**Design Document**

**General overview of system** **with a small user guide**.

When the program is executed, you are first prompted to login.

The user is prompted for a username and a password respectively.

The username is then used to match the user with one of the users in the database.

The password is used to verify their identity by matching it to the user information in the database.

At this point, the user is either verified or rejected.

If the user is verified, they are given access to the interface.

The interface gives the user a list of tasks they can perform and the option to logoff.

If the user if an agent, they are given 6 options: *Register a birth, Register a marriage, Renew a vehicle registration, Process a bill of sale, Process a payment, Get a driver abstract.*

If the user is an officer, they are given 2 options: *Issue a ticket, Find a car owner.*

Once a task is chosen, the associated function is run.

The functions take in needed information from user and the database to return, insert, or update data in the database. All changes are committed to the system.

Once the function is complete, the user is brought back to the interface.

From here the user can continue to perform tasks until they choose to logoff.

When the user chooses to logoff, all data saved from the user is released and the program is ready to accept another user.

User guide: Input username, input password, choose task, follow instructions, logoff

(flowchart)

**Detailed design of software**

Login:

Get username

Get password and hide password at time of typing

Verify user

Login, if matching

Reject, if not matching

Interface:

If user is agent show options Register a birth, Register a marriage, Renew a vehicle registration, Process a bill of sale, Process a payment and Get a driver abstract.

If the user is an officer, show options Issue a ticket and Find a car owner.

Register a birth:

Get first and last name of newborn

Get gender of newborn

Get birth date of newborn

Get birthplace of newborn

Get first and last name of father

If father is not in person, register father

Get first and last name of mother

If mother not in person, register mother

Get today’s date

Chose unique registration number

Get users city as registration place

Get address from mother

Get phone from mother

Insert newborn in person

Register the birth

Register a marriage:

Get first and last name of partner 1

If partner 1 is not in database, register partner 1

Get first and last name of partner 2

If partner 2 is not in database, register partner 2

Chose unique registration number

Get users city as registration place

Register marriage

Renew a vehicle registration:

Get registration number

If expired or expiring today, update expiry to one year from today

If not yet expired, update expiry date to one year after expiring date

Process a bill of sale:

Get vin

Get current owner

If owner matches continue

If owner does not match, cancel sale

Get new owner

Get plate number

Set current registrations expiry date to today

Create new registration

Process a payment:

Get ticket number

Grab ticket fine

Grab sum of prior payments

Find balance

Get payment amount

If amount is less than or equal to balance process payment

If amount is greater than balance, cancel payment

Get a driver abstract:

Get first and last name of driver

Return abstract on driver

Issue a ticket:

Get registration number

Get date

Get reason

Choose unique ticket id

Insert new ticket

Find a car owner:

Input search parameter name

Input search parameter value

Return car details

**Testing strategy**. The testing strategy discusses your general strategy for testing, with the scenarios being tested, the coverage of your test cases and (if applicable) some statistics on the number of bugs found and the nature of those bugs.

Tested two ways:

The lines below was changed slightly and run in appropriate places to test various functions. It would allow us to see the change in the table before and after the insert, or update.

if debugIssueTicket == True:  
 print("TICKETS BEFORE:")  
 cursor.execute('SELECT \* FROM tickets')  
 debugQuery = 0  
 while debugQuery != None:  
 debugQuery = cursor.fetchone()  
 if debugQuery != None:  
 print(debugQuery)

Each function was also tested with scenarios specific to that function.

Test for login: fotray, 46MuaZMf

fotray, 46muazmf

Test for interface: fotray, 46MuaZMf, rb, Trayvon fox, mika grey, ‘’, ‘’, ‘’, ‘’, lo

Boulil, Q6v9n8xn, it, 300, 2019-11-03, lo

Test for registering marriage, case sensitivity, registering person, and inserting null values:

Trayvon fox, mika grey, ‘’, ‘’, ‘’, ‘’

Test for process a sale: 300, Diane Lee, mary fox, ‘’, ‘’, ‘’, ‘’, sdf2318

Tests for processing payment: 101, 100

100, 100

Test for issue a ticket: 300, 2019-11-03

1, 300, 2019-11-03

Test for find a car owner: ‘’, make, Doge, ‘’

y, make, Doge, Plate, ‘plate2’

**group work break-down strategy**.

Ishara Hettiarachchige:

Register a marriage, Process a bill of sale, Process a payment. Integrating all written functions into one

Time spent individually: 12 hours

Matthew Braun:

Login interface, Issue a ticket, Find a car owner. Design document

Time spent individually: 10 hours

Johnson Zhao:

Register a birth, Renew registration, driver abstract.

Time spent individually:

Time spent in meeting: 7 hours

Method of coordination:

Online🡪 git hub repository

3 In person meetups to check on progress, and submit files