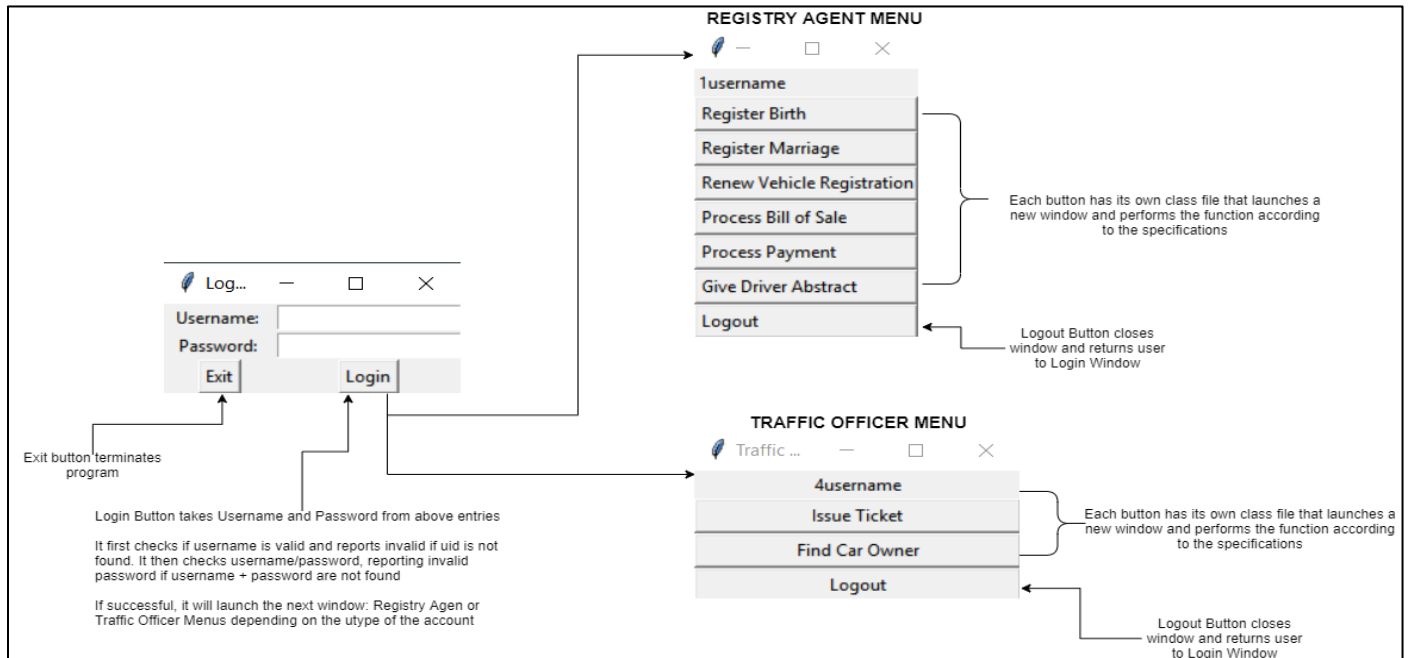


# DESIGN DOCUMENT

Created by: Jian Xian, Marshall Chang, Tina Do

## GENERAL OVERVIEW OF OUR SYSTEM:

Our program is written in python3 using the built-in modules tkinter for GUI and sqlite3. It is launched by running Launcher.py with the database path as the first parameter.



## DESIGN OF SOFTWARE:

Our software is divided into 3 types of classes:

### Launcher.py:

Called in command line to start the program. Launches Login Window and takes argument: database path. Defaults to 'test.db' if no database path provided.

**User Usage:** >>python3 Launcher.py <databasepath>

### Independent Window Classes (LoginApp.py, etc):

Front End windows inherit from tkinter superclass and initialize their own UI. Have methods that handle button or user input actions. Navigation between windows is done by destroying the current window and launching the next one while passing through necessary info.

