

# CODEX Level 3 Program

---

Author: Zack Knight

Challenge: Level 3

1. Create a command-line based user interface program that:
  - a. Accepts 3 command line parameters
    - i. Required: name and age
    - ii. Optional: phone number
  - b. Stores the data in a database
  - c. Supports the following commands:
    - i. User:
      1. Add
      2. Remove
      3. Edit
    - ii. Database:
      1. Export
      2. Clear
    - iii. Print all users (in table format, sorted by name, with all attributes) to stdout or a file

---

## Program Design

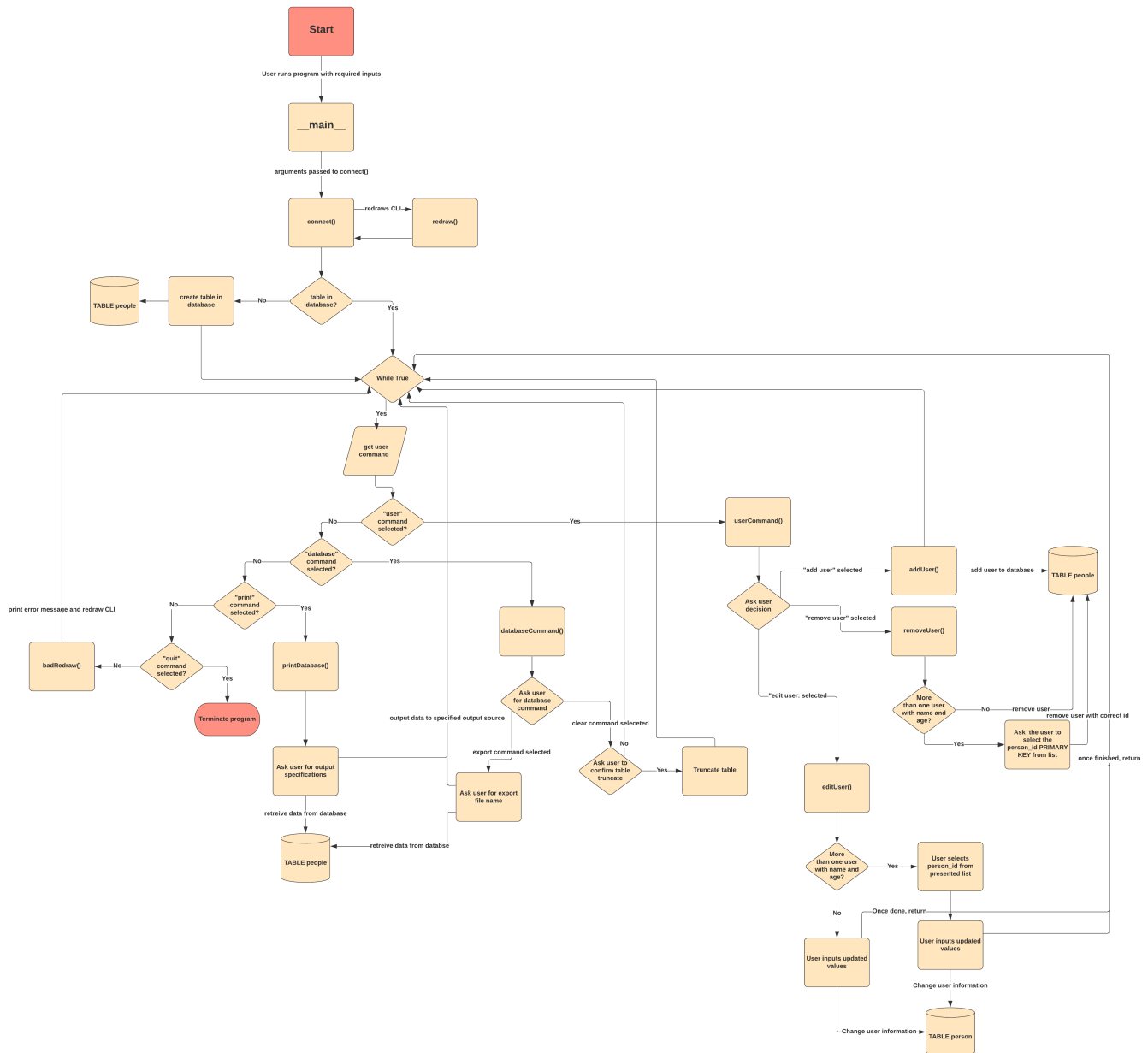
None of the design fundamentals I learned in my Software Engineering class really showed how the program functioned in a way that I wanted.

The result is that I made my own version of a flow diagram. It is a little packed, but it covers the entirety of how the program operates at a high level.

With that said, here is a [link](#) to view the full chart in LucidChart, or you can paste this:

```
https://www.lucidchart.com/invitations/accept/58e40c4d-4c06-4486-add8-6a679c44df8e
```

You may also view the image right below, though it will be difficult to see everything clearly.



