

## 目 录

一.	系统需求分析 .....	1
1.	系统描述 .....	1
2.	数据存储需求 .....	1
3.	系统常做的查询与更新 .....	2
4.	应用程序功能 .....	2
二.	数据库概念设计 .....	4
1.	确定实体和属性 .....	4
2.	E-R 图 .....	4
三.	数据库逻辑结构设计 .....	6
1.	关系模式设计 .....	6
2.	基本表设计 .....	7
四.	数据库物理设计和实施 .....	13
1.	数据库的创建 .....	13
2.	创建基本表 .....	13
3.	触发器设计 .....	19
4.	存储过程设计 .....	23
5.	视图设计 .....	23
6.	事件设计 .....	25
7.	索引设计 .....	26
五.	应用程序设计 .....	27
1.	开发及运行环境介绍 .....	27
2.	主要功能设计 .....	27
3.	主要界面 .....	30
六.	心得体会 .....	38

## 一. 系统需求分析

### 1. 系统描述

随着信息技术的发展，越来越多的政府部门开始引入信息管理系统以提高效率。道路交通违章信息管理系统是实现交通管理信息化的重要手段，它能够提高违章信息管理的效率，减少相关环节工作量，广大驾驶员和车主朋友也因此受益。

道路交通违章信息管理系统包括个人信息查询、机动车信息查询、违章查询等前台查询功能，以及驾驶员/车主、机动车和违章记录的添加、删除、修改等后台管理功能。本系统面向的用户分为管理员、高级职员、职员和驾驶员/车主用户四类。管理员负责整个系统的管理、权限和认证控制；高级职员负责驾驶员、机动车、违章信息的录入与修改；普通职员负责驾驶员、机动车信息的录入，以及违章信息的录入和修改；驾驶员/车主可以对相关信息进行查询。

道路交通违章信息管理系统对于促进社会信息化治理，提高行政效率有着积极的意义。

### 2. 数据存储需求

道路交通违章信息管理系统需要存储如下信息。有关注册用户的信息涉及用户 ID、用户名、密码、身份、用户所在用户组等用于认证的基本信息，以及各个用户以及用户组的权限信息。此外，对于在系统中注册的车主和驾驶员，还需要存储身份证号、姓名和联系方式等个人信息。

对于驾驶员，还应该存储诸如驾照号、准驾车型以及有效期等驾驶证信息，每个车主或驾驶员账户至多仅对应一个唯一的驾驶证信息。

数据库中也存储机动车基本信息以及与机动车关联的行驶证基本信息。机动车基本信息包括品牌型号、VIN 编号、出厂日期以及车牌号等信息。机动车行驶证信息包括机动车 VIN 编号、车辆类型、使用性质、注册日期和发证日期等信息。每个机动车至多仅对应一个唯一的行驶证信息，每个车主或者驾驶员可以拥有多部车辆以及多个机动车行驶证。

数据库中也存储违章信息，违章信息包括违章车辆、驾驶员以及违章日期、位置、扣分、罚款、罚款缴纳截至日期等信息。一辆机动车会有很多违章记录，一位驾驶员也会有许多违章记录，违章记录依附于机动车和驾驶员。驾驶员缴纳罚款以后，也需要存储包括违章编号以及罚款缴纳日期的罚款缴纳记录。如果因为违章导致驾驶员在单一记分周期内累积扣分超过 12 分，还需要存储驾驶员参加道路安全学习的信息。

对于超级管理员、高级职员以及职员账户，由于此类账户可以对在后台数据库中数据进行增加、修改以及删除等操作，出于信息安全考虑，需要对此类账户在后台对数据库数据进行的操作进行记录并存储在数据库中，以供日后需要时查阅。

### 3. 系统常做的查询与更新

经常做的查询，或许对创建索引有影响的：

- 由驾驶员姓名或身份证号查询驾驶员相关信息及其驾驶证信息
- 由车主信息、车牌号码、VIN 编号以及汽车品牌等查询机动车相关信息以及行驶证信息
- 由驾驶员或车主姓名、身份证号以及车牌号码查询违章信息
- 以罚款缴纳状态查询违章信息

根据经常做的查询，需要创建有关视图的：

- 由机动车的品牌、车型、车牌号码等信息查询车主信息
- 查询逾期缴纳罚款的违章及其详细信息
- 查询至今尚未缴纳罚款的违章及其详细信息
- 查询违章处理记录以及与其相关的违章信息

关于更新：

- 用户名和密码等基本认证信息较少发生更新，以插入为主
- 驾驶员或车主的基本个人信息较少发生更新，以插入为主
- 对驾驶证所做的更新包括因违章造成的扣分以及驾驶证状态修改，根据记分周期重置分数等操作，其余以插入为主
- 对机动车所做的更新包括机动车过户、车牌变更以及机动车报废的后续处理等，其余以插入为主
- 对行驶证所做的更新包括使用性质的变更操作，其余以插入为主
- 在罚款缴纳完毕后，需要对有关违章的状态信息进行更新，其余以插入为主；违章处理记录不进行更新，只进行插入
- 在道路安全学习完毕后，需要对有关学习的状态信息进行更新，其余以插入为主；学习完成记录不进行更新，只进行插入
- 管理员和职员的操作记录不进行更新，只进行插入

### 4. 应用程序功能

前台的主要功能如下：

- 1) 用户登录：驾驶员或车主用户通过从主管部门获取的用户名和密码登录前台系统
- 2) 个人中心：
  - a. 个人信息：驾驶员或车主用户查看个人基本信息
  - b. 驾驶证信息：驾驶员或车主用户查看与本人用户关联的驾驶证信息
  - c. 提醒功能：对于驾驶证即将到期或者已经到期的驾驶员，显示提醒信息；对于扣分已达 12 分的驾驶员，显示提醒信息
- 3) 我的车辆：以列表形式查看账户名下机动车的简单信息，点击单个表项的超链接可以查看机动车的详细信息
- 4) 车辆详细信息：查看某个车辆的详细信息，用户只能查看本人名下车辆的详细信息及其行驶证信息，试图查看他人车辆信息的请求会被拒绝

- 5) 我的违章：以列表形式查看与本人关联或与本人名下机动车关联的违章的简单信息，点击单个表项的超链接可以查看违章的详细信息
  - 6) 违章详细信息：查看某条违章的详细信息，用户只能查看与本人关联或与本人名下机动车关联的违章的详细信息，试图查看其它违章信息的请求会被拒绝
  - 7) 登出：从前台系统登出，返回登录界面
- 后台的主要功能如下：
- 1) 用户登录：超级管理员、高级职员和职员账户通过预先设定的密码登录后台管理系统
  - 2) 快捷查询：
    - a. 机动车/车主联合查询：机动车信息及其车主个人信息的整合显示，在此模块中可以通过车主信息查询机动车，并通过机动车信息查询车主
    - b. 未处理违章查询：显示至今未缴纳罚款的违章信息，可以通过驾驶员、车主、车牌号码等信息进行搜索
    - c. 逾期违章查询：显示至今逾期缴纳罚款的违章信息，可以通过驾驶员、车主、车牌号码等信息进行搜索
  - 3) 信息管理/查询：
    - a. 驾驶员/车主信息管理：查看并管理驾驶员或车主的基本信息及其驾驶证信息、参加道路安全学习的信息，可以通过身份证号或姓名等字段进行搜索
    - b. 机动车信息管理：查看并管理机动车的信息及其行驶证信息，可以通过车主姓名、车牌号码、品牌以及机动车状态等字段进行搜索
    - c. 违章信息管理：查看并管理违章信息，可以通过驾驶员/车主信息、车牌号码以及违章处理状态等字段进行搜索
    - d. 违章处理记录查询：查看违章处理记录以及与其相关联的违章信息，可以通过驾驶员信息以及车牌号码等字段进行搜索
  - 4) 认证管理：
    - a. 用户管理：管理用户的基本认证信息及其权限，非超级管理员账户仅有查看权限
    - b. 用户组管理：管理用户组的基本信息及其权限，非超级管理员账户仅有查看权限
  - 5) 操作记录：查看本用户在后台管理界面中进行过的操作
  - 6) 登出：从后台管理系统登出，返回登录界面

## 二. 数据库概念设计

### 1. 确定实体和属性

分析道路交通违章信息管理系统的系统需求，将系统中设计的人、物进行抽象，得到了系统的实体如下所示：

- 1) 用户基本认证信息实体集。属性包括：用户 ID、用户名、密码、最后登录时间、超级管理状态、后台登录权限、有效状态、账户创建时间
- 2) 驾驶员/车主基本信息实体集。属性包括：身份证号、姓名、性别、民族、国籍、地址、电话、手机、出生日期
- 3) 驾驶证实体集。属性包括：驾驶证号、初次签发日期、准驾车型、有效起始日期、有效年限、分数、状态
- 4) 机动车实体集。属性包括：机动车 VIN 编号、品牌、型号、车型、排量、出厂日期、使用年限、车牌号、状态
- 5) 行驶证实体集。属性包括：行驶证号、使用性质、注册日期、发证日期
- 6) 违章实体集。属性包括：违章 ID、日期、类型、区域、地点、扣分、罚款、缴纳截止日期、罚款缴纳状态
- 7) 违章处理记录实体集。属性包括：违章 ID、处理时间
- 8) 道路安全学习信息实体集。属性包括：学习 ID、创建时间、学习状态
- 9) 道路安全学习记录实体集。属性包括：学习 ID、学习完成时间
- 10) 用户组实体集。属性包括：用户组 ID、名称
- 11) 权限列表实体集。属性包括：权限 ID、供系统识别的权限名称、便于用户理解的权限别名
- 12) 内容类别实体集。属性包括：类别 ID、内容所属子系统、内容
- 13) 管理员/职员账户后台操作记录实体集。属性包括：记录 ID、记录时间、操作对象 ID、操作对象名称、操作类型、操作详细信息

