**Database Code Document**

This instruction document is for all the database related code, where we have codes for inserting data into the database, search the data from database based on ID or Name and, systems for update the details on the database. In this project, we are using PostgreSQL Database.

PostgreSQL is a powerful, open-source object-relational database system with over 30 years of active development that has earned a strong reputation for reliability, feature robustness, and performance. PostgreSQL is available in all Ubuntu versions by default. However, Ubuntu "snapshots" a specific version of PostgreSQL that is supported throughout the lifetime of that Ubuntu version. Other versions of PostgreSQL are available through the PostgreSQL apt repository. You can also download and install it on Windows 10.

**Note:** This project is developed on a Linux System (Ubuntu), so it is advisable to use Linux System.

**System Set-Up**

There are a couple of Python packages that need to validate before running any above program in this folder. Please go to the [Installation Documents](https://github.com/ripanmukherjee/Robotic-Greeter/tree/master/Installation_Documents) folder and follow all the steps.

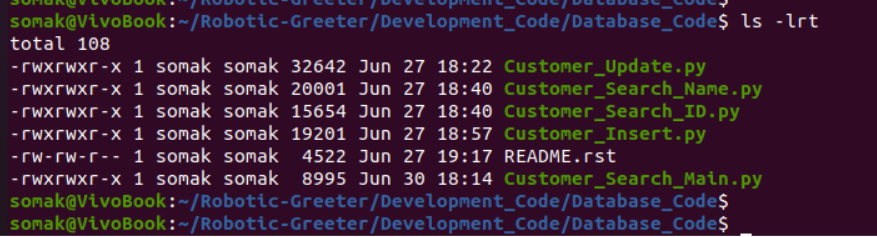
**Pre-Requisites**

* A system running on Windows/Ubuntu APP/Ubuntu OS
* A user account with sudo/administration privileges
* Access to a terminal window/command-line

Before continuing with this tutorial, make sure you are logged in as root or a user with sudo or administration privileges and completed all the necessary steps from the [Installation Documents](https://github.com/ripanmukherjee/Robotic-Greeter/tree/master/Installation_Documents)folder**.**

In this tutorial, we will explain all the Database Codes that are written in Python3 on Ubuntu. We have some codes related to the database as follows:

1. Customer\_Insert.py
2. Customer\_Search\_Main.py
3. Customer\_Search\_ID.py
4. Customer\_Search\_Name.py
5. Customer\_Update.py



All the Python codes in the Database Code folder deals with main customer details tables, which contains all the information of the users. The main tables are in the following:

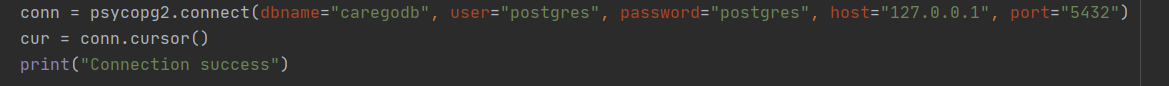
* Development (DEV) : carego\_customer\_dev
* Test (TEST) : carego\_customer\_test
* Production (PROD) : carego\_customer\_prod

Only, Customer\_Insert.py program uses the following sequence table. The sequence table is essential, which helps to create an auto-increment ID for the main table.

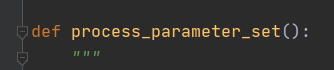
* carego\_customer\_dev\_ID\_seq

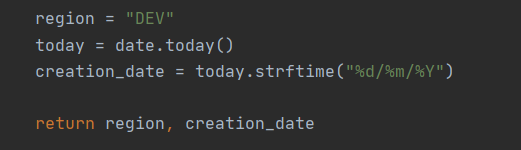
Most of the program uses database connection from Python as follow :

conn = psycopg2.connect(dbname="caregodb", user="postgres", password="postgres", host="127.0.0.1", port="5432")



Currently, we only used the **carego\_customer\_dev** table since it is only in the development phase. Each of the codes in this folder has one function called **process\_parameter\_set(),**which contains the region value. Depending on the region value, the program will select the respective tables. Hence it is essential to validate this function before running any codes. Also, it is crucial to verify the table before running any programs. To check the table, please go through the[PostgreSQL\_pgAdmin4\_Installation](https://github.com/ripanmukherjee/Robotic-Greeter/tree/master/Installation_Documents/PostgreSQL_pgAdmin4_Installation)document, which is present inside of [Installation Documents](https://github.com/ripanmukherjee/Robotic-Greeter/tree/master/Installation_Documents) folder.





