Police Traffic System- Technical Manual

This website uses PHP and MySQL to develop and manage traffic information of police officers, people, vehicles, ownership, fines, incidents, and offences. Administrators can manage users, fines inserted, and the history of every move by police officers.

Setup:

- URL : http://localhost/cw2/index.php.
- Download and extract the zip file to the cw2 folder.
- Insert the only SQL file to the MySQL database.
- Restart the web server to access at http://localhost/coursework2.

System Architecture and Design:

The system is built using PHP and MySQL for the backend, with CSS and Bootstrap for the frontend.

1. Function:

- The System function mainly designed for the CRUD function, you are insert, edit, search and delete the document. (e.g. adminfine.php, adminuser.php, newincident.php, updateincident.php.)

2. Decoration

- The decoration of the whole website to present data using and involving HTML, CSS, JavaScript and Bootstrap to deal with the interface. (e.g.index.css.)

3. System:

- The user can log in and the information that the website will show depends on the username type. (e.g. login.php, logout.php)

How the system works:

1. Request: The user interacts with the frontend interface and requests for login,

editing, and creating data.

2. Processing: The request is handled by the system which uses the model to interact with the database.
3. Interaction: The model communicates with the database, performing necessary operations such as creating new entries, updating records, or deleting data.
4. Display: After processing the data, the controller sends the results to the view, which generates and displays the response in the user interface.
5. User Interaction: The user can then perform further actions, triggering the next cycle of request processing.

PHP Components:

The main php document:

1. index.php: This is designed for the login system. This document verifies whether you are a user or administrator. and giving you different information and functions.
2. login.php and logout.php: this design depends on whether you are a user or administrator and gives you different information and functions.
3. adminuser.php and admineditуser.php: managing the database information for who can use the website.
4. adminfine.php: managing and recording the important part of fines.
5. newincident.php: Managing the information for incidents and can delete and add the new information.
6. Coursework2.sql: a database that includes information and relationships between the data.
7. history.php: record changes to the document

Database Design:

The database consists of the following entities: username, people, vehicle, incident, ownership, fines, offence, and history. The relationship between each other is necessary for the data integrity of the system.

Entity-Relationship Diagram (ERD):

1. Username (of_id, username, password, username_type)
    - Stores user details, username_type for categorizing admin and user
2. People (People_ID, People_name, People_address, People_licence)
    - Stores people's details, including name, address, and licence.
3. Vehicle (Vehicle_ID, Vehicle_plate, Vehicle_type, Vehicle_colour)
    - Stores vehicle details, including vehicle_id, plate, brand and colour.
4. Incident (Incident_ID, Vehicle_ID, People_ID, Incident_date, Incident_Report, Offence_ID, Incident_name)
    - Stores incidents reported by officers. Each incident is linked to a username for distinguishing who can edit or delete the information.
5. Fines (Fine_ID, Fine_Amount, Fine_Points, Incident_ID)
    - Stores each fine to connect with a single incident.
6. Ownership (People_ID, Vehicle_ID)
    - Stores the information for connecting people and vehicles.
7. Offence (Offence_description, Offence_maxFine, ffence_maxPoints)
    - Stores offence for showing the information.
8. history (history_id, username, database, action, detail, type, timestamp)
    - For knowing the information who, and when do the changes to the system.

Relationships:

1. Fines and Incident

- Fine must be linked to an incident.
    - Fines → Incident: I (mandatory)
    - Incident → Fines: M (optional)

2. Incident and Offence

- Incident optionally be associated with an offence.
    - Incident → Offence: 1 (optional)
    - Offence → Incident: M (optional)

3. Incident and People

- Each incident optionally involve one people.
    - Incident → People: 1 (optional)
    - People → Incident: M (optional)

4. Incident and Vehicle

- incident optionally involve one vehicle.
    - Incident → Vehicle: 1 (optional)
    - Vehicle → Incident: M (optional)

