CMPUT 291 Mini Project 1

Design Document

<u>CMPUT 291 Mini Project 1</u>

Design Document Contents:

1. General Overview/User Guide

Our system is written in Python3, and can be run as follows:

```
>py a3.py [database]
```

A database file can optionally be specified, otherwise the system defaults to a database called *‘a3.db’* residing in the same folder as a3.py

User Guide:

Upon starting the program, you will be asked to login by entering *‘ L ’* in the login menu. After specifying an email and the correct password, you will be taken to either the Agent or the Officer main menu. Which menu you are taken to depends on your login credentials.

In the Agent menu, you can select one of 6 operations:

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

In the Officer menu, you can select one of 2 operations:

1. Issue a ticket
2. Find a car owner

You can also logout from both menus by entering *‘ L ’*.

Select an operation by entering its corresponding number. For each operation, you will be shown a menu listing the relevant information.

2. Software Design

Our code is all hosted in the file a3.py, which contains the menus that allow users to interact with the system, as well as all the functions that interact with the SQLite database.

Generally, each operation is handled by a separate primary function, which may also be associated with small group of ancillary functions.

1. Login_screen() / agent_menu(uid,pwd) / officer_menu(uid,pwd) : Handle user login as well as maneuvering through different supported operations.
2. Register_birth() : Registers a birth with a name, gender, date, as well as the names of the parents.
3. Register_marriage() : Registers a marriage between two specified partners.
4. Renew_registration() : Renews a specified registration number.
5. Process_BOS() : Processes a bill of sale for a specified vehicle to a new owner.
6. Process_payment() : Records a payment for a specified ticket number.
7. Get_driver() : Returns the driver abstract of a specific person.
8. Issue_ticket() : Issues a ticket to a specific registration number.
9. Find_owner() : Finds the owner of all vehicles that match the specified search criteria.

3. Testing Strategy

We tested our system on data provided by Ryan Kang on eClass for use in Assignment 2. We then doctored the data a little to fit the specifications for this project. Each operation was tested based on the requirements outlined in the project spec. In addition, testing with invalid input was done to catch and fix errors.

4. Group Work Breakdown Strategy

We split the work up into parts. The login screens was one part, and then each operation was its own part for a total of 9. These parts were split so that Christian did the Login menus as well as the (2) Traffic Officer operations. Levi did Agent operations 1,2, and 3. Allan did Agent operations 4, 5, and 6.

Therefore, the work was an even 33.33% / 33.33% / 33.33% split between everyone. Overall an approximate 5-7hrs per team member were spent on the project, and each member was able to get their share of the work done.

We coordinated by leaving placeholder functions in the top user menu for each operation and then “dibsing” particular functions as we went along. We would then replace the skeleton functions as they were completed.

We used GitHub to keep all our work compiled together and up to date. We also maintained communication over text to ensure specific milestones were on track to being completed on time.