**Database Code**

Database related code is written in Python as follow:

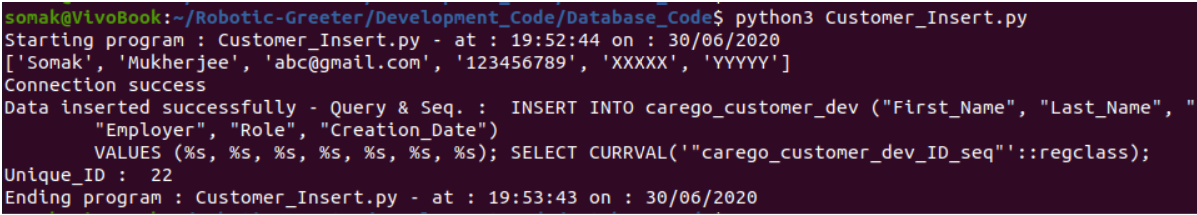
**1. Customer\_Insert.py**

Customer\_Insert.py is use for inserting the data of the user into the table mentioned above. This program will be called from Main\_Process.py. If the user wants to save their details in the database, then Main\_Process.py will call this program, and this process will insert the data into the table mentioned above.

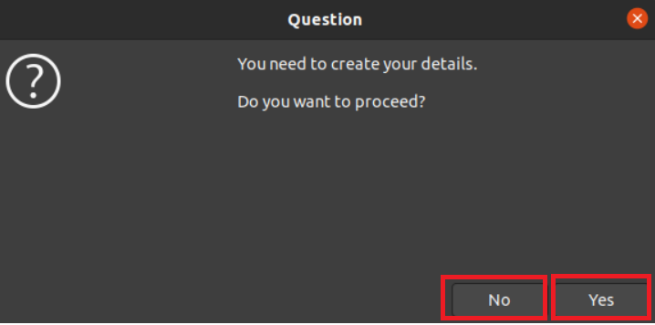
**Execution:**

* To run this program, run the below command from your command prompt.

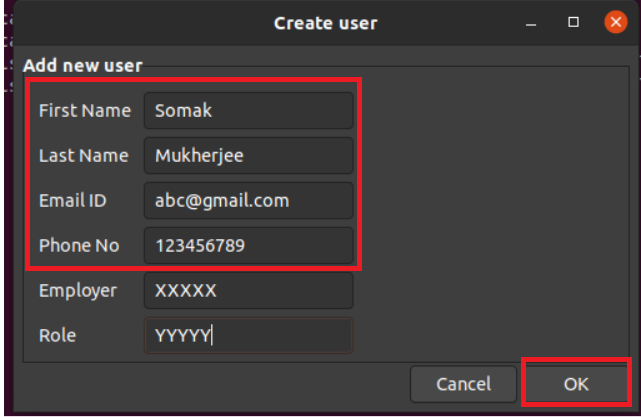
$ python3 Customer\_Insert.py



* Once you run the program, it will prompt a pop-up message as below. Click Yes to continue and No to exit.



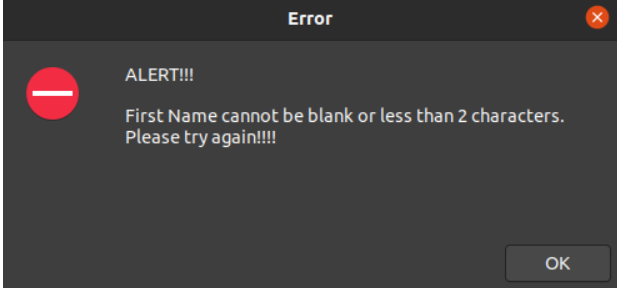
* After you click the Yes button, this program will pop-up a new create user message. In this pop-up screen, you need to fill your details as follows. Then click OK to continue.



