CMPUT 291 Mini-Project #1

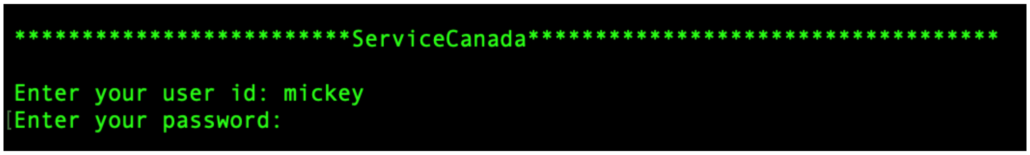
By: Paul Cleofas 1545932, Solomon Song 1557941

General Overview

For Mini Project #1, our main goal was to build an application that keeps government data in a database and provides services to government workers in order for them to access and input specific information.

User Guide

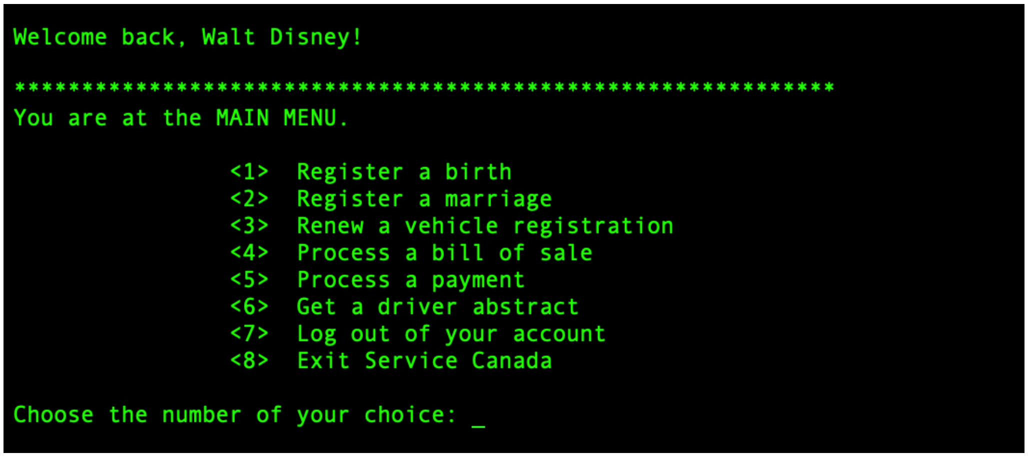
Our application allows a user with correct authentication to access database functions within the scope of their prospective role. In order to start accessing any data, our program prompts the user to login before it grants access to the information.



```
*************************ServiceCanada***********************************

Enter your user id: mickey
[Enter your password:
```

Once the user has logged into the application, there will be a "Welcome <USER>" title, selection menu for desired functionalities, and space for a user input. The menu has a list of numbers that correspond to a specific function. The user will input any digit and press enter in order to access said function.



```
Welcome back, Walt Disney!

*************************************************************
You are at the MAIN MENU.

                <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
                <7>  Log out of your account
                <8>  Exit Service Canada

Choose the number of your choice: _
```

The picture above shows a role dependent menu and the functions that are available to that specific role of the user. The number of functions depend on the user excluding the last two options, <7> and <8>, which can be seen throughout the program respectively.

Once the user chooses an item in the menu, the program will run through each function. The user will need to follow the prompts given. The database will automatically update once it finished an application function without any errors or invalid inputs.