Each of these classes can be tested and run on their own. This modular code design is more maintainable and allows us to work independently at different paces without merge conflict or relying on another's code to be done first.

**User Usage:** buttons are self-explanatory

### SQLController.py:

Backend class that uses sqlite3 module. Has no interface but handles all SQL queries under a single cursor, instantiated in every window. Easy to manage merges for this class because it's just a hub for all methods that require sqlite3 module.

## **OUR TESTING STRATEGY:**

To test the program, we will each test our own work for functionalities (Performs specifications properly) as well as quality assurance (ensure that the program does not break from simple user mistakes). Then we will also perform the same test process on another group member's work.

Functionality Tests will check all valid inputs for each specification, and ensure they produce a correct or reasonable output.

Quality Assurance Tests (QA) will check for possible user errors such as typos, blank entries, incorrect types, and ensure these produce a reasonable error output and do not crash the program.

## **GROUP WORK BREAK-DOWN AND DETAILS**

### **Group Meetings:**

#### **Meeting1 10/16 (1h)**

- Discuss possible strategies for program design

#### **Meeting2: 10/25 (2h)**

- Confirm program design
- Divide remaining work
- Specify planned progress/deadlines

#### **Meeting3: 10/30 (1h)**

- Review testing strategy
- Divide testing work

### **Software Development- Includes Design, Development, and self-QA time of functions**

#### **Marshal:**

- Overall Design and Structure (2h)
- Login, Launcher (0.5h)
- SQLController Outline (0.5h)
- Register a Birth (1h)
- Process a payment (0.5h)
- Misc Classes: Error Window, Registry Agent Navigation, Create Births (0.5h)

#### **Jianxiang:**

- Register a Marriage (2h)
- Renew a Vehicle Registration (0.5h)
- Get a Driver Abstract (2h)

#### **Tina:**

- Create Test Database (1.5h)
- Process a Bill of Sale (1.5h)
- Issue a Ticket (2-3h)
- Find a Car Owner (3h)

## **QUALITY ASSURANCE TESTING:**

### **Tina updating code per Marshall's Quality Assurance Testing: ~1h**

- corrected unique constraint failures and handling of crashes, ensured user input is in correct date format, adjusted case sensitivity of find car owners, made sure log out button re-directed user to the traffic officer window

### **Tina testing Marshall: ~1h**

- Tested "Register Birth" function: ~30min
- Tested "Process Payment" function: ~30 min

- checked for correct date formats, blank entry inputs, updating new data to database correctly, function performs as per the specification and clarifications, updating new data to database correctly, window displays correct title

**Tina testing Jianxiang: ~1.5h**

- Tested "Register Marriage" function: ~30min
- Tested "Renew Vehicle Registration" function: ~30 min
- Tested "Driver Abstract" function~30min
- checked for valid date inputs, review of SQL queries to perform as intended, ensured SQL queries returned correct result and displayed accordingly, found program crashes

**Marshal:**

- Test and Fix Errors in 'Login' (0.5h)
- Test and Fix Errors in 'Register Birth' (0.5h)
- Test and Fix Errors in 'Process Payment' (0.25h)
- Test and Report Errors in 'Register Marriage' (0.25h)
- Test and Report Errors in 'Renew Vehicle Registration' (0.5h)
- Test and Report Errors in 'Process Bill of Sale' (0.25h)
- Test and Report Errors in 'Give Driver Abstract' (0.5h)
- Test and Report Errors in 'Issue Ticket' (0.5h)
- Test and Report errors in 'Find Car Owner' (0.5h)

**Jianxiang:**

- Test regbirth (30min)
- Test and fix regmarriage (1.5h)
- Test and fix renew vehicle (1h)
- Test process sale (30min)
- Test process payment (30min)
- Test and fix driver abstract (2h)
- Test issue ticket (30min)
- Test find car owner (30min)