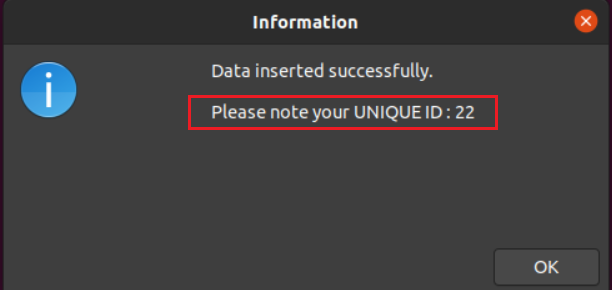
There are some validation present in the fields as below:

|  |  |
| --- | --- |
| **Fields** | **Description (Validation)** |
| First Name | It cannot be less than two characters or numbers. |
| Last Name | It cannot be less than two characters or numbers. |
| Email ID | It cannot be less than seven characters and has to include @ and dot. |
| Phone No | It cannot be less than seven numbers and must be numeric. |
| Employer | NA |
| Role | NA |

* If you do not fill the correct details as per validation, then this process will prompt an Alert pop-up with the description. Then, you can click OK to insert your details again.

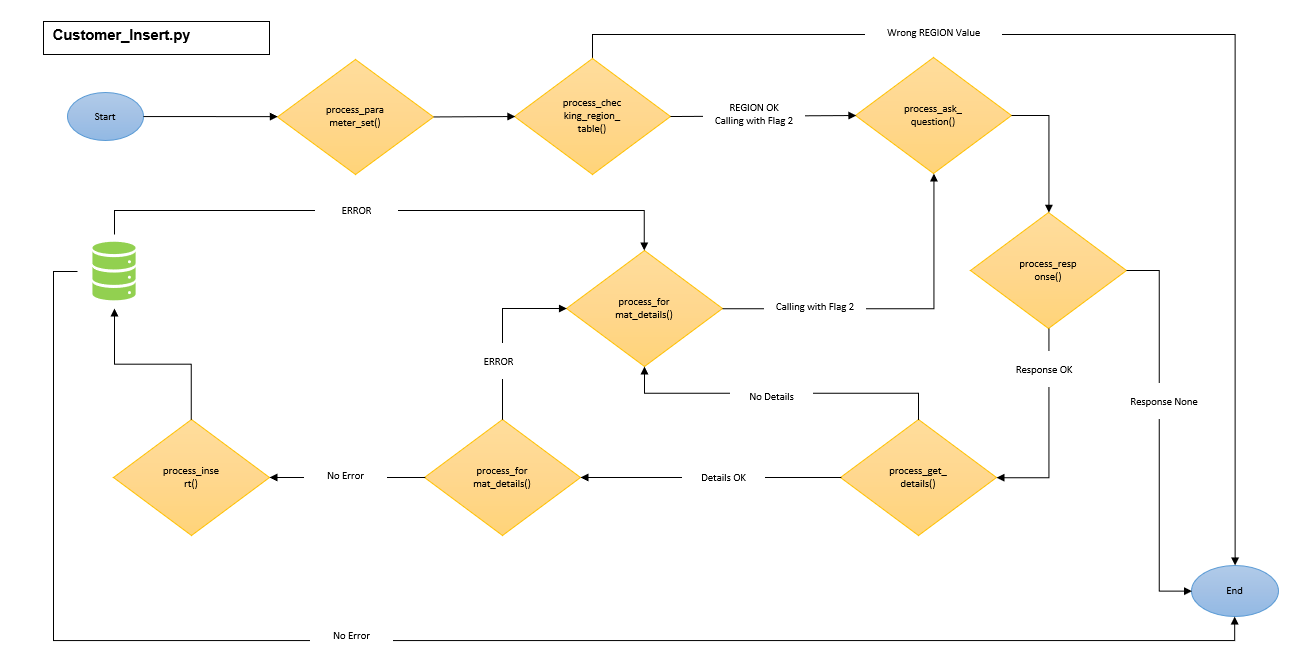


* If you fill-up the details correctly, upon clicking on the OK button, this process will insert your details into the table and later pop-up your UNIQUE ID.



**Flowchart**

Following is the flowchart architecture of the Customer\_Insert.py program.



**2. Customer\_Search\_Main.py**

Customer\_Search\_Main.py is used to search the user's data, and it will be called from Main\_Process.py. If the user wants to explore the data, this process will ask if the user wants to search by ID or Name, and as per the selection, this program will call the below two programs:

**I) Customer\_Search\_ID.py**

Customer\_Search\_ID.py is used to search the user's data from the table mentioned above by using an ID. This program will be called from Customer\_Search\_Main.py based on search criteria. If the user selects the ID option in Customer\_Search\_Main.py, then it will call Customer\_Search\_ID.py. And this program will ask the user to give the ID as input to search the data. Also, you can run this program as a stand-alone program.