## Database table layout

The table that is being used has the label ***people***.

It is set up like this:

person_id	person_name	person_age	person_phone	time_created
1	Test User	50	None	2020-05-08 05:22:24.360292

Where they were created with the following types:

- person\_id = SERIAL PRIMARY KEY
- person\_name = VARCHAR(255) NOT NULL
- person\_age = INTEGER NOT NULL
- person\_phone = VARCHAR(50)
- time\_created = TIMESTAMP WITHOUT TIME ZONE DEFAULT (NOW()) at time zone 'utc')

# Setting up for program execution

## Setting up PostgreSQL

I chose to use PostgreSQL, as that is what I figured would be the most likely to be used, and it was easy for me to set up.

Please make sure that PostgreSQL is installed and then change what you need to in the ***database.ini*** file.

I use Linux when I program, so it was as easy as installing using my package manager and adding a new user.

You **do not** need to create a table in the database. The program will handle that for you.

## Setting up Python

The program was written in python 3.8, so I would recommend using that version. If you do not have that version, I would at least make sure you have 3.7 installed.

There are two ways to get the pip environment set up correctly.

1. Use pipenv (Recommended)
2. Install using pip or a virtual environment

### Using pipenv (recommended)

The way that I would recommend to set up the python environment is to use pipenv. You can install it if you do not have it by using the command `pip install pipenv`. Make sure that you are using the python3 version of pip.

Once pipenv is installed you can then set up the pipenv.

Run the command `pipenv install` and it should detect the Pipfile.lock file and install the correct versions of the required pip packages.

Once it is done, you can run `pipenv shell` and continue on to running the program.

### Using the regular pip or a virtual environment

I did not use this method when writing this program, so there is no guarantee that you will get the correct environment set up.

That being said, I would recommend using a virtual environment like venv over the system pip.

I will leave it up to you to install and set up the virtual environment.

Once it is activated or if you do not want to use one, you can run:

```
pip install -r requirements.txt
```

I just hand copied the packages from the Pipfile, so it should work.

## Running the program

You should now be good to go after setting everything up.

You can see the help options with:

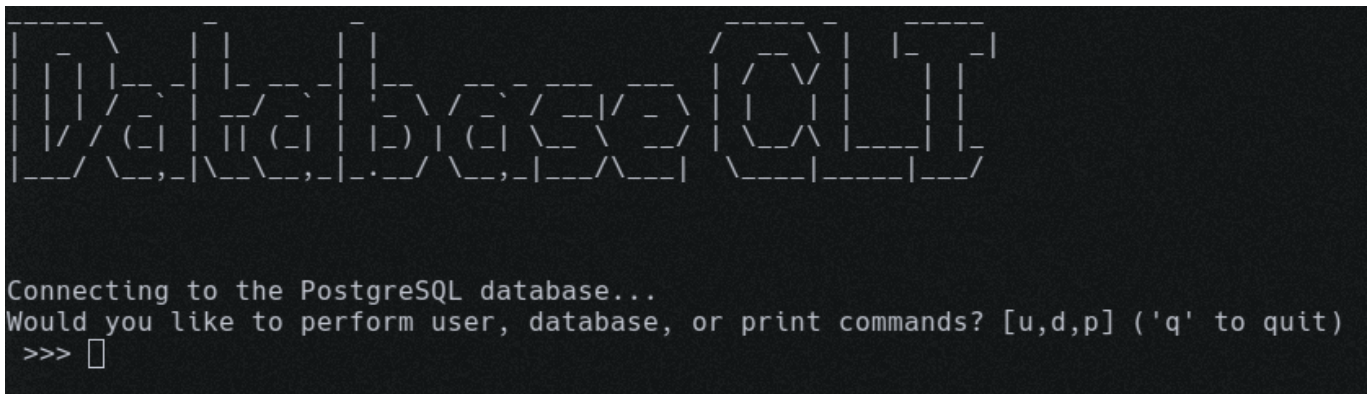
```
python level3.py -h
```

You must pass in the arguments for name and age. The phone number is optional.

An example run command is:

```
python level3.py -n "Test User" -a 50
```

You should then be presented with a screen like so:



```
Database CLI
Connecting to the PostgreSQL database...
Would you like to perform user, database, or print commands? [u,d,p] ('q' to quit)
>>> 
```

The options are printed out before you.

- To perform user actions (add, remove, or edit the user): type **u** and hit enter.
- To perform database actions (export or clear the database): type **d** and hit enter.
- To perform a print command (Print to STDOUT, a text file, or csv file): type **p** and hit enter.
- To quit the program, type **q** and hit enter
- If the input is not what the program is expecting, you will get an error message and be sent back to input your choice again.

If you are in an immediate submenu (You just entered **u**, **d**, or **p**), you can enter **back** to go back to the main menu and select another choice.

After completing a main menu action other than print, the CLI should redraw itself for a cleaner interface.