Design Document

General Overview: This python application allows a user to do certain tasks depending on whether the user is a traffic officer or a registry officer. If the user is a registry officer they have 6 different tasks they can perform. And if the user is a traffic officer they can perform 2 different tasks.

User Guide:

If the user is a traffic officer then the 6 tasks they can perform is:

1. **Register a Birth:**

User will be prompted to enter first name, last name, gender, birth date, birthplace of newborn.
After this, user will be prompted to enter parents' first and last name. If any of the parents are not found in the database, user should input the information of parents separately. When entering parents' information, first name and last name of both of the parents are necessary.
User can leave the rest of parents' information blank. User can skip this by entering "Enter".
But if user choose to enter any of the parents' birth date, birth place, address and phone number, user has to follow the correct format.
Birth date should be entered as "yyyy-mm-dd".
For the birth place and address, users are only allowed to enter English character, blank space, comma and digits.
For the phone number, user should enter a phone number which is less than 10 digits.
If any of the format above is not correct, the system will catch the error and prompt the user to input again. So please follow to correct format of inputting.
After inputting all of this information, the newborn's information all be added in our system.

2. **Register a Marriage:**

User will be asked to enter information of marriage.
User has to enter the first name and last name of both of the partners.
If any of the partners are not in the database, user can add information to the database, which is birth date, address, address and phone number. This is optional, user can just skip this by entering "Enter"
And system will generate the date, registration number, and registration place and finish this registration.

3. **Renew a Vehicle Registration:**

The user should be able to provide an existing registration number. And the system will automatically set the new expiry date depending on the current expiry date. If the expiry date is expired or expires today then it will be set from one year from now. Otherwise it will be one year after the current expiry date.
User can enter the existing registration number to update the expiry date of vehicle.
If the registration number does not exist, user needs to enter again.
If the user entered the right registration number, system will update the expiry date based on the current expiry date of the vehicle.
If the expiry date has already reached or passed, the new expiry date will be one year after today's date. Otherwise, one year will be added to the existing one.

4. **Process a Bill of Sale:**

The user should be able to record a bill of sale by providing the vin of a car, the name of the current owner, the name of the new owner, and a plate number for the new registration. If the name of the current owner (that is provided) does not match the name of the most recent owner of the car in the system, the transfer cannot be made. When the transfer can be made, the expiry date of the current registration is set to today's date and a new registration under the new owner's name is recorded with the registration date and the expiry date set by the system to today's date and a year after today's date respectively. Also a unique registration number should be assigned by the system to the new registration. The vin will be copied from the current registration to the new one.
User input a vin of a car, the name of the most recent owner and the name of the new owner to finish the sale. Sale the car from current owner to the new owner

5. **Process a Payment:**

*Process a payment.*The user should be able to record a payment by entering a valid ticket number and an amount. The payment date is automatically set to the day of the payment (today's date). A ticket can be paid in multiple payments but the sum of those payments cannot exceed the fine amount of the ticket.One tickets cannot have more than 1 ticket on the same day.

6. **Get a Driver Abstract:**

User enter the first name and last name to get a driver's abstract, including:the number of tickets, the number of demerit notices, the total number of demerit points received both within the past two years and within the lifetime. Then the tickets information should be displayed. The user can choose to (1) view in normal order (2)view from old to new (3)view from new to old. And the user should choose to view all tickets and view 5 tickets if the tickets number more than 5. After choosing the result will be displayed.

If the user is a traffic officer they can perform 2 tasks:

1. **Issue a Ticket:**

User enters a valid registration number and can see the person's name that is listed in the registration and the make,model,year and color of the car registered. The user then gives a violation date but does not necessarily have to. If none provided the date will be set to the current date. The user then provides the violation text and the fine amount

2. **Find a Car Owner:**

The user provides one or more of make,model,year,color,plate and finds all matches. If there are more than 4 matches the make, model, year, color, and the plate of the matching cars will be shown and we will let the user select one. When there are less than 4 matches or when a car is selected from a list shown earlier, for each match, the make, model, year, color, and the plate of the matching car will be shown as well as the latest registration date, the expiry date, and the name of the person listed in the latest registration record.

Design of Software:

Question1:

We have one function: input_birth(), which is able to handle the mistake made by user while input birth date. Sometimes the input of date can be ridiculous. For example: 2000-13-40.
This function can match errors like this.
It will continue asking user to input a reasonable date.
Beside this one, we have another function outside, which is “change_ upper()”, switching every character to its upper case. We convert this python function in up() and insert it in SQL statement whenever we need it, because this system should be case insensive. We can convert every letter in upper case whenever we need to compare two strings in SQL statement.In the function: reg_birth(), we ask the user to input information of new born first, than it comes to information of parents. When the input is not in the correct format, the assertion will be activated and the program will prompt the user to input again. The user can choose to terminate the program as well. We have a while loop in the end of the former while loop. This while loop will prompt the user to input either “yes” or “no” to decide whether the user wanna go back to the menu or input information again.If the input is right, system will show user the message for correct input and break the “while loop”. After inputting the information of new born, system will ask for parents’ information. We have the same design of while loop and assertion for the inputting of parents’. Another feature is that the user can choose to skip unnecessary information by entering “Enter”. In this case, the length of string is zero and the assertion statement will not be activated.When the user finished entering everything needed, the SQL statement inside python will add all of the information inputted by user inside database. We used execute function to make this happen.