**II) Customer\_Search\_Name.py**

Customer\_Search\_Name.py is used to search the data of the user from the table mentioned above. This program will be called from Customer\_Search\_Main.py based on search criteria. If the user selects the Name option in Customer\_Search\_Main.py, then it will call Customer\_Search\_Name.py. And this program will ask the user to give the name as input to search the data. Also, you can run this program as a stand-alone program.

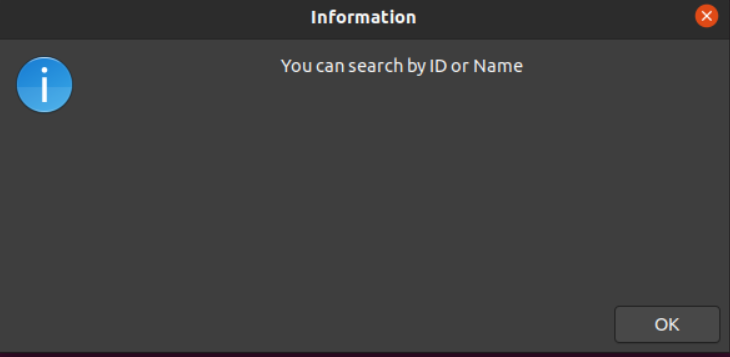
**Execution:**

* To run this program, run the below command from your command prompt.

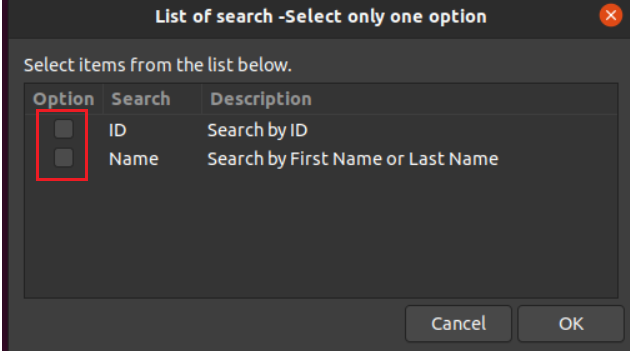
$ python3 Customer\_Search\_Main.py



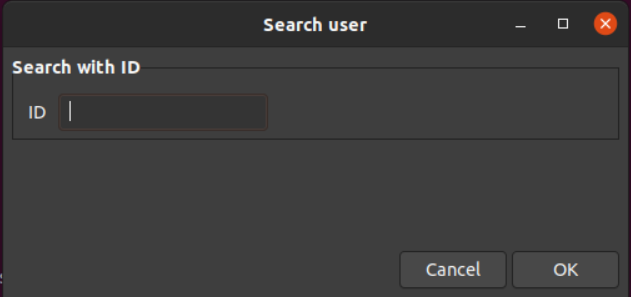
* Once you run the program, it will prompt a pop-up message as below. Click Yes to continue.

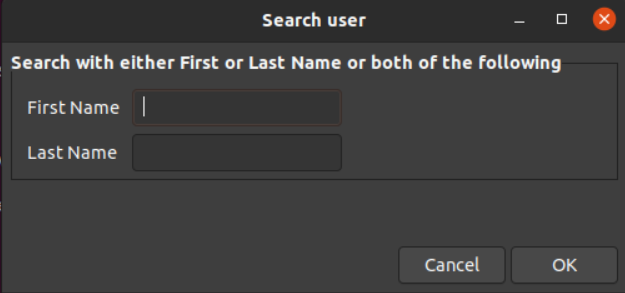


* In the next pop-up message, choose ID or Name and then click OK.

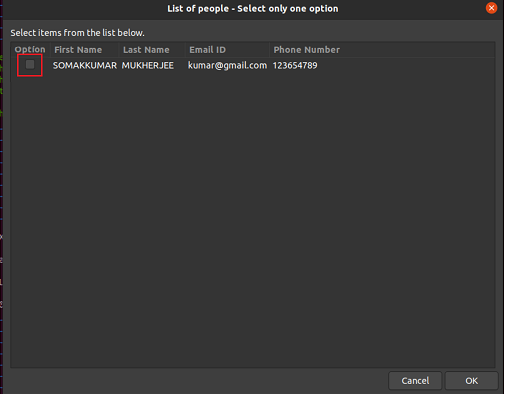


* If you know the person's ID or First Name or Last Name, then fill it and click OK.