### 2. E-R 图

系统 E-R 图如图 2-1 所示：

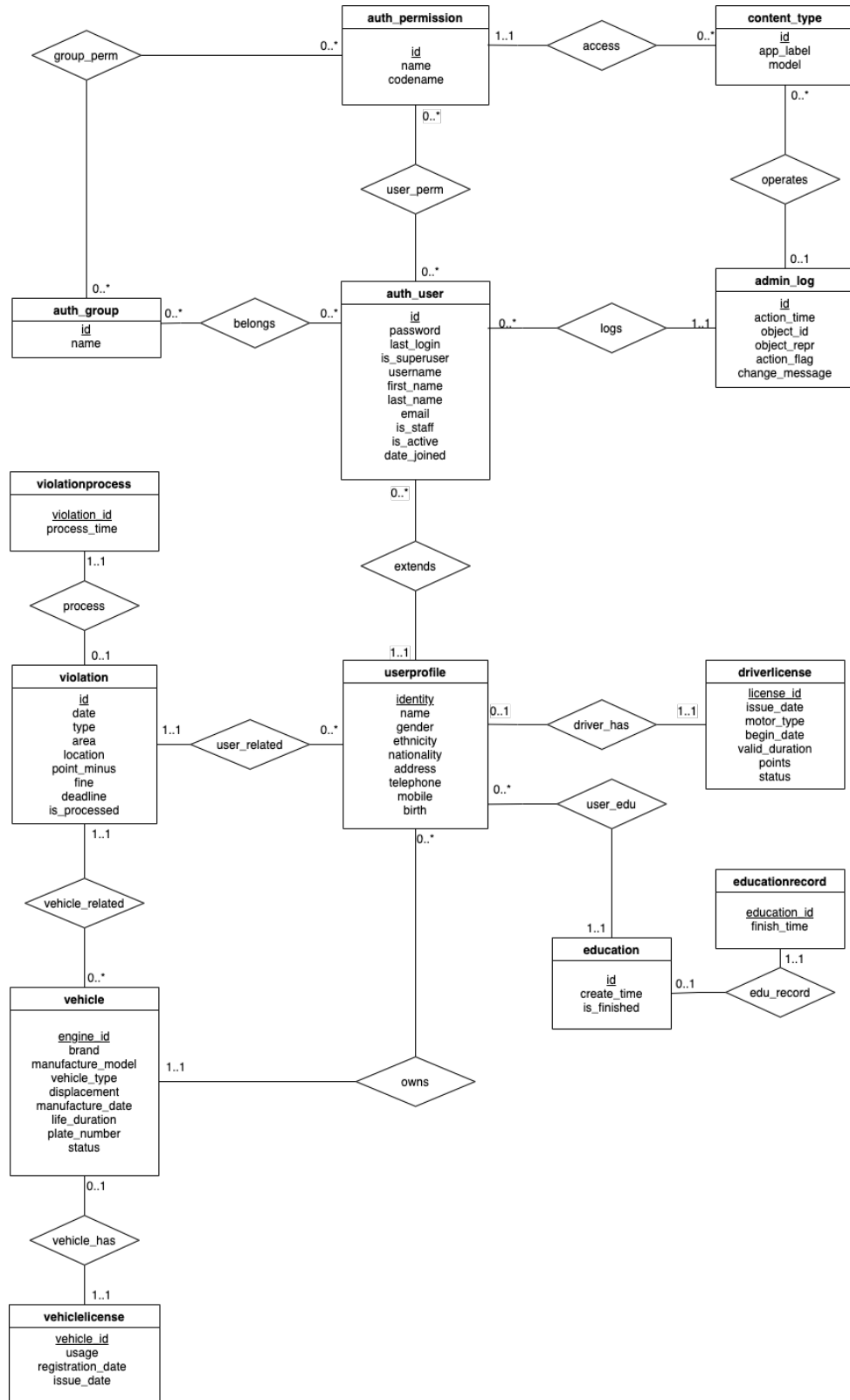


图 2-1 E-R 图

### 三. 数据库逻辑结构设计

#### 1. 关系模式设计

根据概念结构设计得到的 E-R 图和转换规则，得到如下关系模式（主键用下划线标出，外键予以说明）：

1) 用户基本认证信息表(用户 ID，用户名，姓，名，密码，电子邮件，最后登录时间，超级管理状态，后台登录权限，有效状态，账户创建时间)

2) 驾驶员/车主基本信息表(身份证号，姓名，性别，民族，国籍，地址，电话，手机，出生日期，用户 ID)，外键用户 ID 参照用户基本认证信息表的用户 ID 字段。

3) 驾驶证信息表(驾驶证号，初次签发日期，准驾车型，有效起始日期，有效年限，分数，状态)，外键驾驶证号参照驾驶员/车主基本信息表的身份证号字段。

4) 机动车信息表(机动车 VIN 码，品牌，颜色，型号，车型，排量，出厂日期，使用年限，车牌号，状态，车主 ID)，外键车主 ID 参照驾驶员/车主基本信息表的身份证号字段。

5) 行驶证信息表(行驶证号，使用性质，注册日期，发证日期)，外键行驶证号参照机动车信息表的机动车 VIN 码字段。

6) 违章信息表(违章 ID，日期，类型，区域，地点，扣分，罚款，罚款缴纳截止日期，罚款缴纳状态，违章机动车 ID，违章驾驶员 ID)，外键违章机动车 ID 参照机动车信息表的机动车 VIN 码字段，违章驾驶员 ID 参照驾驶员/车主基本信息表的身份证号字段。

7) 违章处理记录表(违章 ID，处理时间)，外键违章 ID 参照违章信息表的违章 ID 字段。

8) 道路安全学习信息表(学习 ID，创建时间，学习状态，驾驶员 ID)，外键驾驶员 ID 参照驾驶员/车主基本信息表的身份证号字段。

9) 道路安全学习记录(学习 ID，学习完成时间)，外键学习 ID 参照道路安全学习信息表的学习 ID 字段。

10) 用户组信息表(用户组 ID，用户组名称)

11) 用户所属用户组信息表(ID，用户 ID，用户组 ID)，外键用户 ID 参照用户基本认证信息表的用户 ID 字段，外键用户组 ID 参照用户组信息表的用户组 ID 字段。

12) 权限信息表(权限 ID，访问内容类别 ID，供系统识别的权限名称，便于用户理解的权限别名)，外键访问内容类别 ID 参照内容类别表的类别 ID 字段。

13) 用户组权限表(ID，用户组 ID，权限 ID)，外键用户组 ID 参照用户组信息表的用户组 ID 字段，外键权限 ID 参照权限信息表的权限 ID 字段。

14) 用户权限表(ID，用户 ID，权限 ID)，外键用户 ID 参照用户基本认证信息表的用户 ID 字段，外键权限 ID 参照权限信息表的权限 ID 字段。

15) 内容类别表(类别 ID，内容所属子系统，内容)

16) 管理员/职员账户后台操作记录表 (记录 ID, 记录时间, 操作对象 ID, 操作对象名称, 操作类型, 操作详细信息, 访问内容类别 ID, 操作用户 ID), 外键访问内容类别 ID 参照内容类别表的类别 ID 字段, 外键操作用户 ID 参照用户基本认证信息表的 ID 字段。

## 2. 基本表设计

基本表设计如表 3-1~3-16 所示。

表 3-1: 用户基本认证信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
date_joined	datetime(6)	否				账户创建时间
email	varchar(254)	否				电子邮件
first_name	varchar(150)	否				名
id	int	否			主键	用户 ID
is_active	tinyint(1)	否				有效状态
is_staff	tinyint(1)	否				后台登录权限
is_superuser	tinyint(1)	否				超级管理状态
last_login	datetime(6)	是		NULL		最后登录时间
last_name	varchar(150)	否				姓
password	varchar(128)	否				用户密码
username	varchar(150)	否	UNI			用户名

表 3-2: 驾驶员/车主基本信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
address	varchar(200)	否				地址
birth	date	否				出生日期
ethnicity	varchar(20)	否				民族
gender	varchar(5)	否				性别
identity	varchar(25)	否			主键	身份证号
mobile	varchar(25)	否				手机
name	varchar(80)	否				姓名
nationality	varchar(10)	否				国籍
telephone	varchar(15)	否				电话



user_id	int	否	UNI		外键	用户 ID
---------	-----	---	-----	--	----	-------

表 3-3: 驾驶证信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
begin_date	date	否				有效期 起始日期
issue_date	date	否				初次签 发日期
motor_type	varchar(3)	否				准驾车 型
points	int	否				分数
status	varchar(5)	否				状态
user_profile_id	varchar(25)	否			主 键 外 键	驾驶证 号
valid_duration	int unsigned	否				有效年 限

表 3-4: 机动车信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
brand	varchar(25)	否				品牌
color	varchar(20)	否				颜色
displacement	decimal(5, 2)	否				排量
engine_id	varchar(25)	否			主 键	机动车 VIN 码
life_duration	int unsigned	否				使用年 限
manufacture_date	date	否				出厂日 期
manufacture_model	varchar(35)	否				型号
owner_id	varchar(25)	否			外 键	车主 ID
plate_number	varchar(20)	是	UNI	NULL		车牌
status	varchar(8)	否				状态
vehicle_type	varchar(2)	否				车型

表 3-5：行驶证信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
issue_date	date	否				发证日期
registration_date	date	否				注册日期
usage	varchar(20)	否				使用性质
vehicle_id	varchar(25)	否			主键 外键	行驶证号

表 3-6：违章信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
area	varchar(50)	否				区域
date	datetime(6)	否				日期
deadline	date	否				缴纳截止日期
driver_id	varchar(25)	否			外键	违章驾驶员 ID
fine	decimal(5, 2)	否				罚款
id	bigint	否			主键	违章 ID
is_processed	tinyint(1)	否				罚款缴纳状态
location	varchar(50)	否				地点
point_minus	int unsigned	否				扣分
type	varchar(100)	否				类型
vehicle_id	varchar(25)	否			外键	违章机动车 ID

