOT NULL,
    AdoptionDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    CurrentStatus ENUM('Completed', 'Pending') DEFAULT 'Pending',
    FOREIGN KEY (PetID) REFERENCES Pets(PetID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

-- 创建领养需求表
CREATE TABLE AdoptionRequirements (
    RequirementID INT AUTO INCREMENT PRIMARY KEY,
    PetID INT NOT NULL,
    RequirementDescription TEXT,
    FOREIGN KEY (PetID) REFERENCES Pets(PetID)
);

-- 创建视图
CREATE VIEW PendingApplications AS
SELECT
    a.ApplicationID,
    u.Username,
    p.Nickname AS PetName,
    a.ApplicationDate,
    a.Status
FROM Applications a
JOIN Users u ON a.UserID = u.UserID
JOIN Pets p ON a.PetID = p.PetID
WHERE a.Status = 'Pending';

-- 创建索引
CREATE INDEX idx_pet_breed ON Pets(Breed);

-- 创建触发器
CREATE TRIGGER SetDefaultUserStatus
```

图 10 SQL 代码

代码内容与 E-R 图设计的数据库一致。

4. 前后端设计

前端技术：

HTML：页面结构设计。

CSS：提供基础页面样式和响应式设计支持。

JavaScript：实现交互功能，如提交表单、动态加载数据。主要功能：普通用户页面：宠物浏览、申请领养、个人信息提交。管理员页面：宠物信息管理、申请审核、领养跟踪记录。

后端技术：

语言和框架：PHP 或 Python（Flask/Django）处理 HTTP 请求，连接数据库。

数据库：MySQL，用于存储用户、宠物、申请等核心数据。

API 设计：提供 RESTful 接口与前端交互。

功能实现流程

（1）用户登录

用户输入用户名和密码，通过 /users/login 接口验证。

返回的 Token 用于后续操作的身份验证。

（2）宠物浏览

前端通过 GET /pets 获取所有宠物列表。

展示宠物信息和图片，支持按条件筛选（如品种、年龄）。

（3）提交领养申请

普通用户点击“申请领养”，填写个人信息。

前端调用 POST /applications 提交申请。
数据库插入申请记录，宠物状态保持“待认领”。

(4) 管理员审核申请
管理员进入后台，通过 GET /applications 获取所有申请。
审核后调用 PUT /applications/{id} 更新申请状态。
若批准，宠物状态更新为“已认领”。

(5) 添加领养跟踪记录
管理员为已认领宠物调用 POST /followups 添加回访记录。
数据库关联宠物 ID 和回访信息。

下面是一些前端界面展示，分别是登录界面，宠物展示界面，管理员界面和领养申请界面。



图 11 登录界面



图 12 宠物展示

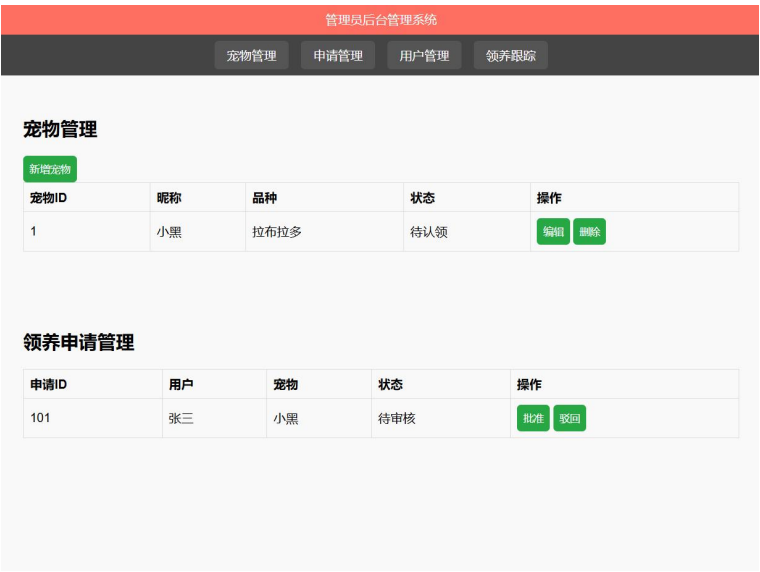


图 13 管理员界面

申请领养

宠物信息

昵称: 小黑

品种: 拉布拉多

年龄: 2 岁

申请信息

姓名

性别

男

年龄

地址

联系电话

提交申请

图 14 领养申请

实验完成成功！！

实验结果：

1. 确定选题与需求分析

选题：宠物认领平台。

需求分析：平台分为普通用户与管理员角色，功能需求包括：用户注册与登录，分普通用户和管理员两类；

宠物信息管理（新增、修改、删除、浏览）；

普通用户提交领养申请；

管理员审核申请，并记录领养跟踪信息；

管理员可冻结用户账号。

系统通过 E-R 图和功能模块分解明确了关键数据实体和相应关系。

2. 数据库设计

E-R 图：数据库包含 8 个实体和 7 个联系，实体包括用户、宠物、品种、领养申请、管理员、领养跟踪记录等，联系包括用户与领养申请、宠物与品种的多对一关系等。

关系完整性约束：在 E-R 图设计中设置主键约束、外键约束以及空值约束，确保数据库数据的一致性和完整性。

逻辑与物理模型：将 E-R 图转化为逻辑数据模型，最终生成物理数据模型，并使用 MySQL 创建了实际数据库。

3. SQL 脚本导出

SQL 脚本成功创建了数据库，包含所有实体表、联系表以及约束条件。

创建了一个视图（v_pet_applications）用于查看宠物和领养申请情况，一个索引用于优化领养申请查询性能。

4. 高级语言实现

前端采用 HTML、CSS 和 JavaScript 实现图形界面。

后端基于 PHP 或 Python 实现 RESTful API，用于数据库交互。

功能包括用户登录、宠物信息展示、申请领养、管理员审核、跟踪记录等模块。

图形界面设计清晰，支持普通用户和管理员角色切换。

深圳大学学生实验报告用纸

实验总结：

数据库设计符合实验要求，E-R 图包含 8 个实体和 7 个联系，设计全面考虑了关系完整性约束。使用 MySQL 导出 SQL 脚本，成功创建了物理数据库模型。设计了视图、索引、触发器和存储过程，有效提高了数据库查询效率和功能扩展能力。

实现了宠物信息管理、领养申请、用户管理和管理员审核等核心功能。数据交互使用 RESTful API，前后端协作顺畅，数据展示与操作都符合预期。

从需求分析到实体关系设计，深刻理解了数据库的设计流程。通过 RESTful API 实现了高效数据交互，强化了前后端协同开发的能力。实验涉及需求分析、数据库设计、脚本编写、前后端开发等，全面提升了系统设计与实现的能力。

指导教师批阅意见：

成绩评定：

指导教师签字：Basker George
2024 年 12 月 18 日

备注：

- 注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。