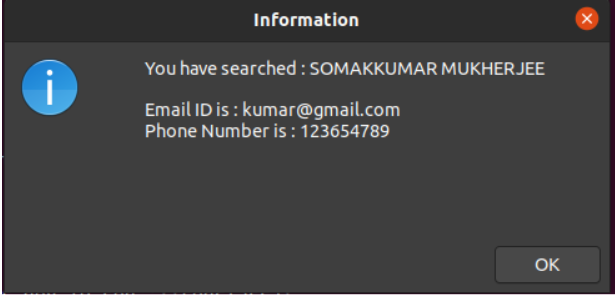




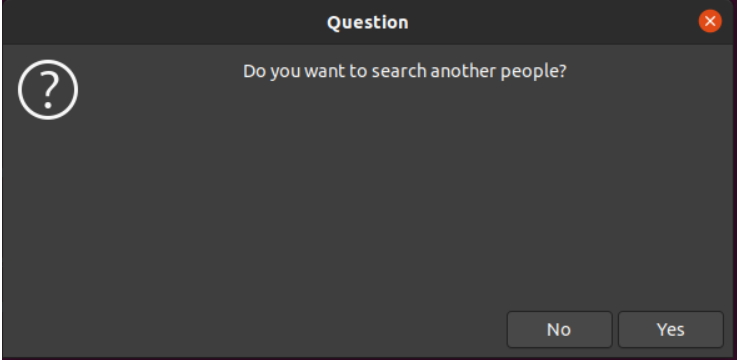
* After you pressed OK in the previous pop-up screen, this process will search the person in the database and pop-up the person's details, as shown below. If the details are not present in the database, this process will prompt as "Details are not present." Select the option box and click OK.



* After you pressed OK in the previous pop-up screen, it will show the information of the searched person as below.

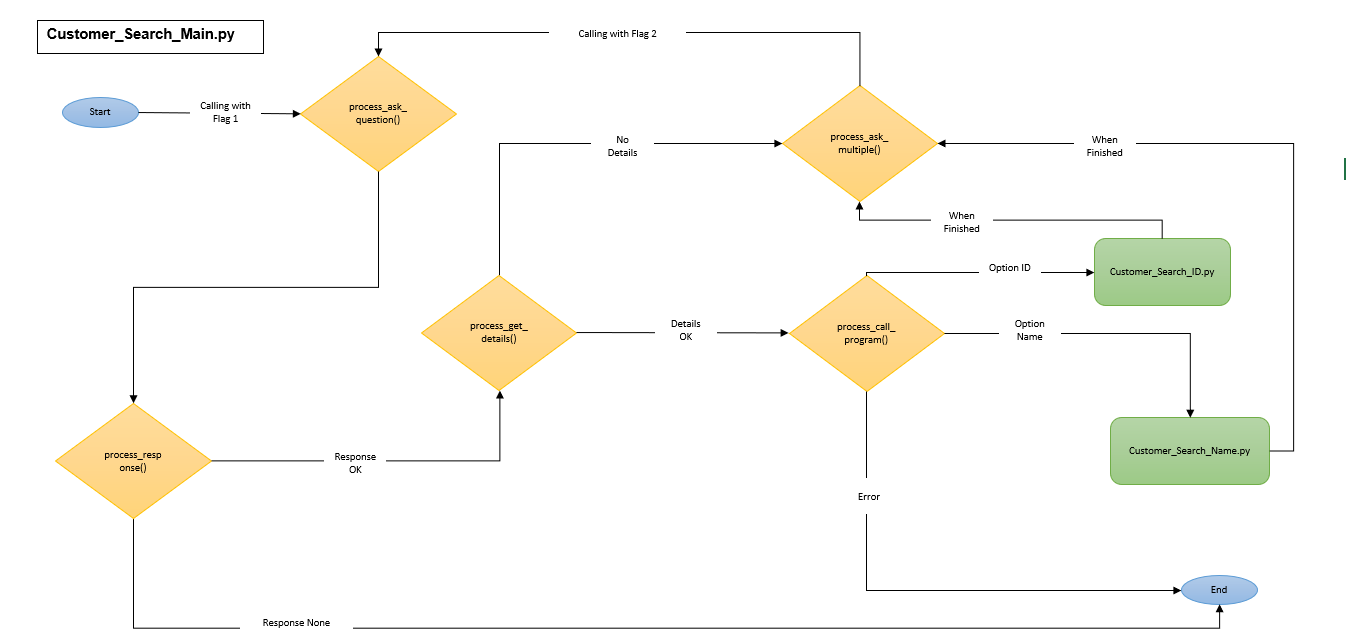


* At last, this process will ask if you want to search again. Click Yes to continue or No to exit.