表 3-7：违章处理记录表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
process_time	datetime(6)	否				处理时间
violation_id	bigint	否			主键 外键	违章 ID

表 3-8：道路安全学习信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
create_time	datetime(6)	否				创建时间
driver_id	varchar(25)	否			外键	驾驶员 ID
id	bigint	否			主键	学习 ID
is_finished	tinyint(1)	否				学习状态

表 3-9：道路安全学习记录表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
education_id	bigint	否			主键 外键	学习 ID
finish_time	datetime(6)	否				学习完成时间

表 3-10：用户组信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
id	int	否			主键	用户组 ID
name	varchar(150)	否	UNI			名称

表 3-11：用户所属用户组信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
group_id	int	否			外键	用户组 ID
id	bigint	否			主键	
user_id	int	否			外键	用户 ID

表 3-12：权限信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
codename	varchar(100)	否				供系统识别的权限名称

content_type_id	int	否			外键	访问内容类别 ID
id	int	否			主键	权限 ID
name	varchar(255)	否				便于用户理解的权限别名

表 3-13：用户组权限表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
group_id	int	否			外键	用户组 ID
id	bigint	否			主键	
permission_id	int	否			外键	权限 ID

表 3-14：用户权限表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
id	bigint	否			主键	
permission_id	int	否			外键	权限 ID
user_id	int	否			外键	用户 ID

表 3-15：内容类别表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
app_label	varchar(100)	否				内容所属子系统
id	int	否			主键	内容 ID
model	varchar(100)	否				内容

表 3-16：管理员/职员账户后台操作记录表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
-----	------	------	-----	-----	---	----

action_flag	smallint unsigned	否				操作类型
action_time	datetime(6)	否				记录时间
change_message	longtext	否				操作详细信息
content_type_id	int	是		NULL	外键	访问内容类别 ID
id	int	否			主键	记录 ID
object_id	longtext	是		NULL		操作对象 ID
object_repr	varchar(200)	否				操作对象名称
user_id	int	否			外键	操作用户 ID

## 四. 数据库物理设计和实施

### 1. 数据库的创建

使用华为 GaussDB(for MySQL) 云数据库建立道路交通违章信息管理系统的数据库。数据库基本信息如表 4-1 所示：

表 4-1：数据库基本信息

参数	值
数据库名	user002db
字符集	utf8mb4
排序规则	utf8mb4_0900_ai_ci

### 2. 创建基本表

创建基本表的语句如下所示

```
CREATE TABLE `auth_group` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(150) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_0900_ai_ci NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `name` (`name`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
  
CREATE TABLE `auth_group_permissions` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `group_id` int(11) NOT NULL,  
  `permission_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY  
`auth_group_permissions_group_id_permission_id_0cd325b0_uniq`  
(`group_id`,`permission_id`),  
  KEY `auth_group_permissio_permission_id_84c5c92e_fk_auth_perm`  
(`permission_id`),  
  CONSTRAINT  
`auth_group_permissio_permission_id_84c5c92e_fk_auth_perm`  
FOREIGN KEY (`permission_id`) REFERENCES `auth_permission`  
(`id`),
```

```

CONSTRAINT
`auth_group_permissions_group_id_b120cbf9_fk_auth_group_id`
FOREIGN KEY (`group_id`) REFERENCES `auth_group` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=59 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `auth_permission` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `content_type_id` int(11) NOT NULL,
  `codename` varchar(100) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY
`auth_permission_content_type_id_codename_01ab375a_uniq`
(`content_type_id`,`codename`),
  CONSTRAINT
`auth_permission_content_type_id_2f476e4b_fk_django_co` FOREIGN
KEY (`content_type_id`) REFERENCES `django_content_type` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=73 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `auth_user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `password` varchar(128) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `last_login` datetime(6) DEFAULT NULL,
  `is_superuser` tinyint(1) NOT NULL,
  `username` varchar(150) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `first_name` varchar(150) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `last_name` varchar(150) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `email` varchar(254) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
  `is_staff` tinyint(1) NOT NULL,
  `is_active` tinyint(1) NOT NULL,
  `date_joined` datetime(6) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

```

CREATE TABLE `auth_user_groups` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `group_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `auth_user_groups_user_id_group_id_94350c0c_uniq`
  (`user_id`,`group_id`),
  KEY `auth_user_groups_group_id_97559544_fk_auth_group_id`
  (`group_id`),
  CONSTRAINT `auth_user_groups_group_id_97559544_fk_auth_group_id`
  FOREIGN KEY (`group_id`) REFERENCES `auth_group` (`id`),
  CONSTRAINT `auth_user_groups_user_id_6a12ed8b_fk_auth_user_id`
  FOREIGN KEY (`user_id`) REFERENCES `auth_user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `auth_user_user_permissions` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `permission_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY
  `auth_user_user_permissions_user_id_permission_id_14a6b632_uniq`
  (`user_id`,`permission_id`),
  KEY `auth_user_user_permi_permission_id_1fbb5f2c_fk_auth_perm`
  (`permission_id`),
  CONSTRAINT
  `auth_user_user_permi_permission_id_1fbb5f2c_fk_auth_perm`
  FOREIGN KEY (`permission_id`) REFERENCES `auth_permission`
  (`id`),
  CONSTRAINT
  `auth_user_user_permissions_user_id_a95ead1b_fk_auth_user_id`
  FOREIGN KEY (`user_id`) REFERENCES `auth_user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `django_admin_log` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `action_time` datetime(6) NOT NULL,
  `object_id` longtext CHARACTER SET utf8mb4 COLLATE
  utf8mb4_0900_ai_ci,
  `object_repr` varchar(200) CHARACTER SET utf8mb4 COLLATE
  utf8mb4_0900_ai_ci NOT NULL,

```



```

`action_flag` smallint(5) unsigned NOT NULL,
`change_message` longtext CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
`content_type_id` int(11) DEFAULT NULL,
`user_id` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `django_admin_log_content_type_id_c4bce8eb_fk_django_co`
(`content_type_id`),
KEY `django_admin_log_user_id_c564eba6_fk_auth_user_id`
(`user_id`),
CONSTRAINT
`django_admin_log_content_type_id_c4bce8eb_fk_django_co` FOREIGN
KEY (`content_type_id`) REFERENCES `django_content_type` (`id`),
CONSTRAINT `django_admin_log_user_id_c564eba6_fk_auth_user_id`
FOREIGN KEY (`user_id`) REFERENCES `auth_user` (`id`),
CONSTRAINT `django_admin_log_chk_1` CHECK ((`action_flag` >= 0))
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `django_content_type` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`app_label` varchar(100) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
`model` varchar(100) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `django_content_type_app_label_model_76bd3d3b_uniq`
(`app_label`,`model`)
) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_driverlicense` (
`user_profile_id` varchar(25) NOT NULL,
`issue_date` date NOT NULL,
`motor_type` varchar(3) NOT NULL,
`begin_date` date NOT NULL,
`valid_duration` int(10) unsigned NOT NULL,
`points` int(11) NOT NULL,
`status` varchar(5) NOT NULL,
PRIMARY KEY (`user_profile_id`),
CONSTRAINT
`TrafficMan_driverlic_user_profile_id_143c3701_fk_TrafficMa`
FOREIGN KEY (`user_profile_id`) REFERENCES
`trafficman_userprofile` (`identity`),

```

```

CONSTRAINT `trafficman_driverlicense_chk_1` CHECK
((`valid_duration` >= 0))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_education` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `create_time` datetime(6) NOT NULL,
  `is_finished` tinyint(1) NOT NULL,
  `driver_id` varchar(25) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `TrafficMan_education_driver_id_a328707e_fk_TrafficMa`
  (`driver_id`),
  CONSTRAINT
  `TrafficMan_education_driver_id_a328707e_fk_TrafficMa` FOREIGN
  KEY (`driver_id`) REFERENCES `trafficman_userprofile`
  (`identity`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_educationrecord` (
  `finish_time` datetime(6) NOT NULL,
  `education_id` bigint(20) NOT NULL,
  PRIMARY KEY (`education_id`),
  CONSTRAINT
  `TrafficMan_education_education_id_d56e8c7b_fk_TrafficMa` FOREIGN
  KEY (`education_id`) REFERENCES `trafficman_education` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_userprofile` (
  `identity` varchar(25) NOT NULL,
  `name` varchar(80) NOT NULL,
  `gender` varchar(5) NOT NULL,
  `ethnicity` varchar(20) NOT NULL,
  `nationality` varchar(10) NOT NULL,
  `address` varchar(200) NOT NULL,
  `telephone` varchar(15) NOT NULL,
  `mobile` varchar(25) NOT NULL,
  `birth` date NOT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`identity`),
  UNIQUE KEY `user_id` (`user_id`),

```

```

CONSTRAINT
`TrafficMan_userprofile_user_id_346b9bda_fk_auth_user_id` FOREIGN
KEY (`user_id`) REFERENCES `auth_user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_vehicle` (
  `engine_id` varchar(25) NOT NULL,
  `brand` varchar(25) NOT NULL,
  `manufacture_model` varchar(35) NOT NULL,
  `color` varchar(20) NOT NULL,
  `vehicle_type` varchar(2) NOT NULL,
  `displacement` decimal(5,2) NOT NULL,
  `manufacture_date` date NOT NULL,
  `life_duration` int(10) unsigned NOT NULL,
  `plate_number` varchar(20) DEFAULT NULL,
  `status` varchar(8) NOT NULL,
  `owner_id` varchar(25) NOT NULL,
  PRIMARY KEY (`engine_id`),
  UNIQUE KEY `plate_number` (`plate_number`),
  KEY `TrafficMan_vehicle_owner_id_68f4ee54_fk_TrafficMa`
  (`owner_id`),
  CONSTRAINT `TrafficMan_vehicle_owner_id_68f4ee54_fk_TrafficMa`
  FOREIGN KEY (`owner_id`) REFERENCES `trafficman_userprofile`
  (`identity`),
  CONSTRAINT `trafficman_vehicle_chk_1` CHECK ((`life_duration` >=
  0))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_vehiclelicense` (
  `vehicle_id` varchar(25) NOT NULL,
  `usage` varchar(20) NOT NULL,
  `registration_date` date NOT NULL,
  `issue_date` date NOT NULL,
  PRIMARY KEY (`vehicle_id`),
  CONSTRAINT
  `TrafficMan_vehicleli_vehicle_id_5400c632_fk_TrafficMa` FOREIGN
  KEY (`vehicle_id`) REFERENCES `trafficman_vehicle` (`engine_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_violation` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,

```

