Mini Project 1

Aakriti Bhatia
Joshua Patrick
Mohit Kumar

USER GUIDE

This database handles Enterprise data for Registry Agents and Traffic officers. All data and functionalities will be printed to the terminal window of whoever is using it.

Upon startup, the user is asked to enter a .db file to query the data with. Each user that is using this system is able to login using a valid user id and password. When a user logs in, they will be taken to a page where they can choose to perform some functions based on whether they have identified as an agent or officer. The functions are as follows:

1. **RegBirth** = register a birth: a user can register their birth by providing all asked details about their birth. Their birth will then be registered in the database.
2. **RegMar** = register a marriage: a user can register their marriage by providing all asked details about their marriage. Their marriage will then be registered in the database.
3. **RenewVR**= Provide an existing registration number and renew the registration. The user can set a new expiry date to one year from today's date if the current registration either has expired or expires today. Or a user can set the new expiry to one year after the current expiry date.
4. **procBill** = user can 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. This will only be valid if the name of the current owner matches the name of the most recent owner of the car in the system.
5. **procPay**= The user can 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).
6. **getDrAb** = The user can enter a first name and a last name and get a driver abstract, which includes 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.
7. **FindCarOwn**= A user can look for the owner of a car by providing one or more of make, model, year, color, and plate.
8. **issueTick**= The user can provide a registration number and see the person name that is listed in the registration and the make, model, year and color of the car registered. They are then able to proceed and ticket the registration by providing a violation date, a violation text and a fine amount. A unique ticket number will be assigned automatically and the ticket will be recorded.
9. A user can **logout** and be returned to the main screen.
10. A user can **quit** using ctrl-z

Mini Project 1

Aakriti Bhatia
Joshua Patrick
Mohit Kumar

DESIGN:

Language = python 2.7

MAIN()

The program is called from a main function titled main(). Within this function display_login() is called.

DISPLAY_LOGIN()

Login takes input of whether a user is an agent or officer. Each user of the system is able to login using a valid user id and password. If username and password are found, then the user is directed to their appropriate options using agent_actions(uid) if user is a registry agent and officer_actions(uid) if the user is a traffic officer. Password is hidden using getpass() library.

HOMESCREEN

The home screen is where almost all functionalities of the program are called from. Upon entering the home screen, a user will have many different options to choose from. The functions will be different based on whether or not they are an agent or an officer. These options can be found within the User Guide.

PROGRAM

We used three different python files to complete this project, Setup_login.py, Reg_Agent.py and Traffic_Officer.Py. The Setup_login.py represents the login page and it shows options for the agent and officers. The Reg_Agent.py file contains the class for the registry Agents and all the methods for the Class. The Traffic_Officer.Py file contains the class for the Traffic Officer and all the methods for this particular Class.

1. Login.Py

Login.Py acts as the main program. It displays the login screen to the user and if the user provides a valid login info then it displays the options for the user, depending of if they are an Agent or an Officer.

2. Traffic_Officer.Py

The Traffic Officer class contains 2 methods for each of the options that the Traffic officer can perform. These methods include:

issue_ticket(self,regno)

This method issues a ticket under the officer's name

find_car_owner (self,make, model, year, color, plate)

This method finds a car owner under the officer's name. Incomplete

Mini Project 1

Aakriti Bhatia
Joshua Patrick
Mohit Kumar

3. Reg_Agent.py

The Traffic Officer class contains 6 methods for each of the options that the Registry Agent can perform. These methods include:

register_birth(self,fname,lname,gender,bdate,bplace,mother_fname,mother_lname,father_fname,father_lname)

This method allows a user to register a birth. It sets the registration to today's date and the registration place to the city of the user. It also automatically assigns a unique registration number to the birth record.

register_marriage(self,p1_fname,p1_lname, p2_fname, p2_lname)

This method allows a user to register a marriage. It can also get information about the partner including first name, last name, birth date, birth place, address and phone if the partner is not in the database **process_payment(self, tno,amount)**

This method allows a user 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 ie today's date.

process_bill_sale(self,vin,curr_owner_fname,curr_owner_lname,new_owner_fname,new_owner_lname,plate_no)

This method allows a user 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. This is only possible if the name of the current owner matches the name of the most recent owner of the car in the system.

renew_reg(self,Num_Reg)

The method allows the user to provide an existing registration number and renew the registration at any time.

Driver_Abs(Self)

The method allows the user get a driver abstract, which includes 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. It also allows the user to see tickets ordered from the latest to the oldest.

TESTING STRATEGY:

Our testing strategy consisted of us using a python script we had created specifically to test each part of the assignment. We created methods designed to test each part that we had contributed. We tried entering as many possible errors and different inputs into our program as possible, to check for all errors we may have in our code. We also tested our project on all of our computers to see the ability of our program to run on different machines. We have attached our test file that we used on our Program. One problem that we faced was when testing our program on the lab machine was that it did not run due to certain restraints in python 2.7. After realizing this we changed our program from Python 3 and fixed it so that it would run in Python 2.7.

Mini Project 1

Aakriti Bhatia
Joshua Patrick
Mohit Kumar

GROUP WORK STRATEGY

As a group we worked very efficiently and effectively with each other. In our first meeting, we divided up the work equally (12 parts) between all 3 of us, using a random number generator:

- Mohit – 5. Process a payment, 4. Process a bill of sale, 1. Register a birth. & 11. Counter SQL injection attacks
- Aakriti – 6. Get a driver abstract, 3. Renew a vehicle registration, 2. Register a marriage. & 12. the hidden password component , and the Report.Pdf
- Joshua – 10. Login Screen, 9. Database Tables, 7. Issue a Ticket and 8. Find a Car Owner

We all helped each other out with each part, and approximately each took 9-12 hours to complete this project. We uploaded code to a GitHub repository frequently to keep each other updated on our work and had a group chat over Whatsapp to coordinate meetings and any other questions we had in relation with our project.