CS3130
Assignment #4

For this assignment, you will take your employee database program from Assignment #1 and convert it to a TCP-based client and server.

The client portion will be responsible for interacting with the user. Its responsibilities include:

- present the menu
- handle incorrect inputs
- send query to server using a well-defined (and well-documented) protocol
- wait for server reply
- display output

The server portion will handle the database duties. Its responsibilities include:

- wait for client request
- address bad requests
- process the clients request
- return the results using a well-defined (and well-documented) protocol
- wait for another query

The main screen for client should be like the one shown below:

```
--
Employee FMS

Select one of the following:

   1) Add a new employee
   2) Search for an employee
   3) Remove an employee from FMS
   4) Display entire employee FMS

Option?

--
```

Each option will present a new menu. An invalid option / input must be flagged.

Option 1 will ask the user to enter the 4 fields and the program will then attempt to insert that record in the database. If a record with that employee ID already exists, the addition will fail and user will be asked to try to re-enter the record. If addition is successful, user is notified of this and is then asked if they want to enter another employee or return to main menu.

Option 2 will ask user to enter an Employee ID and will display the details for that given Employee ID. If employee doesn't exist, user should be notified of this. After displaying the details, program will ask user if they want to display another employee or return to main menu.

Option 3 will remove an employee from the database. Program prompts user for an employee ID. All

removals must be confirmed and attempts to delete non-existing employees must be flagged.

Option 4 will display all records in the database

Your client /server will use TCP as the underlying transport with the server listening on port 2015. I want to see written documentation on the protocol that client and server use to communicate. This documentation should include the format of messages, order in which messages are sent/received as well as how the protocol deals with bad / poorly formed messages.

Your program must be written to exploit modules. All modules/functions must be documented. All possible exceptions (database files not found / illegal /invalid values) must be caught and handled appropriately. **Your program's user interface must be clear and its output easy to read**. Your final project must include a README file, 2 or more python script files (one for the database handling functions and another for the user interaction /main function) and a test database.
Franco Carlacci