```

`date` datetime(6) NOT NULL,
`type` varchar(100) NOT NULL,
`area` varchar(50) NOT NULL,
`location` varchar(50) NOT NULL,
`point_minus` int(10) unsigned NOT NULL,
`fine` decimal(5,2) NOT NULL,
`deadline` date NOT NULL,
`is_processed` tinyint(1) NOT NULL,
`driver_id` varchar(25) NOT NULL,
`vehicle_id` varchar(25) NOT NULL,
PRIMARY KEY (`id`),
KEY `TrafficMan_violation_driver_id_0dd0c9dc_fk_TrafficMa`
(`driver_id`),
KEY `TrafficMan_violation_vehicle_id_a69985f8_fk_TrafficMa`
(`vehicle_id`),
CONSTRAINT
`TrafficMan_violation_driver_id_0dd0c9dc_fk_TrafficMa` FOREIGN
KEY (`driver_id`) REFERENCES `trafficman_userprofile`
(`identity`),
CONSTRAINT
`TrafficMan_violation_vehicle_id_a69985f8_fk_TrafficMa` FOREIGN
KEY (`vehicle_id`) REFERENCES `trafficman_vehicle` (`engine_id`),
CONSTRAINT `trafficman_violation_chk_1` CHECK ((`point_minus` >=
0))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `trafficman_violationprocess` (
  `process_time` datetime(6) NOT NULL,
  `violation_id` bigint(20) NOT NULL,
  PRIMARY KEY (`violation_id`),
  CONSTRAINT
  `TrafficMan_violation_violation_id_7ff9dd18_fk_TrafficMa` FOREIGN
  KEY (`violation_id`) REFERENCES `trafficman_violation` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

### 3. 触发器设计

当一条违章记录被插入时，与这一违章关联的驾驶员的驾驶证信息和机动车信息都需要被更新。如下触发器 `t_violationInsertTrigger` 实现了这一功能。如果这一违章后驾驶员驾驶证扣分已经超过 12 分，则将其驾驶证分数设为 0，状态设为超分，并在道路安全学习信息表中添加与该驾驶员关联的学习信

息，如果在违章记录插入时罚款已经缴纳，则在违章处理记录表中添加与该违章相关的记录，否则，将与该违章关联的机动车状态设置为违法未处理；如果扣分没有超过 12 分，则直接扣除驾驶证分数，如果在违章记录插入时罚款已经缴纳，则在违章处理记录表中添加与该违章相关的记录，否则，将与该违章关联的驾驶员的驾驶证状态和机动车状态设置为违法未处理。

```
DELIMITER $$
CREATE DEFINER=`DB_USER002`@`%` TRIGGER
`t_violationInsertTrigger` AFTER INSERT ON `trafficman_violation`
FOR EACH ROW BEGIN
    DECLARE d_points INT;
    SELECT points INTO d_points
    FROM TrafficMan_driverlicense
    WHERE TrafficMan_driverlicense.user_profile_id =
NEW.driver_id;

    IF CAST(d_points AS SIGNED)-CAST(NEW.point_minus AS SIGNED)<=0
    THEN
        UPDATE TrafficMan_driverlicense
        SET points=0, status='B'
        WHERE TrafficMan_driverlicense.user_profile_id =
NEW.driver_id;

        INSERT INTO TrafficMan_education(create_time, is_finished,
driver_id)
        VALUES(NOW(), 0, NEW.driver_id);

        IF NEW.is_processed = 0 THEN
            UPDATE TrafficMan_vehicle
            SET status = 'G'
            WHERE TrafficMan_vehicle.engine_id=NEW.vehicle_id;
        ELSE
            INSERT INTO TrafficMan_violationprocess
            VALUES (NOW(),NEW.id);
        END IF;
    ELSE
        UPDATE TrafficMan_driverlicense
        SET points=d_points-NEW.point_minus
        WHERE TrafficMan_driverlicense.user_profile_id =
NEW.driver_id;

        IF NEW.is_processed=0 THEN
            UPDATE TrafficMan_vehicle
            SET status = 'G'
            WHERE TrafficMan_vehicle.engine_id=NEW.vehicle_id;
```

```

        UPDATE TrafficMan_driverlicense
        SET status='H'
        WHERE TrafficMan_driverlicense.user_profile_id =
NEW.driver_id;
    ELSE
        INSERT INTO TrafficMan_violationprocess
        VALUES (NOW(),NEW.id);
    END IF;
END IF;
END $$
DELIMITER ;

```

当把一条违章记录的罚款缴纳状态更改为已缴纳时，则自动在违章处理记录表中添加与该违章相关的记录，如果与该违章相关联的机动车状态和驾驶员的驾驶证状态是违法未处理，则将其状态设置为正常。如果违章处理记录表中已经有该违章的处理记录，则不进行上述修改。如下触发器 t\_violationUpdateTrigger 实现了这一功能。

```

DELIMITER $$
CREATE DEFINER=`DB_USER002`@`%` TRIGGER
`t_violationUpdateTrigger` AFTER UPDATE ON `trafficman_violation`
FOR EACH ROW BEGIN
    DECLARE process_count INT;
    SELECT count(*) INTO process_count
    FROM TrafficMan_violationprocess
    WHERE TrafficMan_violationprocess.violation_id=NEW.id;

    IF process_count=0 THEN
        IF NEW.is_processed=1 THEN
            INSERT INTO TrafficMan_violationprocess
            VALUES (NOW(),NEW.id);
            UPDATE TrafficMan_driverlicense
            SET status='A'
            WHERE
TrafficMan_driverlicense.user_profile_id=NEW.driver_id AND
TrafficMan_driverlicense.status='H';
            UPDATE TrafficMan_vehicle
            SET status='A'
            WHERE TrafficMan_vehicle.engine_id=NEW.vehicle_id AND
TrafficMan_vehicle.status='G';
        END IF;
    END IF;
END $$
DELIMITER ;

```

如果一条道路安全学习信息的学习状态被设定为已完成，则自动在道路安全学习记录表中添加相关记录。如果与该学习信息相关联的驾驶员的驾驶证状

态为超分，则设置为正常。然后，将与该学习信息相关联的驾驶员的驾驶证分数重置为 12 分。如果道路安全学习记录表中已经有该学习信息的完成记录，则不进行上述修改。如下触发器 t\_educationUpdateTrigger 实现了这一功能。

```
DELIMITER $$
CREATE DEFINER=`DB_USER002`@`%` TRIGGER
`t_educationUpdateTrigger` AFTER UPDATE ON `trafficman_education`
FOR EACH ROW BEGIN
    DECLARE record_count INT;
    SELECT count(*) INTO record_count
    FROM TrafficMan_educationrecord
    WHERE TrafficMan_educationrecord.education_id = NEW.id;

    IF NEW.is_finished=1 AND record_count=0 THEN
        INSERT INTO TrafficMan_educationrecord
        VALUES (NOW(), NEW.id);

        UPDATE TrafficMan_driverlicense
        SET status='A'
        WHERE
TrafficMan_driverlicense.user_profile_id=NEW.driver_id AND
TrafficMan_driverlicense.status='B';
        UPDATE TrafficMan_driverlicense
        SET points=12
        WHERE
TrafficMan_driverlicense.user_profile_id=NEW.driver_id;
    END IF;
END $$
DELIMITER ;
```

如果一个机动车因报废等原因被注销或者强制注销，其车牌号码会被释放，下述触发器 t\_closeVehicleTrigger 实现了这一功能。

```
DELIMITER $$
CREATE DEFINER=`DB_USER002`@`%` TRIGGER `t_closeVehicleTrigger`
BEFORE UPDATE ON `trafficman_vehicle` FOR EACH ROW BEGIN
    IF NEW.status='E' OR NEW.status='M' THEN
        SET NEW.plate_number=null;
    END IF;
END $$
DELIMITER ;
```

#### 4. 存储过程设计

如果驾驶证过期满一年还未办理换发手续，则驾驶证会被吊销。同时，如果状态为正常的驾驶证已经到达记分周期，则需要将驾驶证分数重置为 12 分。下述存储过程 p\_SetDriverLicenseStatus 实现了这一功能。

```
DELIMITER $$
CREATE DEFINER=`DB_USER002`@`%` PROCEDURE
`p_SetDriverLicenseStatus`()
BEGIN
    UPDATE TrafficMan_driverlicense
        SET status='F'
        WHERE
TIMESTAMPDIFF(YEAR,begin_date,CURDATE())>=valid_duration+1 AND
status<>'F';

    UPDATE TrafficMan_driverlicense
        SET points=12
        WHERE MONTH(begin_date)=MONTH(CURDATE()) AND
DAY(begin_date)=DAY(CURDATE()) AND status='A';
END $$
DELIMITER ;
```

