CMPUT 291 Mini-Project 1

Design Document

User Guide

This guide is designed to aid users that will be using our program. The program is designed for registry agents and traffic officers to perform a variety of actions that appear in their day-to-day jobs.

For a registry agent, this includes:

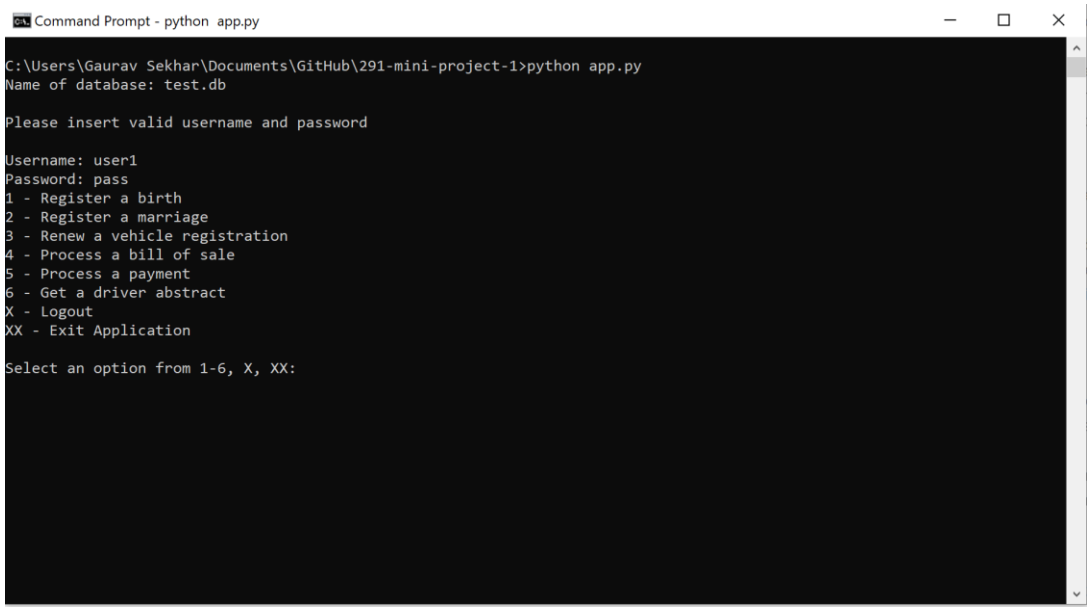
- Registering a birth of a child
- Registering a new marriage
- Renewing the registration of a vehicle
- Processing a bill of sale for a vehicle
- Processing a payment made towards a ticket
- Obtaining a given driver's abstract including tickets and demerit points

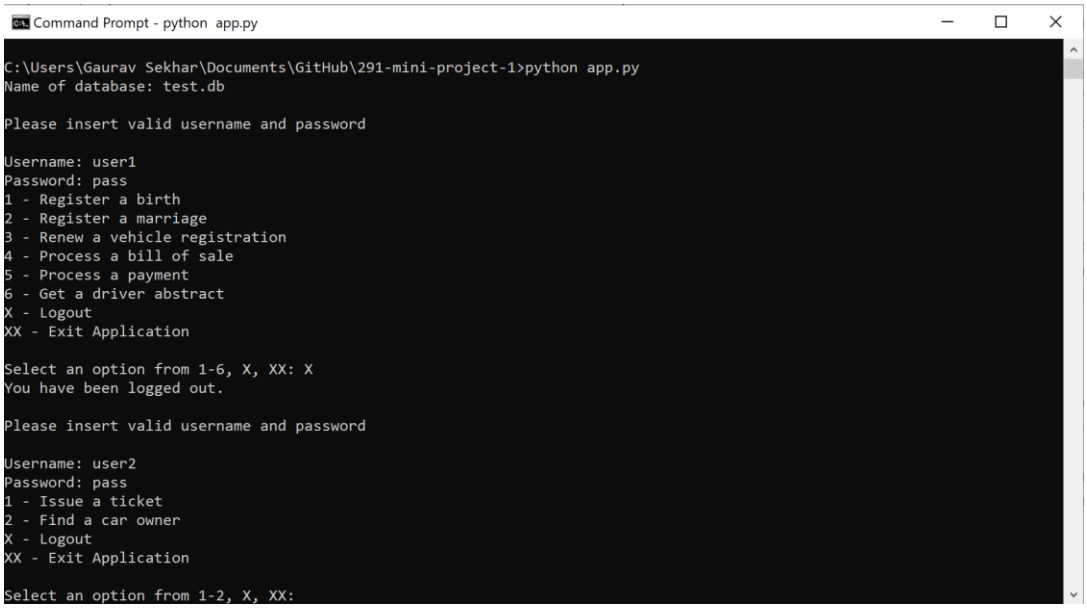
For a traffic officer, this includes:

- Issuing a ticket for an offence
- Finding a car owner based on preliminary information

