

CMPUT 291 Mini-Project 01

– Design Document

System Overview and User Guide

This python program uses a PYQT to incorporate a GUI interface which can be navigated view mouse clicks and keyboard inputs. The user is required to input their login information and check (inside the check box) whether they are a registry agent or a traffic officer in order to use features exclusive to each either the officer or agent. The user can exit at any point via closing the window, but their information (if they input any) will not be stored/collected if not submitted via submit button located on each page of the interface.

Login Screen

The user must input their credentials before accessing the functionalities that are exclusively available to a person of their status (either being a registry agent or traffic officer)

Registry Agent

If you sign in as a registry agent, you are given the options to use the functionalities below. Each functionality is under its own tab which can be navigated via mouse clicks. Each functionality works essentially the same as you input information specified by each text box and replace it with the correct information and click submit.

1. Register a birth
2. Register a marriage
3. Renew a vehicle registration
4. Process a bill of sale
5. Process a payment
6. Get a driver abstract.

Traffic Officer

If you sign in as a traffic officer, you are given the options to use the functionalities below. Each functionality is under its own tab which can be navigated via mouse clicks. Each functionality works essentially the same as you input information specified by each text box and replace it with the correct information and click submit. s

1. Issue a ticket.
2. Find a car owner

Software Description

Login Screen

Text boxes for username and password input are available as well as a check box for registry agents traffic officers. The username input is then converted to UPPPERCASE while the password is left in whatever casings it is in, to which said inputs reference an SQLite database via query to see if the person is registered. If the person is not registered in the system the program outputs an error message, otherwise they are taken to their respective screens for either the registry agent or traffic officer.

Registry Agent Screen

This screen offers tabs to which each functionality is outlined in the tab title, you may select the tab to input said data to follow out titled functionalities. Below is the list of functionalities accompanied by a description of the methodology behind their execution.

1. Register a birth
The inputs of the textboxes are taken. A new registration number is generated to be set as this person's unique registration number. This information is then taken and inputted into the SQLite database (as long as it is all filled in) to create a new column for this person.
2. Register a marriage
After names of the partners as well as other information is to be submitted via the text boxes in the window, are inputted and submitted they are registered in the database with their own unique registration number.
3. Renew a vehicle registration
A registration number is to be inputted via text box in GUI, the column for said registration number is set to have the expiry date updated to one year from today if the expiry date has passed or is today, otherwise expiry is renewed/updated to one year

after the expiry date that is currently listed in the database (the expiry date associated with the registration number is found via querying for the expiry date in the database and then compared via if statements to today's date).

4. Process a bill of sale

All information inputted is taken in and set as variables, the input of the first/last name of the current owner is referenced against the name associated with the vin inputted (who's expiry has not occurred yet) in the SQLite database to ensure they match, otherwise the transfer is not made. If the names do match (and there are no errors in the other input information) then the owner's registration data is updated in the SQLite database to have the expiry date of today while a new registration is also made for the new owner with an expiry of one year from today.

5. Process a payment

Takes payment and ticket number as input. The function then checks to see if the payment amount is larger than the amount remaining, if so, the payment cannot be made. If the payment amount is equal to or less than the payment amount the remaining payment amount is updated in the database and the pdate is updated to today's date.

6. Get a driver abstract.

If the input name is found in the database, then a driver abstract is pulled via query from the database and printed out in the first box in the GUI. The first five tickets are printed in the textbox on the bottom of the window, if there are more than five then a scroll wheel appears which will allow you to scroll down to view more tickets.

Traffic Officer Screen

This screen offers tabs to which each functionality is outlined in the tab title, you may select the tab to input said data to follow out titled functionalities. Below is the list of functionalities accompanied by a description of the methodology behind their execution.

1. Issue a ticket

After inputting the registration number, it is taken to be referenced using a query to find the first/last name and various information under the registration table as well as referencing the vin number related to the regno under registrations to find various information about the vehicle registered to that person. The user may then input various data to input a ticket, after being validated by the function, it is put into the registry. Note, the violation date does not need to be inputted, but if it is not the current date is used as the violation date.

2. Find a car owner.

Based on the user inputs (either all filled in or partial), that information is used to query the database, all unique vins are found. Via for loops, the most recent registration dates are put into a new separate list. If there are four or less results, the information is listed in this window, if there are more than 4 results only partial information is listed, to which if clicked, all information is listed at the top of the current window.

Testing Strategy

Each functionality was tested independently to ensure proper operation. Each command was verified to give proper updates and to said database by way of inputting information and then checking the database for appropriate changes. Letter casing, inputs left empty, full of spaces, null values and various improper inputs were used to test and correct source code that in anyway deterred from what the functionalities were meant to do. Counter measures were taken to protect against SQL injection attacks. Passwords were tested to be printed as non-visible when inputted into the UI.

Group Work Break-Down

All work was split to what was agreed even and fair between group members. All work was completed to meet the standards of all other group members. Collaboration and aggregation of work was done via GitHub. Communication via text, in person meeting and Discord chat.

The Breakdown of who did exactly what work is listed below but note that overlap did occur between group members for error checking, aggregation of independent works into one finalized unit and bug fixing.

Members and Responsibilities

Member 1 (victor3): Login Screen, Traffic officer's functionality 1 and 2

Member 2 (tcarlson): Registry agents' functionality 1, 2 and 3

Member 3 (berhe): Registry agents' functionality 4, 5 and 6