#### 5. 视图设计

逾期缴纳罚款的违章会影响到机动车年检。视图 v\_exceededviolation 中包含了逾期缴纳罚款的违章的详细信息、关联驾驶员的信息、关联机动车的车主的信息。其中，逾期缴纳罚款的违章包含已过缴纳截止日期但至今尚未缴纳罚款的违章，以及在缴纳截止日期之后缴纳罚款的违章。

```
CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW
`v_exceededviolation` AS
select `violation`.`id` AS `id`,`violation`.`date` AS
`date`,`violation`.`location` AS `location`,`violation`.`type` AS
`type`,`violation`.`deadline` AS `deadline`,`violation`.`fine` AS
`fine`,`vehicle`.`plate_number` AS
`plate_number`,`violation`.`driver_id` AS
`driver_id`,`driver`.`name` AS `driver_name`,`owner`.`identity`
AS `owner_id`,`owner`.`name` AS
`owner_name`,`violation`.`is_processed` AS `is_processed`
from (((`trafficman_violation` `violation` join
`trafficman_vehicle` `vehicle`) join `trafficman_userprofile`
`driver`) join `trafficman_userprofile` `owner`)
```



```
where ((`violation`.`vehicle_id` = `vehicle`.`engine_id`) and
(`violation`.`driver_id` = `driver`.`identity`) and
(`vehicle`.`owner_id` = `owner`.`identity`) and (curdate() >
`violation`.`deadline`) and (exists(select 1 from
`trafficman_violationprocess` where (`violation`.`id` =
`trafficman_violationprocess`.`violation_id`)) is false or
((select `trafficman_violationprocess`.`process_time` from
`trafficman_violationprocess` where
(`trafficman_violationprocess`.`violation_id` =
`violation`.`id`)) > `violation`.`deadline`)));
```

管理人员可能会查看至今尚未缴纳罚款的违章信息，视图  
v\_unprocessedviolation 包含了至今尚未缴纳罚款的违章的详细信息、关联驾  
驶员的信息、关联机动车的车主的信息。

```
CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW
`v_unprocessedviolation` AS
select `trafficman_violation`.`id` AS
`id`,`trafficman_violation`.`date` AS
`date`,`trafficman_violation`.`area` AS
`area`,`trafficman_violation`.`type` AS
`type`,`trafficman_violation`.`deadline` AS
`deadline`,`trafficman_violation`.`fine` AS
`fine`,`trafficman_vehicle`.`plate_number` AS
`plate_number`,`trafficman_violation`.`driver_id` AS
`driver_id`,`driver`.`name` AS `driver_name`,`own`.`identity` AS
`owner_id`,`own`.`name` AS
`owner_name`,`trafficman_violation`.`is_processed` AS
`is_processed`
from (((`trafficman_violation` join `trafficman_vehicle`) join
`trafficman_userprofile` `driver`) join `trafficman_userprofile`
`own`)
where ((`trafficman_violation`.`is_processed` = 0) and
(`trafficman_violation`.`driver_id` = `driver`.`identity`) and
(`trafficman_violation`.`vehicle_id` =
`trafficman_vehicle`.`engine_id`) and
(`trafficman_vehicle`.`owner_id` = `own`.`identity`));
```

管理人员可能会查看违章的罚款缴纳记录，视图  
v\_violationprocessrecord 包含了违章罚款缴纳记录信息，及与该记录关联的  
违章的详细信息。

```
CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW
`v_violationprocessrecord` AS
select `v`.`id` AS `id`,`v`.`date` AS `date`,`v`.`type` AS
`type`,`v`.`area` AS `area`,`v`.`location` AS
`location`,`v`.`point_minus` AS `point_minus`,`v`.`fine` AS
`fine`,`v`.`deadline` AS `deadline`,`v`.`is_processed` AS
```

```
`is_processed`,`v`.`driver_id` AS `driver_id`,`vh`.`plate_number`
AS `plate_number`,`vp`.`process_time` AS `process_time`
from ((`trafficman_violation` `v` join
`trafficman_violationprocess` `vp`) join `trafficman_vehicle`
`vh`)
where ((`v`.`id` = `vp`.`violation_id`) and (`v`.`vehicle_id` =
`vh`.`engine_id`));
```

将机动车和车主信息整合显示可能会方便管理人员，视图 v\_vehicleowner 包含了所有机动车的详细信息及其车主的个人信息。

```
CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW
`v_vehicleowner` AS
select `trafficman_userprofile`.`identity` AS
`identity`,`trafficman_userprofile`.`name` AS
`name`,`trafficman_userprofile`.`gender` AS
`gender`,`trafficman_userprofile`.`ethnicity` AS
`ethnicity`,`trafficman_userprofile`.`nationality` AS
`nationality`,`trafficman_userprofile`.`address` AS
`address`,`trafficman_userprofile`.`telephone` AS
`telephone`,`trafficman_userprofile`.`mobile` AS
`mobile`,`trafficman_userprofile`.`birth` AS
`birth`,`trafficman_vehicle`.`engine_id` AS
`engine_id`,`trafficman_vehicle`.`brand` AS
`brand`,`trafficman_vehicle`.`manufacture_model` AS
`manufacture_model`,`trafficman_vehicle`.`color` AS
`color`,`trafficman_vehicle`.`vehicle_type` AS
`vehicle_type`,`trafficman_vehicle`.`displacement` AS
`displacement`,`trafficman_vehicle`.`manufacture_date` AS
`manufacture_date`,`trafficman_vehicle`.`life_duration` AS
`life_duration`,`trafficman_vehicle`.`plate_number` AS
`plate_number`,`trafficman_vehicle`.`status` AS `status`
from (`trafficman_userprofile` join `trafficman_vehicle`)
where (`trafficman_userprofile`.`identity` =
`trafficman_vehicle`.`owner_id`);
```

## 6. 事件设计

存储过程 p\_SetDriverLicenseStatus 实现了驾驶证吊销以及分数重置。然而，考虑到每天都可能会有新的驾驶证过期满一年需要吊销，也会有新的驾驶证达到记分周期需要重置分数，因此这一存储过程需要每日执行。事件 Schedule\_SetDriverLicenseStatus 实现了这一功能。

```
DELIMITER $$
CREATE DEFINER=`DB_USER002`@`%` EVENT
`Schedule_SetDriverLicenseStatus` ON SCHEDULE EVERY 1 DAY STARTS
```

```
'2021-05-26 06:15:00' ON COMPLETION NOT PRESERVE ENABLE DO call
p_SetDriverLicenseStatus
; $$
DELIMITER ;
```

## 7. 索引设计

为了方便查询，提高系统效率，除了每个表由主键、外键、唯一约束创建的索引，还建立了索引如表 4-2 所示

表 4-2：索引信息

表名	字段	索引类型	索引方法
trafficman_ userprofile	name	NORMAL	BTREE
trafficman_ vehicle	status	NORMAL	BTREE
	vehicle_type	NORMAL	BTREE
	brand	NORMAL	BTREE
trafficman_ violation	is_processed	NORMAL	BTREE
	type	NORMAL	BTREE

## 五. 应用程序设计

### 1. 开发及运行环境介绍

应用程序采用 Python 语言进行开发，开发过程中采用了 Django 框架，开发环境的详细信息如表 5-1 所示：

表 5-1：开发环境

参数	值
Python 版本	3.8.5
Django 版本	3.2.3
IDE 信息	PyCharm 2020.3
操作系统	macOS 11.3 Big Sur
软件依赖	见源代码文件夹下的 requirements.txt 文件

### 2. 主要功能设计

应用程序的主要功能如图 5-1 所示

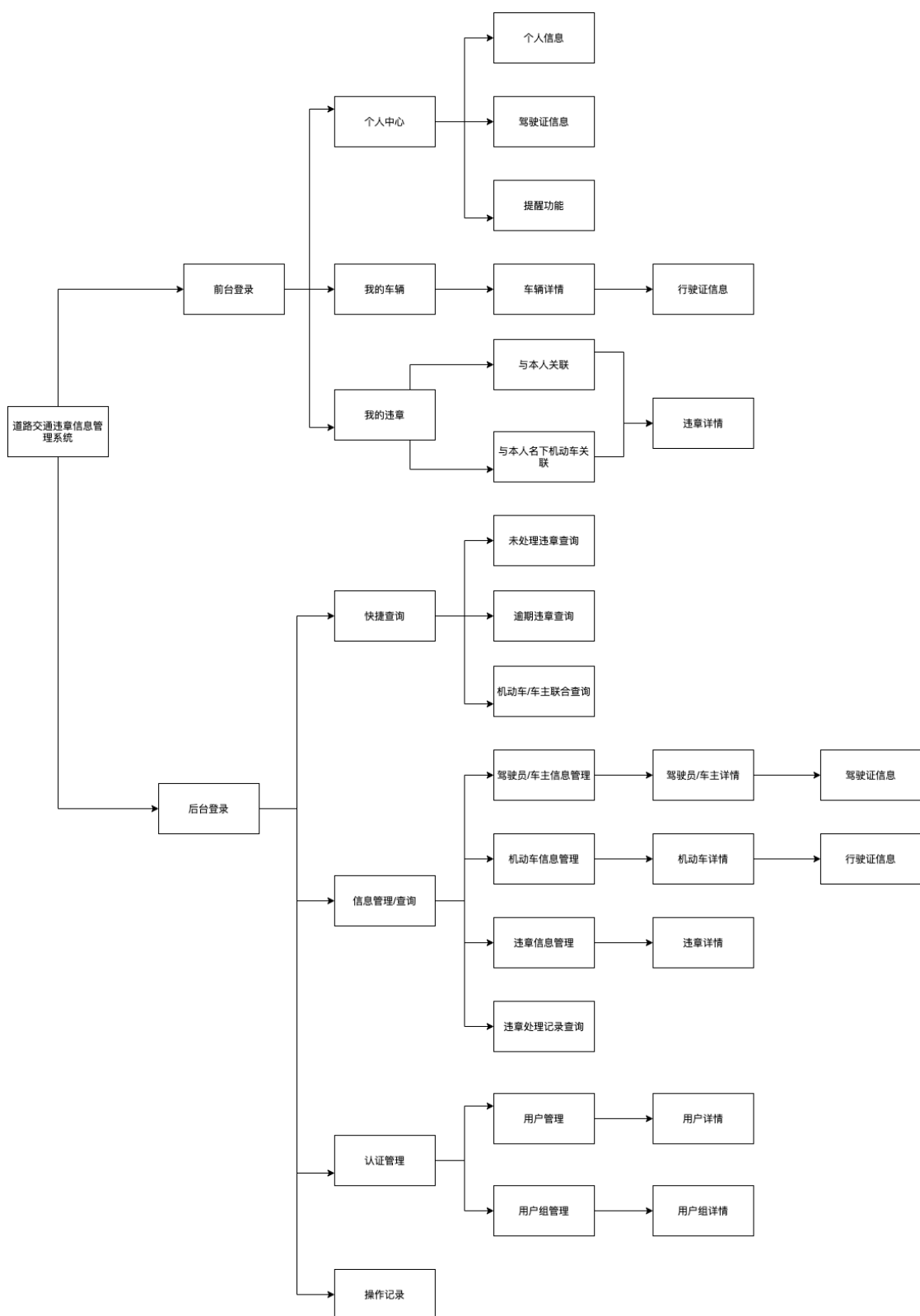


图 5-1 应用程序的主要功能

应用程序架构采用 B/S 模式，用户使用浏览器来访问应用程序，前端仅实现少部分的业务逻辑，包括数据库访问在内的大部分的业务逻辑在服务器端实现，应用程序的运行流程如图 5-2 所示