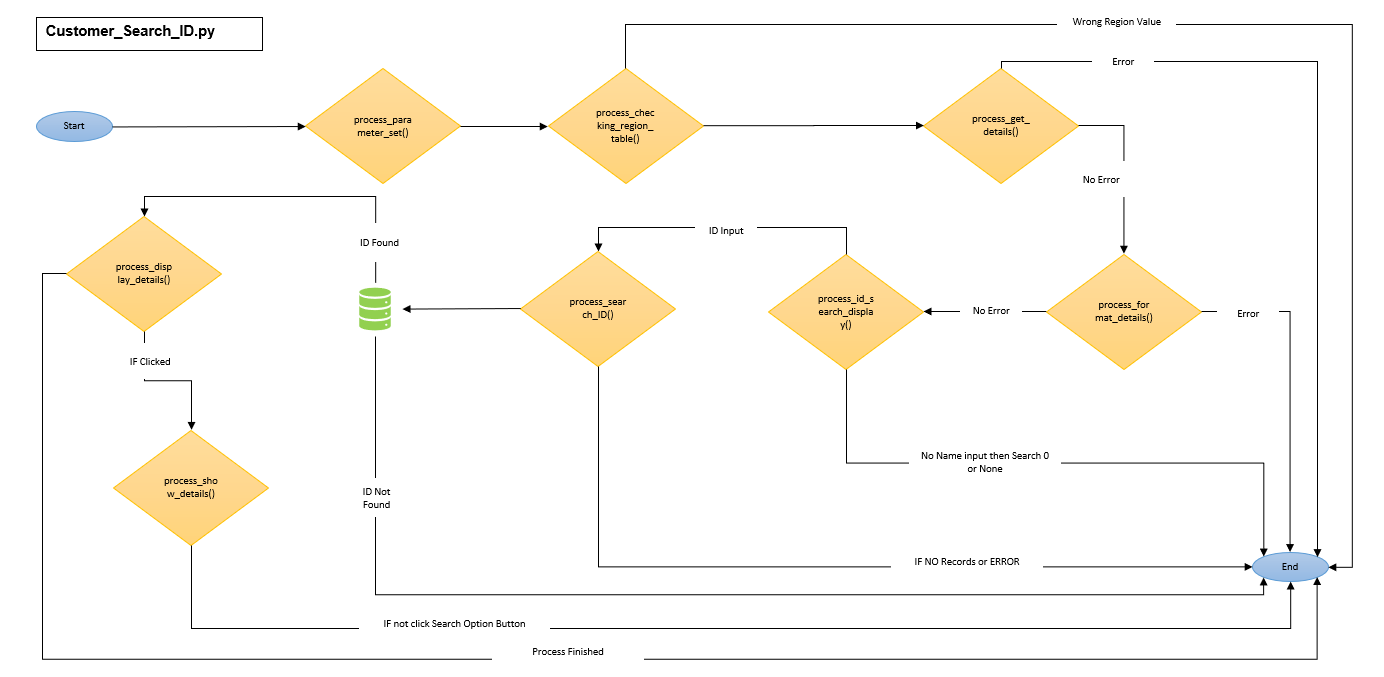


**Flowchart**

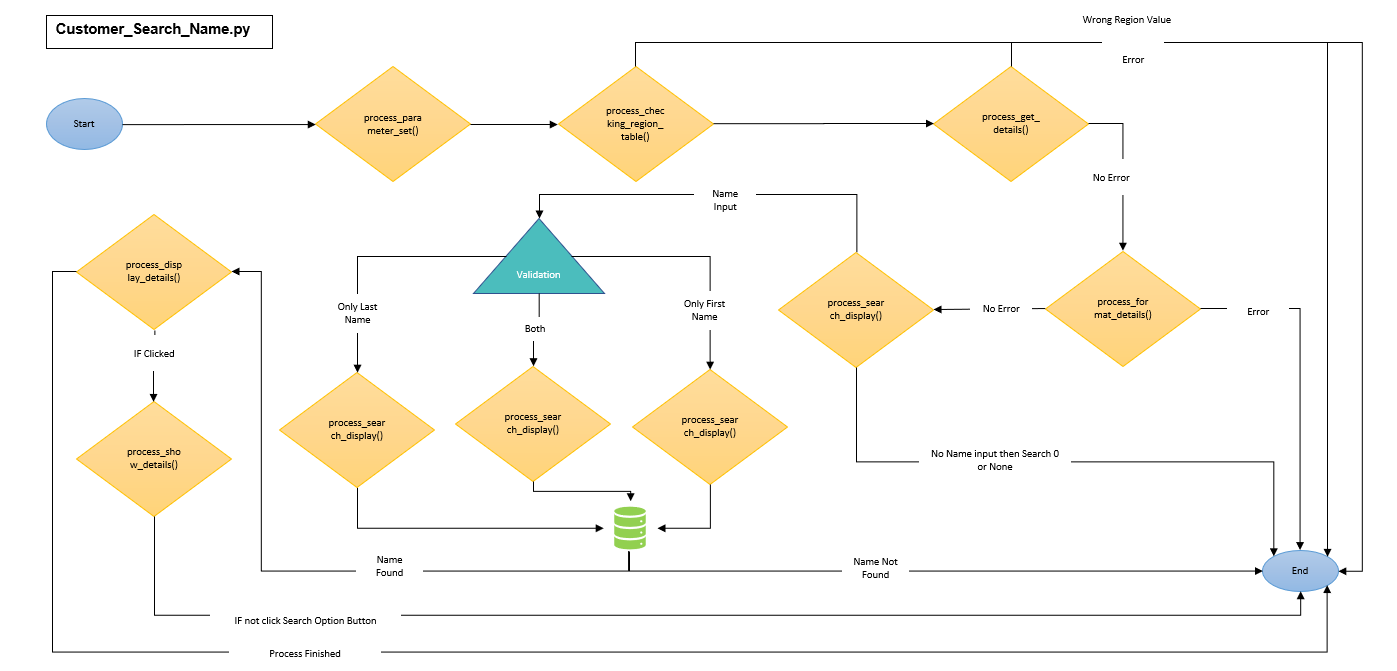
Following is the flowchart architecture of the Customer\_Search\_Main.py program.



Following is the flowchart architecture of the Customer\_Search\_ID.py program.



Following is the flowchart architecture of the Customer\_Search\_Name.py program.



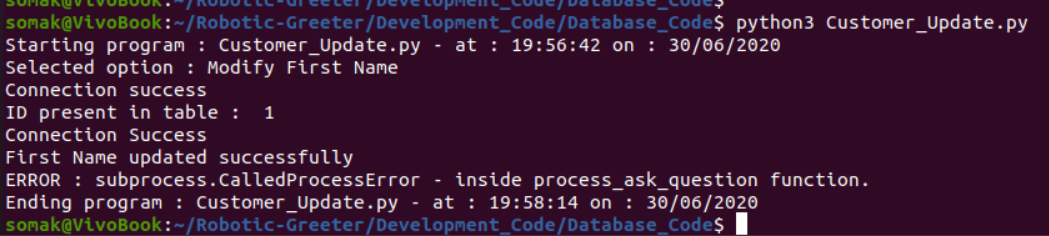
**3. Customer\_Update.py**

Customer\_Update.py is used to update the data of the user into the table mentioned above. This program will be called from Main\_Process.py if the user wants to update their details. To update the details, the user should know their unique ID, as this program will ask to confirm the ID before updating.

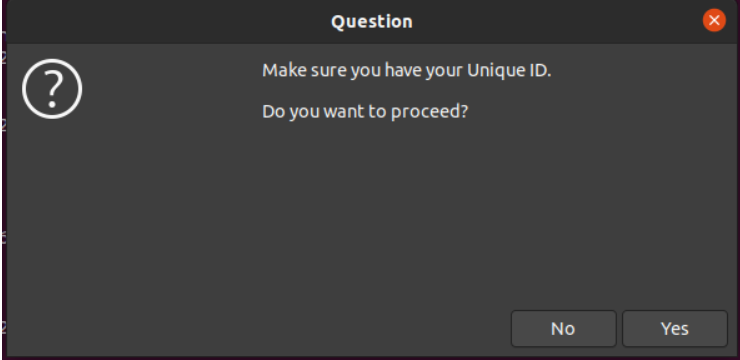
**Execution:**

* To run this program, run the below command from your command prompt.

