CMPUT291 Project 1 Design Report

By

Zhaorui (Teppie) CHEN

Zhengyang (Nicholas) LI

Jiaxuan YUE

Abstraction

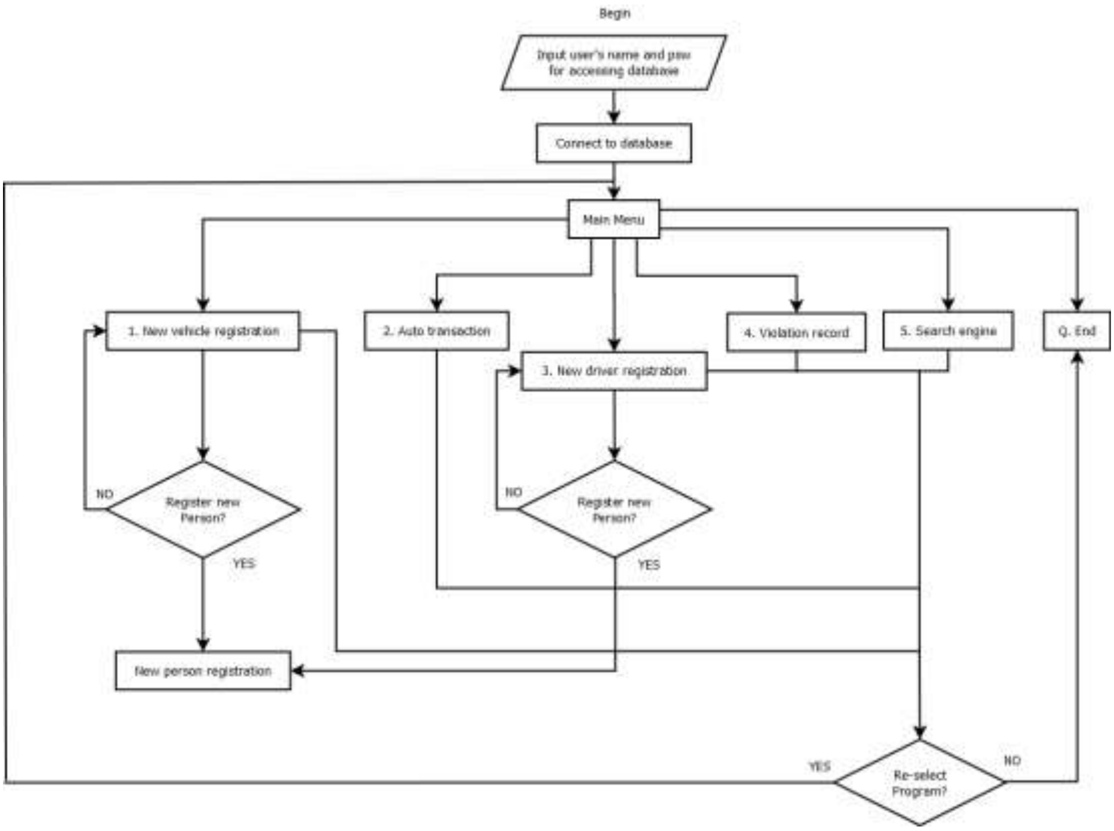
This report introduces the design of an auto registration system based on SQL database and Python 3. The system is able to register for new vehicle, auto transaction, licence, and violation record, as well as draw required information for customers.

Introduction

Auto registration system is database widely used by registry and government to manage information related to drivers and vehicles. In this auto registration system five functions are achieved, including:

- New vehicle registration:
  By this component an auto registration officer can put detailed information about a new vehicle and personal information of its new owner into database.
- Auto transaction:
  By this component an auto registration officer can enter and update all necessary information to complete an auto transaction, including, but not limiting to, the details about the seller, the buyer, the date, and the price.
- Driver licence registration:
  By this component the information needed to issuing a drive licence are recorded.
- Violation record:
  By this component a police officer can issue a traffic ticket and record the violation.
- Search engine:
  By entering a licence number or a given name, all personal information could be returned. By entering a licence number or a SIN, all violation records of the person could be returned. By entering a vehicle’ serial number, all vehicle history could be returned.