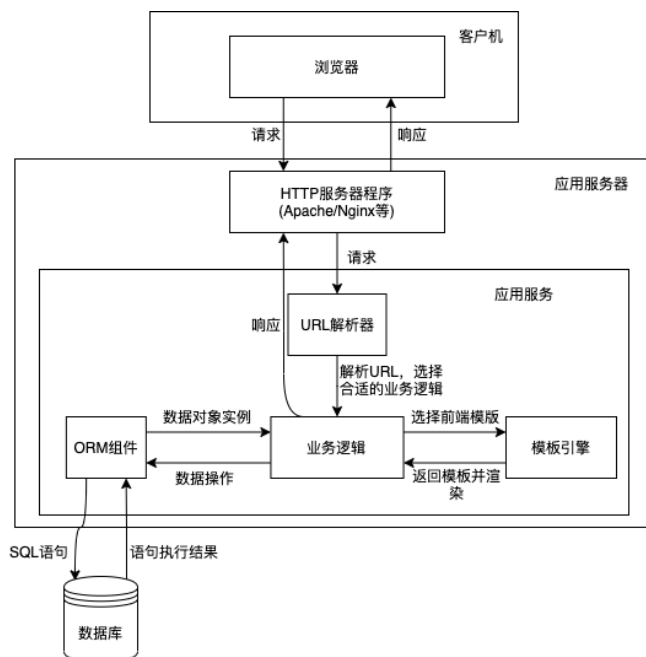


图 5-2 应用程序的运行流程示意图

应用程序开发时采用了对象关系映射(Object Relational Mapping, ORM)机制，其映射关系如图 5-3 所示

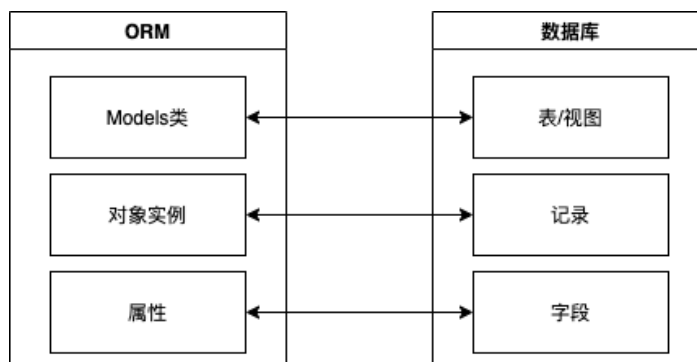


图 5-3 ORM 映射关系

ORM 将对数据库的操作转化为对类属性和方法的操作，开发时仅需面向对象编程，而不需要面向数据库编写代码，极大地提高了开发效率。同时，实现了数据模型与数据库的解耦，屏蔽了不同数据库操作上的差异，使得不同数据库可以平滑切换。

由于使用后台管理界面的用户的职位可能不尽相同，而对于不同的职位，系统管理人员希望赋予的数据访问权限可能也会不尽相同。同时，灵活的权限控制也有助于提高整个系统的安全性，在本应用程序中，预先设定了如下四个用户类别与用户组，其权限如下所示：

1. superuser: 允许登录前台和后台界面，具有所有数据的访问控制权限。
2. superior\_staff: 允许登录前台和后台界面，具有驾驶员/车主信息、机动车信息的插入、查看、修改权限，行驶证信息、驾驶证信息、违章信息的插入、删除、查看、修改权限，道路安全学习信息的查看、

修改权限，违章罚款缴纳记录、道路安全学习记录、用户认证信息、用户组信息的查看权限。

3. staff: 允许登录前台和后台界面，具有驾驶员/车主信息、机动车信息的插入、查看权限，行驶证信息、驾驶证信息的插入、删除、查看权限，违章信息的插入、查看、修改权限，道路安全学习信息的查看、修改权限，违章罚款缴纳记录、道路安全学习记录、用户认证信息、用户组信息的查看权限。

4. user: 仅允许登录前台界面，查看与用户有关的数据。

此外，在数据库中以明文方式存储用户密码是极度危险的，如果数据库信息被窃取，则用户个人的信息安全将会受到极大的威胁。

因此，在向数据库存储密码之前，本应用程序会采用带有 SHA256 哈希的 PBKDF2 算法来加密用户密码，这是一种被广泛使用的加密算法，具有较高的安全性。

### 3. 主要界面

后台登录界面如图 5-4 所示。

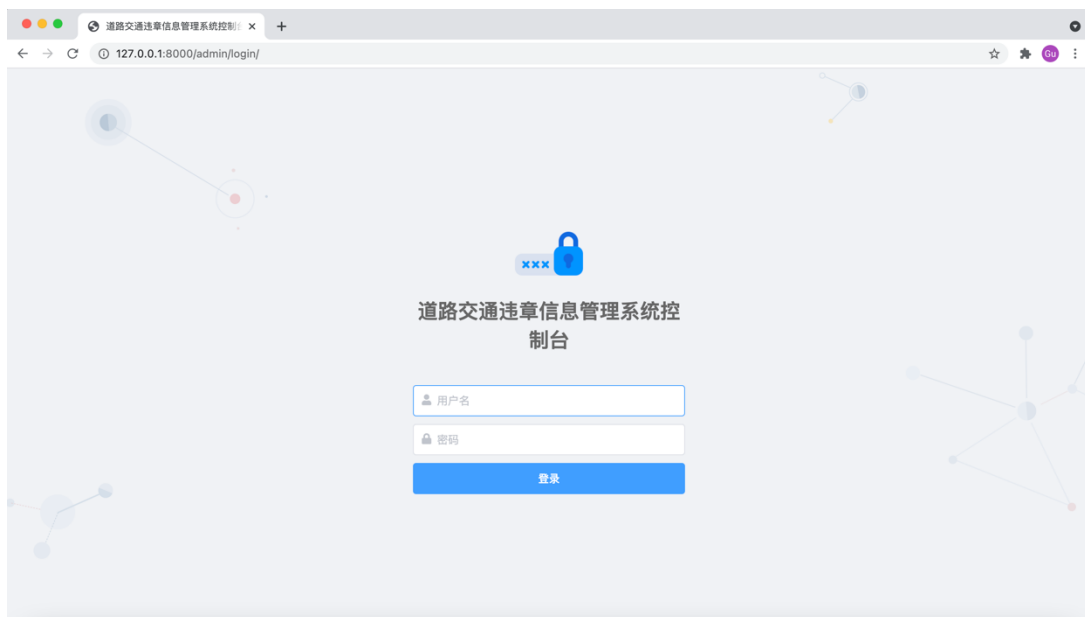


图 5-4 后台登录界面

后台主界面如图 5-5 所示，其中包含各个功能的入口，同时可以查看用户在后台管理系统的最近动作

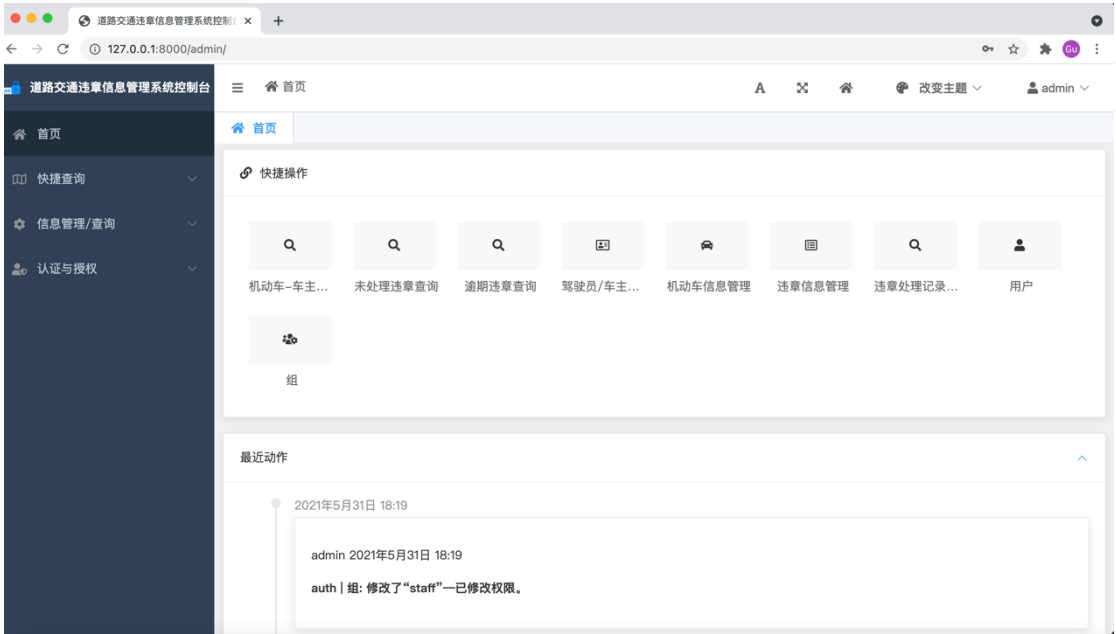


图 5-5 后台界面

快捷查询栏目旨在为了方便管理人员的常用需求，包含机动车/车主联合查询、未处理违章查询和逾期违章查询，以机动车/车主联合查询功能为例，其界面如图 5-6 所示，在界面中可以通过车主信息、机动车信息等字段搜索记录，点击链接可以查看详情，如图 5-7 所示。未处理违章查询、逾期违章查询功能与之类似。对于所有后台用户，快捷查询栏目内的所有功能以及信息管理/查询内的违章处理记录查询功能仅允许查看，不允许插入、删除、修改。