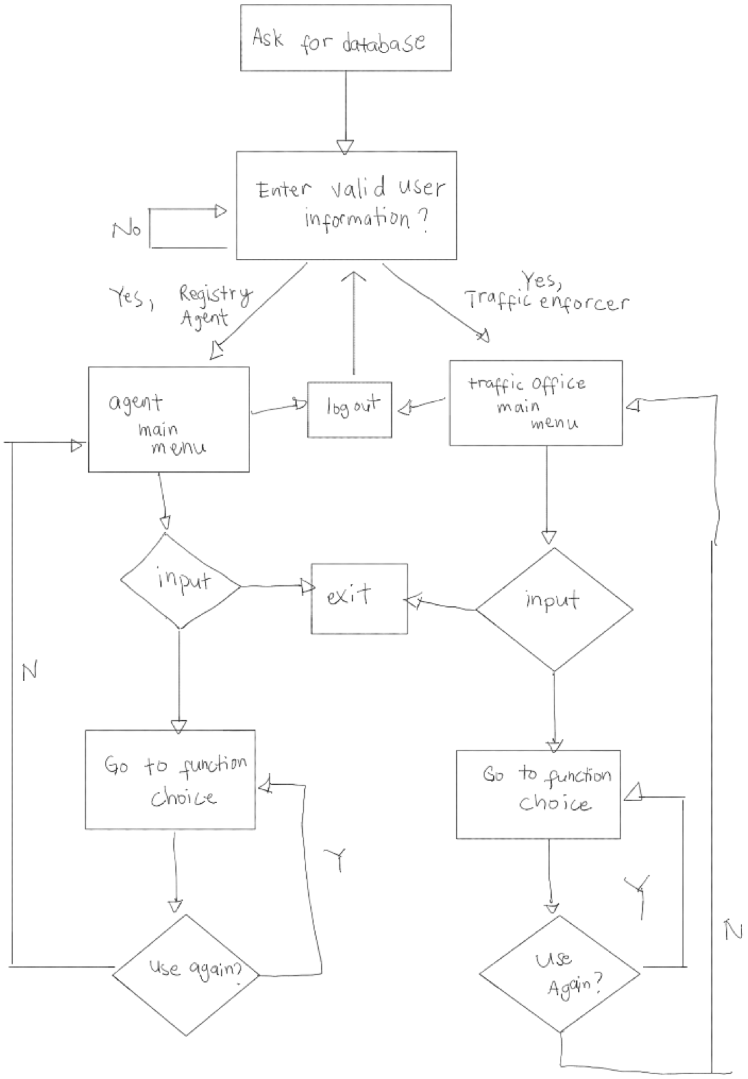
Some functions, such as issuing a ticket and finding a car owner, can only be accessed by an account belonging to an officer.

Design of Our Software

Design Overview

Our program is written in one python source file where all the functions can be found. The choice of doing a single file is due to heavy dependency of functionalities to multiple utilities function that acts as effective error checkers for both inputs and operation. We wanted to prevent circular dependencies in modular approach, for example, when main menu redirects to the functions and functions redirect back to the main menu.

Flow Diagram



Functionalities

For each type of user, we created a specific module for each allowing user access to specific functions.

- *Traffic Officer: access to issuing tickets and finding a car owner*
- *Registry Agent: access to registering a birth, registering a marriage, renew a vehicle registration, process a bill, process a bill payment, and getting a driver abstract.*

Once a module is accessed, the program will display the users name and the specified menu.

i.e.

```
Welcome Back, <User’s Name>!

*************************************

You are at the MAIN MENU.

<1> Issue a ticket

<2> Find a car owner

<3> Log out of your account

<4> Exit Service Canada

Choose the number of your choice:
```

The program will initiate and run a function once the user chooses and enters a corresponding number.

Testing & Group Work

Testing

Our testing entailed of running the other partners code and using specific inputs to find problems or holes in the code. We ran possible scenarios that a user could input either by mistake or intentionally. We also tried "breaking" the code through numerous of invalid inputs and arbitrary values. This gave us a better understanding of the nature of our program which ultimately allowed us to implement error checking that is quite thorough.

Group Work Breakdown

Our approach to splitting up the group work was quite simple in that we only had two members in our team. This meant that we could essentially split the work straight in half.

Function Breakdown:

1. *registerBirth() – created by sysong1*
2. *registerMarriage() – created by cleofas*
3. *renewReg() – created by cleofas*
4. *recordBill() – created by sysong1*
5. *processPayment() – created by cleofas*
6. *getDriverView() – created by sysong1*
7. *issueTicket() – created by sysong1*
8. *find_owner() – created by cleofas*

As seen in the list above, we assigned each function to a person to work on and then we brought all of them together into one main python file.

For communication, we mainly used Facebook messenger to communicate meet up times.

For version control, we pushed most of our code to Github so we were able to access each other's code when necessary.

And finally, we used Google Docs to record work assignment.