The flowchart is shown below.



Method

The program starts from class *application*, and from the constructor of *application*, class *Connection* is initialized.
Class *Connection* has ten methods.

- __init__(self): This method build up a connection to database using user's name and password.
- doCommit(self)/disconnect(self): These methods apply encapsulation for class *Connection*.
- executeStmt(self,stmt): This method execute the given statement but does not return anything.
- fetchResult(self,query): This method execute the given query to the database and return the result into a list.
- createInsertion(self,table,*args): This method returns the SQL INSERT statement in string type automatically.
- createDeletion(self,table,keyAttr,value): This method returns the SQL DELETE statement in string type automatically.
- createQuery(self,columns,tables,conditions): This method returns the SQL SELECT statement in string type automatically.
- ifExist(self,table,keyAttr,value): This method checksif the given key is already existed in the table.
- createCursor(self): This method creates a new cursor.

After the *application* class is called, main() method is executed. In this method, a menu will be provided to user by caling selectMenu() method. After user input his/her choice from the five functions, main() method will call the corresponding method to do further steps.

- If user chooses 1 from main menu, newVehicleRegistration() will be called. This system will ask user to input the serial number of the new vehicle. The method checkFormat() checks if the input format is correct or not, and ifSerialNumExist checks whether the input has been existed in database (A *new vehicle* means the serial number has not been existed in database). Re-enter is required if anything does not meet the requirement. Then the system will ask user to input SIN of the primary owner. Again format and existence are checked by methods checkFormat() and ifSinExist(). If SIN has not been existed in database, the user will have a choice between whether registering this SIN as a new person or re-input the SIN number. Choosing registering this SIN will leads the user to new person registration system(newPeopleRegistration() method), which will gather necessary information of the person and put them into database by createInsertion() method and executeStmt() method. If the SIN input by user is acceptable, the system will gather necessary information of the vehicle and put them into database using the same way as newPeopleRegistration() do. This system will also ask user if he/she would like to add a secondary owner.
- If user chooses 2 from main menu, autoTransaction() wil be called. The system will ask user to input SIN of the seller and serial number of the vehicle. If both input are valid, system will check if the seller is primary owner of the vehicle. Only primary owner can sell the car. Then the system will gather the SIN of buyer and price of the car. If a buyer’s SIN does not exist, user will have a choice between registering it as a new person or re-enter SIN. The system will also generate a transaction id based on the original database, and pull the current date from operating system for transaction time. Finally it will update the owner table and make the buyer primary owner for this vehicle.
- If user chooses 3 from main menu, newDriverRegistration() will be called. First the system will ask for SIN of the new driver and check if it is valid. If the SIN is valid the system will check if the person is holding a driver’s licence or not. Then the system will gather information about the new driver, including a new licence number, class of licence, expiring date, and file name of photo. System time will be pulled as issuing date. After all these basic information are put into database, the system will ask if the user would like to add a driving condition. The user may enter an existing condition id, or create a new condition for this driver.
- If user chooses 4 from main menu, violationRecord() will be called. This system will generate a new ticket number according to the original database first, and ask the officer to input his/her SIN. Then the system will ask if the officer identify the violator and his/her SIN. If the violator is identified, his/her SIN will be used as violator ID, else the system will use SIN of the vehicle’s primary owner as violator ID. Then the system will gather information including violation type, place, and description. Violation date will be automatically generated from

system time.

- If user chooses 5 from main menu, searchEngine() will be called. The system will let user choose what they want to search for. The user also have to choose by what they want to search. After the user input his/her choice, the system will gather necessary information and use SELECT statement to pull information from database.
- If user chooses ‘Q’, end() will be called and program ends.
- If the user choose to register a new person, newPeopleRegistration() will be called. This system will ask the user to input name, height, weight, eye color, hair color, address, gender, and date of birth of the person.

For high efficiency and ease of use, we have two assumptions. One is the system will automatically generate the licence issuing date and ticket date from the system time, and the other one is the new ticket number and new transaction id will be automatically generated according to the initial database.

Conclusion

This reprot briefly introduced the five functions of this auto registration system. Following the tips of input requirements, the user will be able to do what they want. This system is easy to use and with high efficiency.