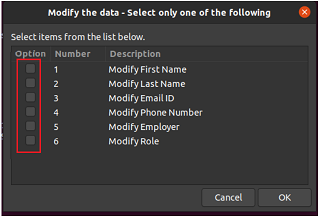
$ python3 Customer\_Update.py



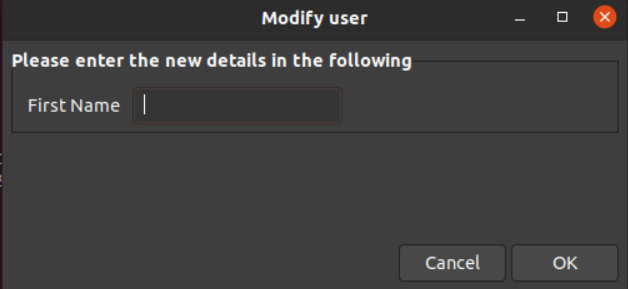
* Once you run the program, it will prompt a pop-up message as below. Click Yes to continue and No to exit.



* On this screen, choose the option field that you want to update. You cannot select multiple options at a single time. Then click OK to continue.

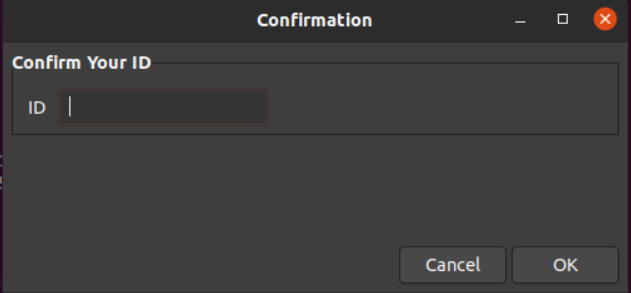


* Fill the new details that you want to modify and then click OK to continue. But all the fields have some validation.

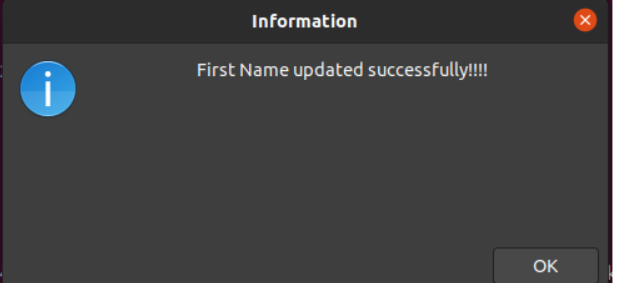


|  |  |
| --- | --- |
| **Fields** | **Description (Validation)** |
| First Name | It cannot be less than two characters or numbers. |
| Last Name | It cannot be less than two characters or numbers. |
| Email ID | It cannot be less than seven characters and has to include @ and dot. |
| Phone No | It cannot be less than seven numbers and must be numeric. |
| Employer | It cannot be less than two characters or numbers. |
| Role | It cannot be less than two characters or numbers. |

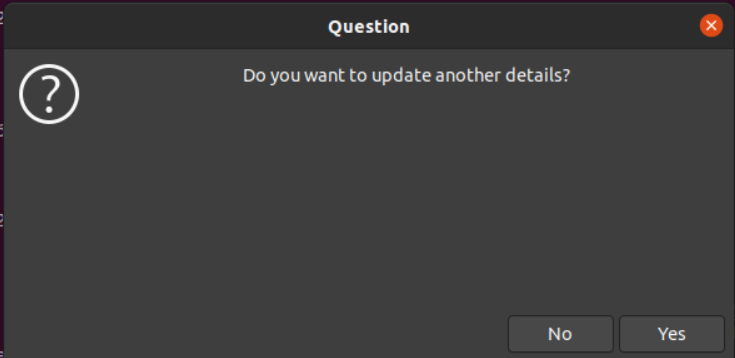
* Then this process will ask to confirm your UNIQUE ID because you can only modify your details. Fill your ID and click OK to alter the details.



* Now, this program will validate your ID (present in the database or not) and will update your details. If this process can modify your details, it will prompt a pop-up message as below, but if it cannot be able to change, it will prompt an error message.



* At last, this process will ask if you want to update any details again. Click Yes to continue or No to exit.



**Flowchart**

Following is the flowchart architecture of the Customer\_Search\_Main.py program.