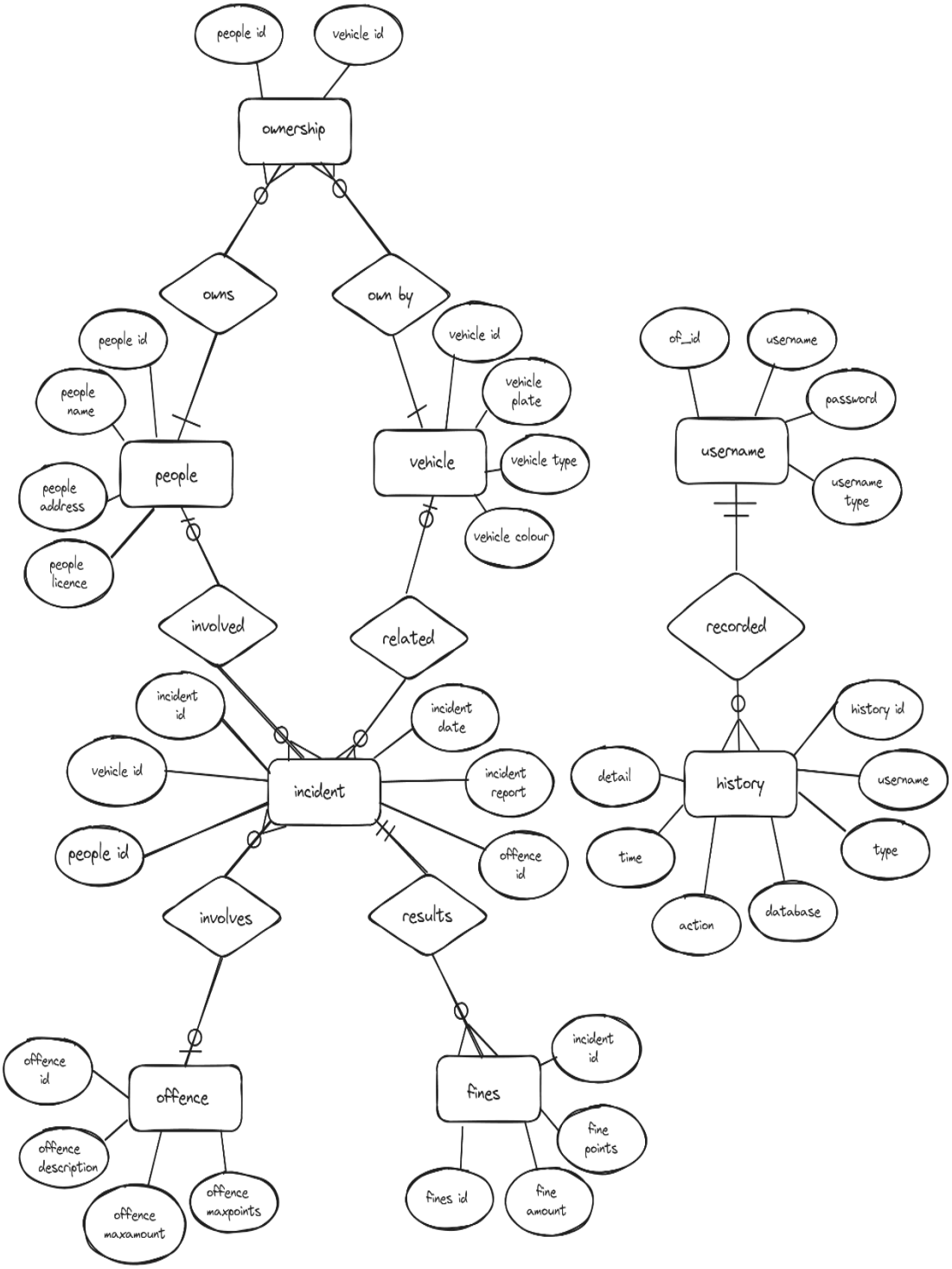
5. Ownership

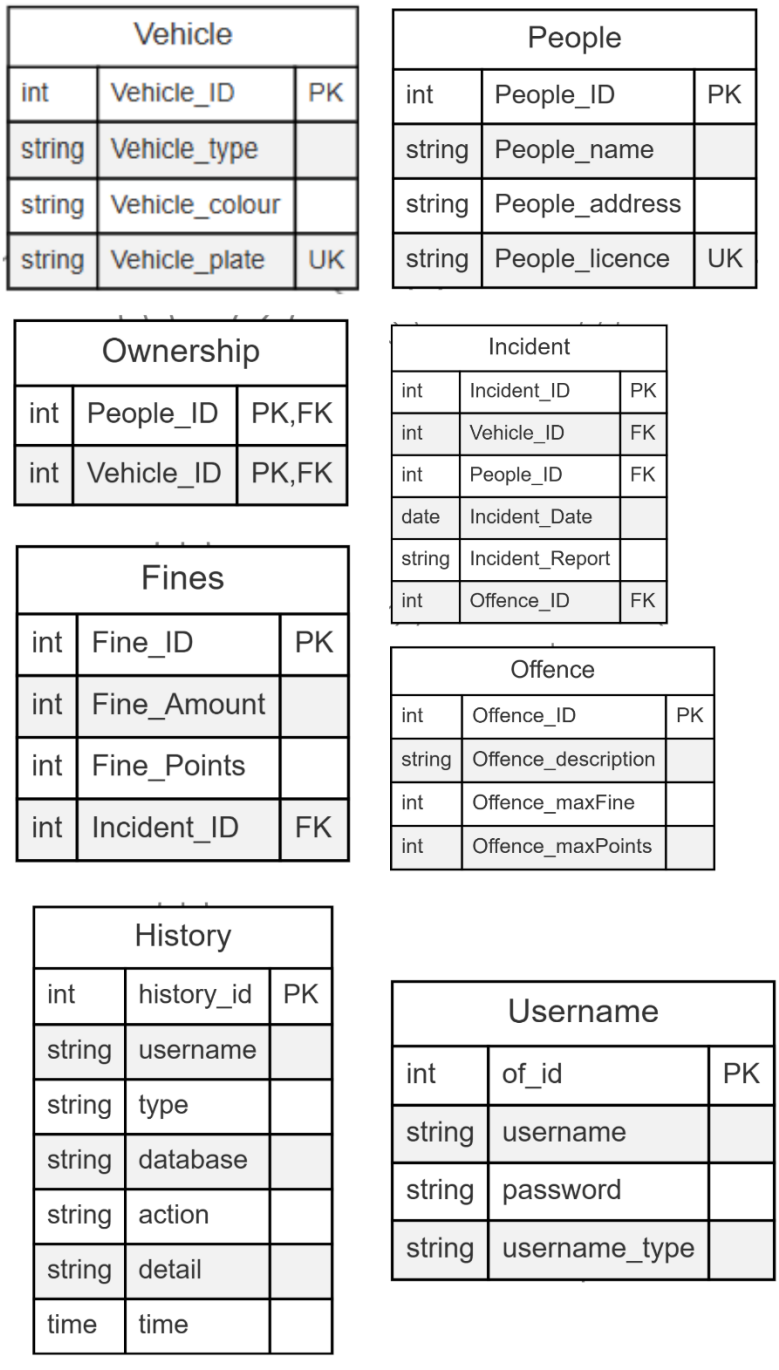
- People and vehicles have a many-to-many relationship.
    - People → Ownership: M (mandatory)
    - Vehicle → Ownership: M (mandatory)

6. Username and History

- Username optionally has multiple histories.
    - Username → History: M (optional)
    - History → Username: I (mandatory)

ER diagram:


people id
vehicle id
ownership
owns
own by
people id
people name
people address
people licence
people
vehicle id
vehicle plate
vehicle type
vehicle colour
vehicle
of_id
username
password
username type
username
involved
related
recorded
incident id
incident date
vehicle id
incident
incident report
people id
offence id
history id
detail
history
username
time
type
action
database
involves
results
offence id
offence description
offence maxamount
offence maxpoints
offence
incident id
fine points
fines id
fine amount
fines


Vehicle
int Vehicle_ID PK
string Vehicle_type
string Vehicle_colour
string Vehicle_plate UK
People
int People_ID PK
string People_name
string People_address
string People_licence UK
Ownership
int People_ID PK,FK
int Vehicle_ID PK,FK
Incident
int Incident_ID PK
int Vehicle_ID FK
int People_ID FK
date Incident_Date
string Incident_Report
int Offence_ID FK
Fines
int Fine_ID PK
int Fine_Amount
int Fine_Points
int Incident_ID FK
Offence
int Offence_ID PK
string Offence_description
int Offence_maxFine
int Offence_maxPoints
History
int history_id PK
string username
string type
string database
string action
string detail
time time
Username
int of_id PK
string username
string password
string username_type