

CMPUT291 Mini-Project Design Document

General Overview

This program is designed for users to search and register data from the database. It has the tables of registration of birth, marriages, vehicles and other convenient tables for the drivers which are the bill of sale, the tickets payment and the abstract of each driver. For the traffic officers, it provides functionalities to find the car owner and issue the tickets.

After starting the program, users are allowed to log in their accounts, for those accounts, the program distinguishes two characters 'a' and 'o'. The character 'a' means agents and 'o' means officers. After logging in, agents have rights to edit and browse all of the data, they can select the categories at the top of the interface, after clicking the category button, it leads to a new interface to enter the searching condition or the data which they want to add. The traffic officers have the right to access the information of issued tickets and searching for car owners.

The users also have an option to logout from their account and exit the program midway if they wish.

Detailed Design

1. Register a birth

After the user logs in to the system, for example, an agent wants to register a newborn, the interface will switch to the fill information part, which requires the full name, gender, birthdate, birthplace, mother and father's full names. When the agent is done with these pieces of information, they may submit and the entered data will update the original data file, then the system will automatically give a unique registration number and set the registration date as today for the newborn.

Relationship: If there is no information in the database for parents, the user is able to fill out their parents' first name and last name in order to update the person's information according to the items in the persons' table, for the birth date, birthplace, address and phone number are optional. Then the parents' data is stored in the person table.

2. Register a marriage

The second function is for the users to register a marriage. After the user(agent) has opened the 'register a marriage' category, the interface will display the blank of the first partner and the second partner's full name. Similar to the first function, the system will automatically set a registration date, place and a unique registration number.

Relationship: If the entered person cannot find in the database, the marriage table should connect the person table to gather the information of that person which are first name, last name. For the rest of the information, birth date, birthplace, address and phone number are optional.

3. **Renew a vehicle registration**

For the renew vehicle function, after clicked the renew category, users need to fill out the information contains the existing registration number, after entering the number, the system will automatically update the expiry date to 1 year from the current date. If the vehicle has expired or expiring today, the system will set the start date as the day after the current date.

Relationship: After the system gets the registration number, it will retrieve the data from the database which belongs to the registration schema. If it is not on the list, then the system will give another chance to enter a new registration number. If it's correct, the system will get the vin from registration then according to the vin number find the expiry date from the vehicle and renew it.

4. **Process a bill of sale**

Process bill means the car owners need to transfer their cars to a new person. So in this category's interface, it requires the vin of vehicle, the full name of the current owner and new owner, a new plate number. If the current name does not match the car owner, the transfer cannot be successful. Otherwise, the system will set the expiry date as the current date for the original registration, the new registration will move to the new owner and the registration date for the new one will set as the day after the current date, and the expiry date will be the date after one year starting from the registration date and the new registration will have a new unique registration number. The vin of old registration will move to the new registration.

Relationship: This function connects many data from different tables. At first, the system needs to match the entered owner name by retrieving data from the registrations table, then update a new registration with new names. Then retrieve the vin, expiry date from the vehicle table, set a new expiry date and copy the vin to the new registration.

5. **Process Payment**

In the process payment category, at the interface, users are prompt to enter the ticket number and the amount of fine that they want to pay. After they submit, the system will set the payment date as the current date, then update the payment amount for the ticket table by subtracting the total fine and the payment amount. The payment can be paid separately if the total payment larger than the total fine, the system will say "Invalid payment". Then the user needs to enter a new number for payment.

Relationship: For this function, after users submit the information, the system will retrieve the fine amount in the tickets table by matching the entered ticket number, then use the total fine minus input amount to update the new ticket fine.

6. **Get a driver Abstract**

This function is working for agents to get the driver's basic information. After entering the category, the interface requires agents to enter the first name and last name and a check button in order to see the data in order from the latest to the oldest. After the agent submits, there will be three parts display, the first part is the number of tickets, the number of demerits notices and the number of points within 2 years. The second part is the number of tickets, the number of demerits notices and the number of points within the lifetime. The third part is to show a list of tickets ordered by the violence date if the agents click the check button.

Relationship: There is a connection between the ticket table, the demeritNotices table, the vehicle table and the registration table. After entering the first name and the last name, the system will retrieve the number of demerits notices in the demeritNotices table, then use the foreign key first name and last name to find the details of each ticket, then use the vin of vehicles to gather the make and model of car in the vehicle table.

7. **Issue a ticket**

Traffic officers have permission to access this functionality. After the officers provide the registration number, the person's name, the make, model, year and the colour of the registered car will be displayed. Then the officers are prompt to provide violation text, the total fine and violation date to update the tickets for that person. Then the system will set a unique ticket number for each ticket, if the officer didn't provide the violation date, the system will also set the date as the current date.

Relationship: In this function, after the system gets the name, it needs to match the name in the registration table, then retrieves the information of that person's detail and car's detail. After that, the officer needs to provide the violation text and the total fine in the interface, then the system updates the ticket table to record the ticket for that person.

8. **Find a car owner**

The last function is to find the owner of a car. After entering the find a car owner category, the officers are going to provide one or more car information in the interface, then the system will display all matched results. If the system found more than 4 results, it will let officers select one, if the total results are less than 4, then each car will be shown as the latest registration date and expiry date. The name of the person listed in the latest registration record.

Relationship: In this function, it needs the connection between the vehicle, registration tables. The system needs to match the first name and last name in the

registration table first, then retrieve the data from the vehicle table, then display them. For more than 4 results, the system will choose the user-selected result to display, for less than 4 results situation, the system automatically selects the latest date of registration and expiry to display.

Test Strategy

For each functionality, we test with both the normal data and corner case. For example, set the data which perfectly satisfies the question conditions. Then use empty data to test, and some null values.

- For questions 1 and 2, we test in two scenarios, first is the general one which does not enter any names are not in the database. The second scenario is entering the person's name which is not in the data of person table.
- For question 3, we just use three different data of date to test the automatically set expiry date.
- For question 4, we set the scenarios as same as the questions 1 and 2 for the name part, then the same way as the question 3.
- For question 5, test in scenarios that exceed the amount of fine and not exceed.
- For question 6, set the number of total tickets greater than 5 which means need to separate into multiple pages. Also, set the date test which means insert the date of each ticket out of order, then use the check button to test display the results in order.
- For question 7, we set the scenario which is the entered name is not listed on the data of the registration table.
- For question 8, we set the count of completely identical cars for more than 4 and less than or equal to 4 to test the functionality entirely.

Break-down Strategy

The Breakdown of the project between the three members of this team is:

mharis: Wrote the main part for the GUI and wrote the code for 4 functions(1,6,7 & 8) of the total 8 and helped others with their functions. Time Spent:4 days

polavara: Compiled the code from the 3 members and wrote 3 functions(2,3 & 4) of the total 8 and helped on the GUI. Time Spent: 3 - 4 days

xn: Wrote the design document for the project and wrote 2 functions(5 & 6) of the total 8 functions. Time Spent: ~2.5 days

Method of Coordination: The team communicated on messenger and put their respective work on a shared google drive for every member to view.