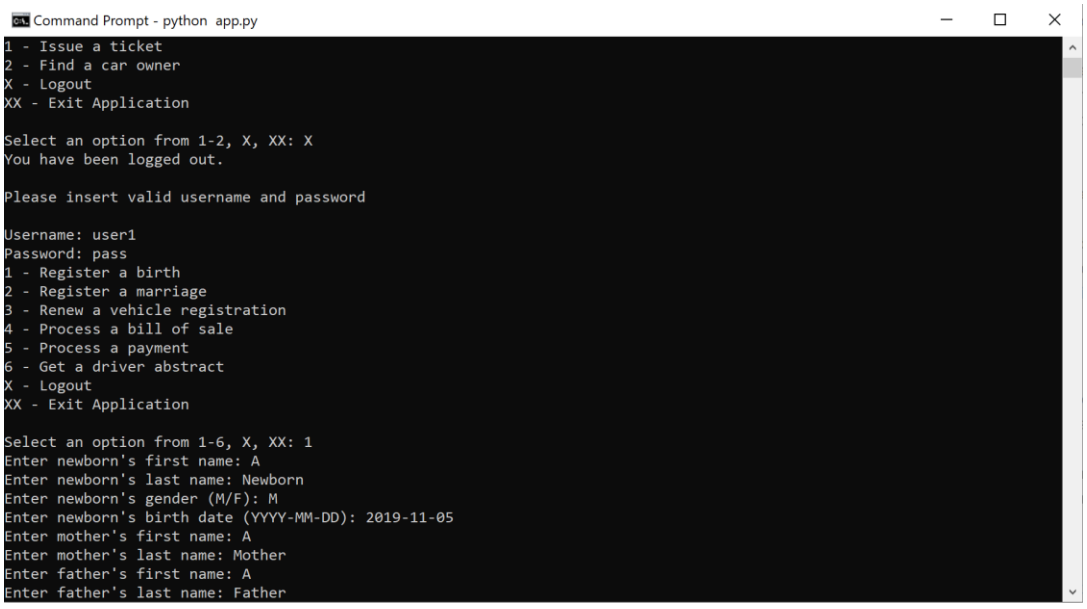
To launch, please type in the command `python app.py` .

Our program is designed such that a registry agent or traffic officer is able to login with their credentials. Based on the account that is used to login, a menu is displayed with the options above.





The user is then able to choose from the list of actions, including an option to logout or exit the application. Once the user has selected an option, further prompts are displayed regarding that specific action. For example, if a registry agent selects the '1 – Register a birth' option, they are prompted to enter information regarding the birth including information about the newborn and parents.

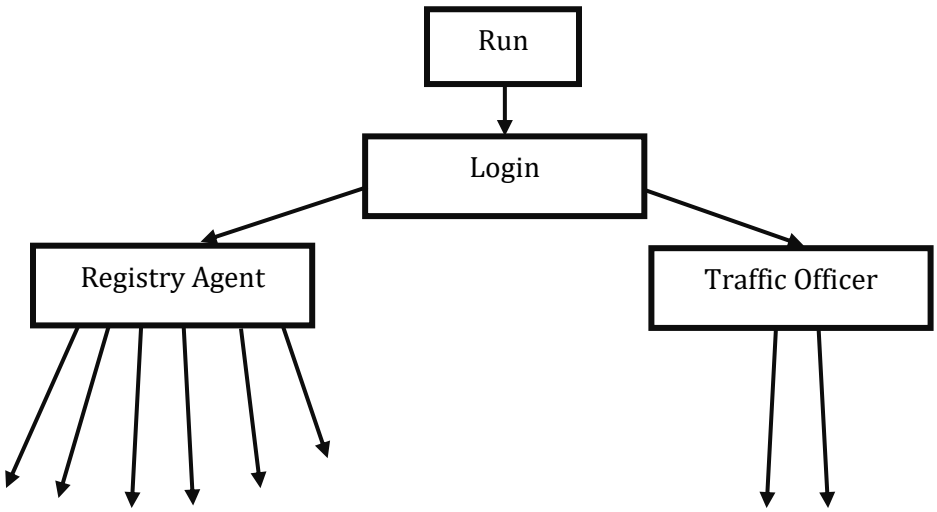


The user is able to also logout by selecting 'X' or exit the application by selecting 'XX'.

Program Design

The program was primarily designed within the file 'app.py'. This file contains an interface with functions for the application execution, login, and several actions which can be performed by the registry agent and traffic officer.

Each action calls a separate function which runs through a set of python commands to execute the desired action.



Program Testing

- In order to test the functions, we inserted values into a test database called 'test.db' with information regarding persons, vehicles, etc.
- The SQL statements provided for Assignment 2 proved to be helpful.
- NULL values were inserted for testing.
- Instead of inserting data into tables such as 'tickets', we used the "issue_ticket()" function which was implemented for a traffic officer account.

Group Work Break-Down Strategy

The project was essentially broken down into 9 sections, not inclusive of testing – login screen/CLI and the 8 functions (6 for a registry agent, 2 for a traffic officer).

Gaurav Sekhar – login screen/CLI, renew_registration(), process_sale(), find_owner(), code compiling, testing, debugging, design documentation.

Sirjan Chawla – register_birth(), register_marriage(), process_payment(), get_driver_abstract(), issue_ticket().

Time spent (Gaurav Sekhar) >12 hours

Time spent (Sirjan Chawla) > 12 hours

We used a GitHub repository for sharing code (github.com/gauravsekhar), along with emails for further communication.