Question2

For reg_marriage(), the front part of the function is similar to reg_birth(). When the user is done with inputting partner’s information, system will automatically generate a registration place, registration number and registration date. The registration place is user’s city which comes along with the user’s information (in the database). The registration number is generated by adding 1 to the biggest one existing, and we can make sure that the registration number is unique. We use “datetime” module to generate today’s date in the local time.

Question3

We have “renew_vehicle_registration()” for question3. We use the same while loop to prompt the user to enter the registration number. If this does not exist, it will raise the assertion, the user either terminate the process or input again. Beside this, everything other than digits will be catched by assertion statement. After inputting the registration number correctly, we use max() function to compare the date of expiry and today’s date. Which one better will be added by one year. Then, the new expiry date will be added to the registration table by SQL statement”update”.

Question4:

In this function process_bill_sale() we control the input process in a while loop. Here’s how i check input:

1. **vin of a car** : the length should be no more than 5 or less than 1. No characters outside 0-9 A-Z a-z.
This vehicle should in the database

2. **most recent owner name**: the length for firstname and lastname should not be more than 12 or less than 1
Only letters allowed. The name should be in our database and should be the current owner of the vehicle

3. **New owner name**:the length for firstname and lastname should not be more than 12 or less than 1. Only letters allowed. The name should be in our database

4. **Plate**: The length of the plate should be 1 to 7. No characters outside 0-9 A-Z a-z

I use mostly assertion to check and if there’s something wrong let the user to input input again or back to the menu
After successful input the information, we first update the current owner’s expiry to today, then insert a new registration for new owner with regdate on that day and expiry one year after. Finally i print out the sale result

Question5

This process_payment() function use while loop + try except else + assertion to control input. Here’s how i check the input:

1.**tno**: A positive integer . In our tickets table. This ticket hasn’t been paid today. Current payment sum < balance

2. **Amount to pay:** A positive integer. This amount + Current payment sum <= balance.I use mostly assertion to check and if there’s something wrong let the user to input input again or back to the menu. Then insert tno, amount, today into the payments table.And display the payment result at the end.

Question6

This function get_driver_abstract() function use while loop + try except else + assertion to control input. Here’s how i check the input:

fname, lname:

1 no more than length 12

2 a-z A-Z

fname,lname in the registrations

I use mostly assertion to check and if there’s something wrong let the user to input input again or back to the menu
I use a while loop to choose

"(1) normal order"

"(2) from the latest to the oldest"

"(3) from the oldest to the latest"

And give user the option to back to the menu
When tickets num larger than 5, i use while loop to let the user to choose
View only top 5 tickets or all tickets.
Then display the result
If there is no ticket for the driver it will gives notice

Question 1-Traffic Officer:

ticketIssue() function

Use while loops to get correct user information along with assertions and exceptions. If the user enters incorrect registration info then display error and ask the user to re enter information or quit. Use sqlite commands to get the information to display to the user. Then ask the user for ticket information, date can be not given and will be set to current date but the violation text and fine will be required from user. Update the database with the new information and commit to database. Display the ticket information to the user.

Question 2-Traffic Officer:

find_carOwner() function.
Use while loops to ask user to enter the 5 fields. User can leave the 4 fields blank but must enter one. If the user enters input that is not all characters display error and raise assertions and exceptions. The model must be made up of characters and numbers otherwise raise assertions and exceptions. While loop to make sure the year is an integer and is valid. Use while loop to make sure the user enters only characters for the color and use assertions and exceptions otherwise. Use while loop for the plate and make sure the format is correct using Assertions and exceptions. The plate must be made up of characters and numbers. Search for the car using sql statement. If no matching cars print no matching cars otherwise find which user the car is registered to by using sql statements. If the car has no registrations print no registrations. Print out the information for the user.

Testing Strategy:

Question one:
I test it with upper and lower case, unreasonable date, phone number longer than 10 digits, name with more than 12 letters and name of parent whose name is not in database to make sure the program catch every mistake made by the user.
I will check the database every time i am done with birth registration.

Question two:
For the input part, it is pretty much the same. I will check the database every time i am done with registration marriage.

Question three:
For the input part, my testing strategy is pretty much the same to question one. I check the database before and after renewing the expiry to see if the date is incremented by one year. I also be aware of whether the current date has passed the expiry date.

Question 4:
Test the vin larger than 5 or smaller than 0 and enter character outside 0-9 A-Z a-z.
Check a vin that not in the database
Check a vin of a car that does not have current owner
For current owner and new owner name:
Check length more than 12 and characters outside A-Z a-z
Check the name that doesn't in our database

Question 5:
Test tno: input a string except for positive integer
Give a tno that doesn't exist in the database
Give a tno that the ticket has been paid today
Give an amount that is negative or over the balance or not an integer

Question6:
Name format test same as Q4, input name that doesn't have a registration
For view order choices: the number outside 1,2,3 and non number

Question 1- Traffic Officer:
Give a regno that is invalid or does not exist in database. Give incorrect format or information for the date, text, and fine amount.

Question 2-Traffic Officer:
Try makes,models,years,colors,plates that were invalid or were not in the database. Try to find cars that did not exist in database or did not have registrations. Try the printing of output with more than 4 matches and less than 4 matches.

Group Work Break Down:
Yanlin Chen: question 1, 2, 3
Qianxi Li: question 4, 5, 6
Kishan Patel: user login and interface, question 1 and 2 for traffic officer