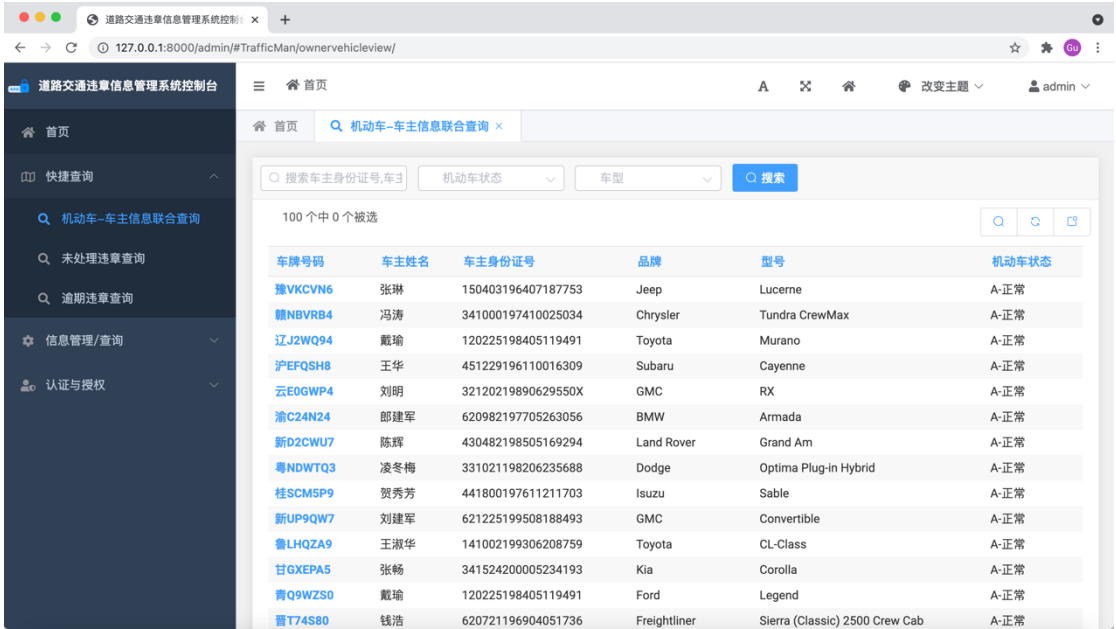


图 5-6 机动车/车主联合查询界面



首页

机动车-车主信息联合查询

返回

查看 机动车-车主信息联合查询

历史

车主身份证号:

150403196407187753

车主姓名:

张琳

性别:

男

民族:

汉

国籍:

中国

住址:

青海省北镇县静安胡街r座 246764

固定电话:

15014693111

手机号码:

18592392790

出生日期:

1969年4月1日

机动车VIN编号:

ZAVGFM8HX7U6WJEYR

图 5-7 机动车/车主联合查询详情界面

信息管理/查询栏目旨在使后台用户能够查看并管理各种信息，包含驾驶员/车主信息管理、机动车信息管理、违章信息管理和违章处理记录查询，对数据的操作权限视各个用户和用户组的权限而定。以驾驶员/车主信息管理为例，其界面如图 5-8 所示，在界面中可以通过个人信息搜索驾驶员/车主信息。点击链接可以查看详情，如图 5-9 所示，在详情界面可以管理驾驶员/车主的个人信息及其驾驶证信息，为了方便管理人员，详情界面还包括了驾驶员/车主账户名下车辆的列表、驾驶员/车主的道路安全学习信息列表，以及与驾驶员/车主个人关联的违章列表。机动车信息管理和违章信息管理与之类似。

首页

驾驶员/车主信息管理

搜索姓名,身份证号

搜索

+ 增加

删除

100 个中 0 个被选

搜索

刷新

打印

<input type="checkbox"/>	姓名	身份证号	性别	手机号码
<input type="checkbox"/>	颜涛	654326196701274090	男	13168257988
<input type="checkbox"/>	杨秀英	654325198804191581	女	15274155369
<input type="checkbox"/>	黄燕	654201195109236979	男	13395102166
<input type="checkbox"/>	朱利	652825195305038015	男	15635624369
<input type="checkbox"/>	黄莹	652702196512206127	男	15588345134
<input type="checkbox"/>	徐欣	640100199110071519	男	14516958807
<input type="checkbox"/>	徐亮	630222200005175081	男	15602908504
<input type="checkbox"/>	贾玉华	630102196001191893	男	18226053053
<input type="checkbox"/>	刘建军	621225199508188493	男	18965649871
<input type="checkbox"/>	郎建军	620982197705263056	女	13851177450
<input type="checkbox"/>	韩帆	620923196610098735	男	18052799335
<input type="checkbox"/>	韩畅	620800197012090244	女	15589720623
<input type="checkbox"/>	钱浩	620721196904051736	女	14557191923
<input type="checkbox"/>	李艳	620122196301182175	女	13179009038

图 5-8 驾驶员/车主信息管理界面

民族:

汉

国籍:

中国

住址:

湖北省丽娟县孝南张街e座 261638

固定电话:

14558221471

手机号码:

13168257988

出生日期:

1965/06/18

今天 | 龄

驾驶证信息

驾驶证信息: 颜涛(654326196701274090)

驾驶证初次领证日期:

2007/11/01

今天 | 龄

驾驶证准驾车型:

A2-牵引车

▼

驾驶证有效起始日期:

2018/10/30

今天 | 龄

驾驶证有效年限 (年) :

10

驾驶证分数:

12

驾驶证状态:

A-正常

▼

道路交通安全学习信息管理

是否完成

登记时间

机动车信息管理

车牌号码

机动车VIN编号

津NR2X97/颜涛(654326196701274090)/J635EMS72HWP0BD81

查看

津NR2X97

J635EMS72HWP0BD81

违章信息管理

车牌号/车主/机动车VIN号

违章时间

位置

扣分

图 5-9 驾驶员/车主信息管理详情界面

同样地，在机动车信息管理详情界面可以管理机动车的详细信息及其行驶证信息，为了方便管理人员，详情界面还包括了与机动车关联的违章列表。

认证管理栏目包含了用户管理和用户组管理，可以管理用户的密码等基本认证信息和权限信息，以及用户组的权限信息，如图 5-10 所示。

← 返回

修改 用户

用户名:

130200196507020842

必填：长度为150个字符或以下；只能包含字母、数字、特殊字符"@","-","."和"\_"。

密码:

算法: pbkdf2\_sha256 迭代次数: 260000 盐: Ro9y4U\*\*\*\*\* 哈希: gM1YF9\*\*\*\*\*

密码原文未存储在系统中，因此无法看到该用户的密码。然而你可以通过[这个表单](#)来修改密码。

个人信息

名字:

姓氏:

电子邮件地址:

权限

☒ 有效

指明用户是否被认为是活跃的。以反选代替删除帐号。

☐ 工作人员状态

指明用户是否可以登录到这个管理站点。

☐ 超级用户状态

指明该用户缺省拥有所有权限。

图 5-10 用户管理详情界面

对于不同类别的账户，后台操作的权限也有所不同。例如，用户类别为 staff 的驾驶员/车主信息管理详情界面如图 5-11 所示，可以看到，字段不允许修改，而用户类别为 superior\_staff 与 superuser 的用户则无此限制。详细的权限配置信息请查阅上文说明。

[← 返回](#) | [查看 驾驶员/车主信息管理](#) 🕒 历史


用户名:	654326196701274090
身份证号:	654326196701274090
姓名:	颜涛
性别:	男
民族:	汉
国籍:	中国
住址:	湖北省丽娟县孝南张街e座 261638
固定电话:	14558221471
手机号码:	13168257988
出生日期:	1965年6月18日

**驾驶证信息**  
驾驶证信息: 颜涛(654326196701274090)

图 5-11 staff 用户类别的驾驶员/车主信息管理详情界面

前台登录界面如图 5-12 所示

[←](#) [→](#) [🔄](#) 127.0.0.1:8000/auth/?next=/dashboard/ ☆ 🌙 🌞 ☰



道路交通违章信息管理系统-登录

用户名

密码

登录

图 5-12 前台登录界面

前台个人中心界面如图 5-13 所示，登录后会显示本用户的个人信息和驾驶证信息，对于驾驶证即将过期或者已经过期的用户，以及驾驶证分数被扣完的用户，也会在这个页面进行提醒。



图 5-13 前台个人中心

我的车辆界面如图 5-14 所示，列表显示了这个用户名下的所有车辆，点击链接可以查看车辆的详细信息和行驶证信息，如图 5-15 所示。同时，也可以使用车牌号码和品牌字段进行搜索。

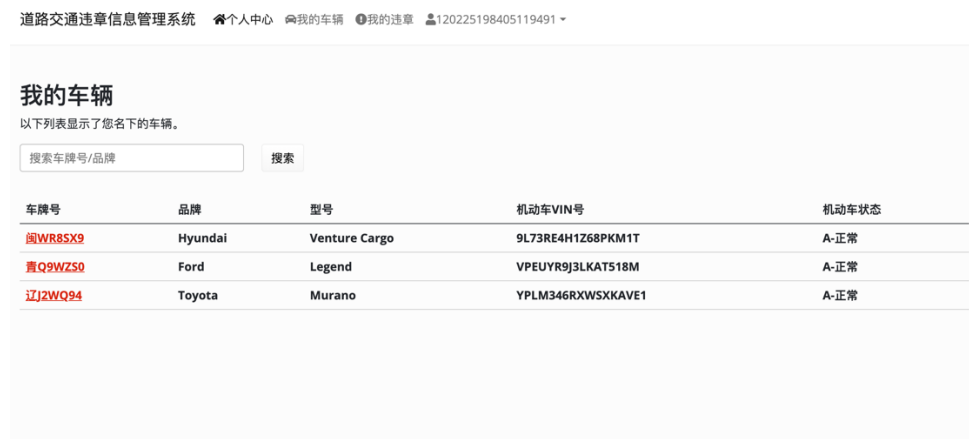


图 5-14 我的车辆界面

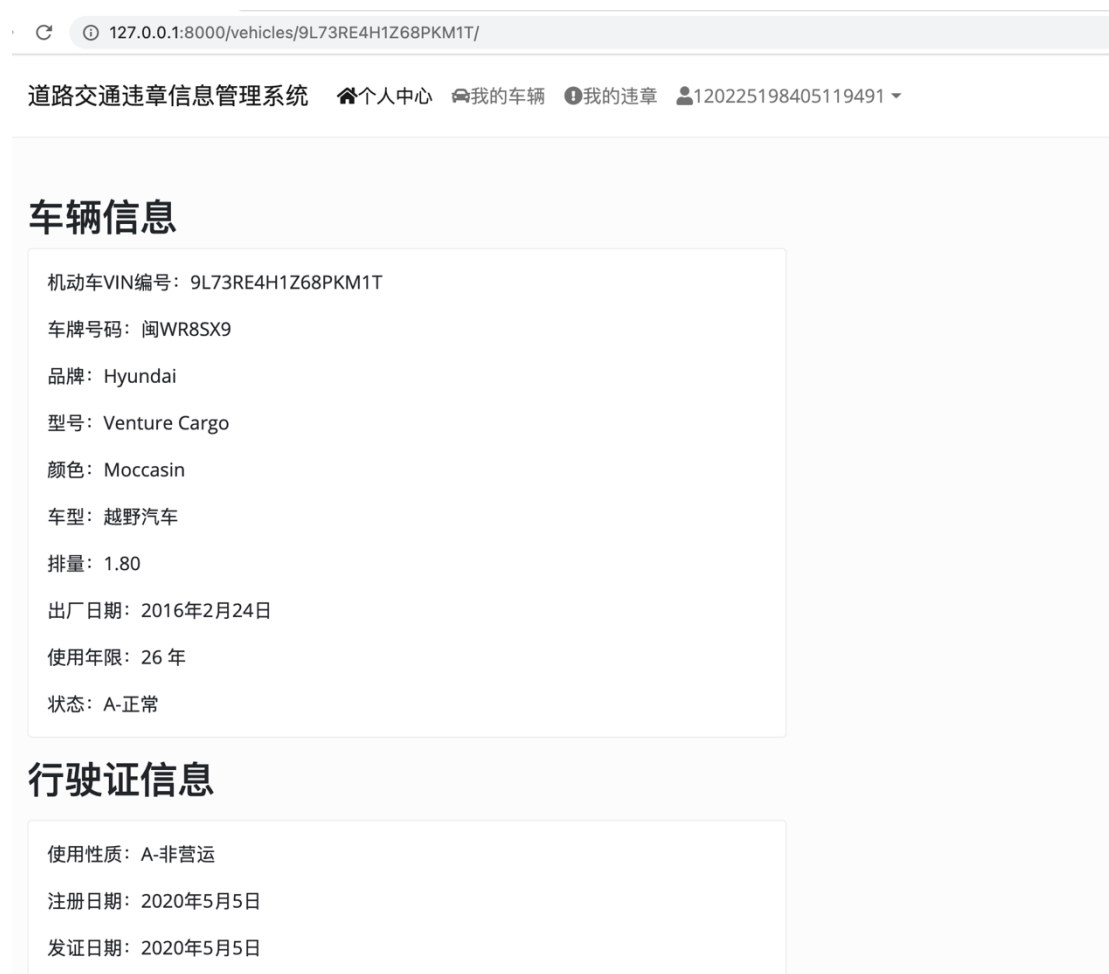


图 5-15 车辆详细信息界面

如果用户试图通过修改 URL 访问不属于该用户的车辆详情，则访问会被拒绝，如图 5-16 所示。

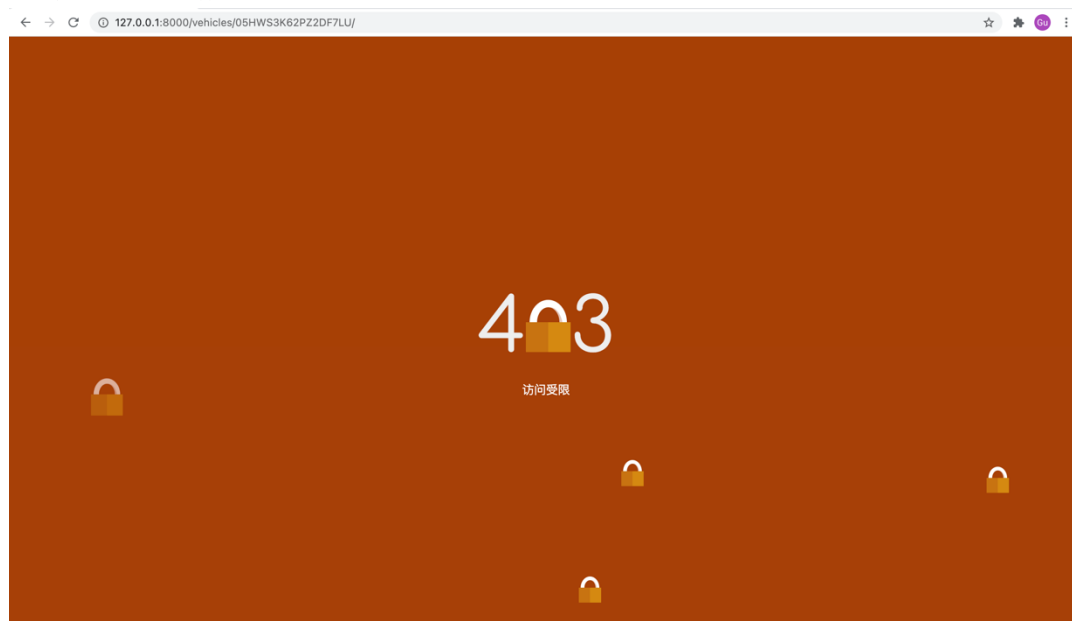


图 5-16 拒绝访问界面

我的违章界面如图 5-17 所示，列表分类显示了与这个用户以及与该用户名下的所有车辆关联的违章记录。点击链接可以查看违章的详细信息，如图 5-18 所示，详情界面中也包含了驾驶员和车主的姓名和联系方式，方便车主与驾驶员信息不一致的情况下双方进行协商。

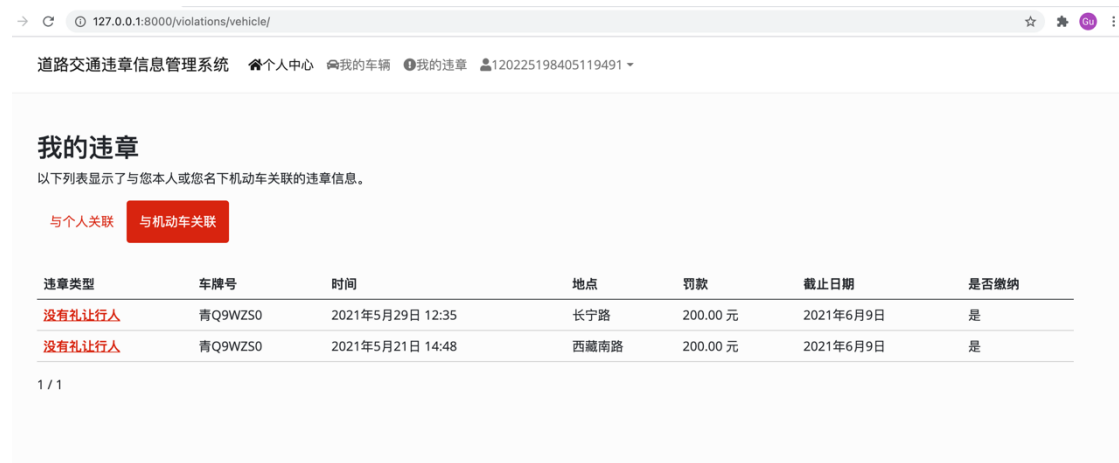


图 5-17 我的违章界面



图 5-18 违章详情界面

同样地，如果用户试图访问与该用户无关的违章信息，访问亦会被拒绝。

## 六. 心得体会

在本次实践中，由于严格遵循了完善的软件开发流程与开发规范，同时得益于易用的软件开发工具和成熟的软件开发框架，因此并没有遇到特别大的困难，实践过程较为顺利。

通过本次实践，深入了解了数据库系统的开发规范和开发流程，熟悉了 B/S 架构应用程序的基本运行流程，初步掌握了华为 GaussDB (for MySQL) 数据库的使用方法以及利用 Django 框架开发 Web 应用程序的基本方法。

此外，在本次实践中，ORM 机制的引入也令我们印象深刻，其建立起了关系型数据库与面向对象编程语言之间的桥梁，使得我们能够更加便捷地在 Python 编程语言中操作数据库中的数据，ORM 机制的引入极大地提高了软件的开发效率。