EMS Program – Final Project

Steven On

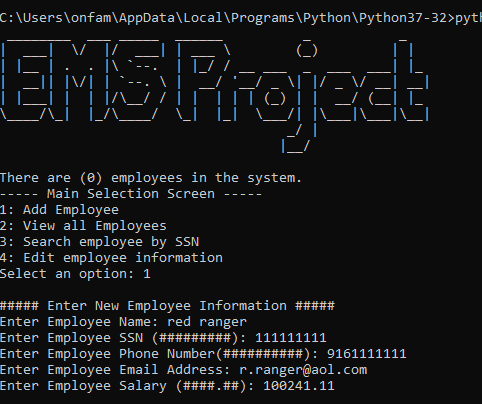
CPT200: Fundamentals of Programming Languages

Professor Amjad Alkilani

June 24, 2019

EMS Program – Final Project

Screenshot of functionality one:



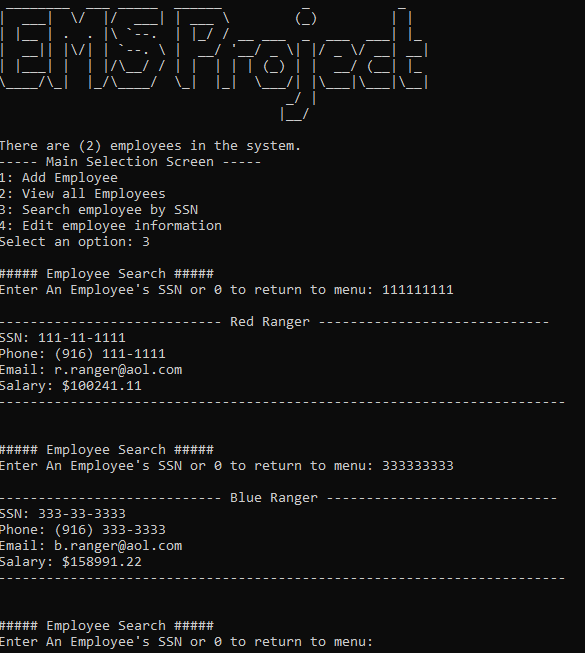
Functionality one allows the user of the employee management system (EMS) to add employee data which contains the following employee information: Name, SSN, Phone Number, Email Address, and Salary.

Screenshot of functionality two:



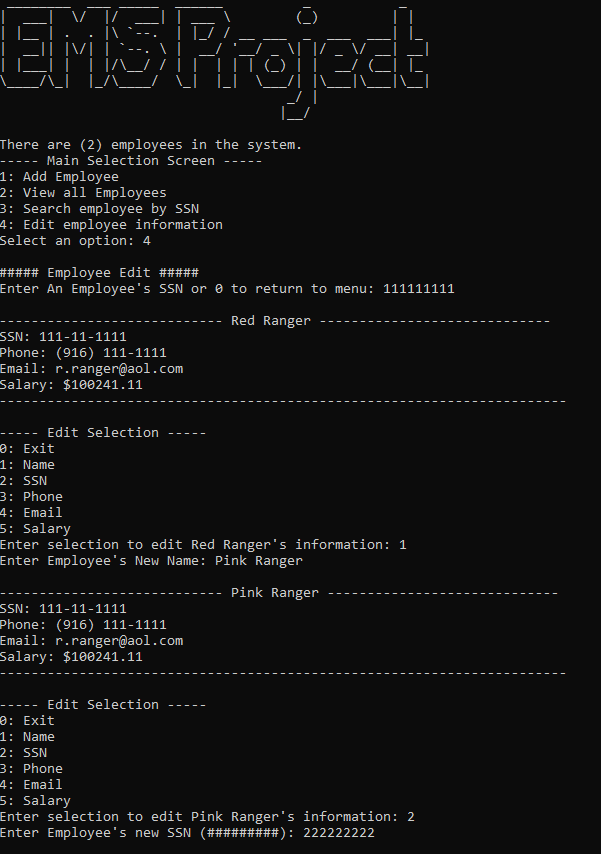
Functionality two allows the EMS user to display the employee information that is currently entered in the EMS. The user is presented with a selection list of employees by name and then the user can select the corresponding number for the employee data they would like to view.

Screenshot of functionality three:

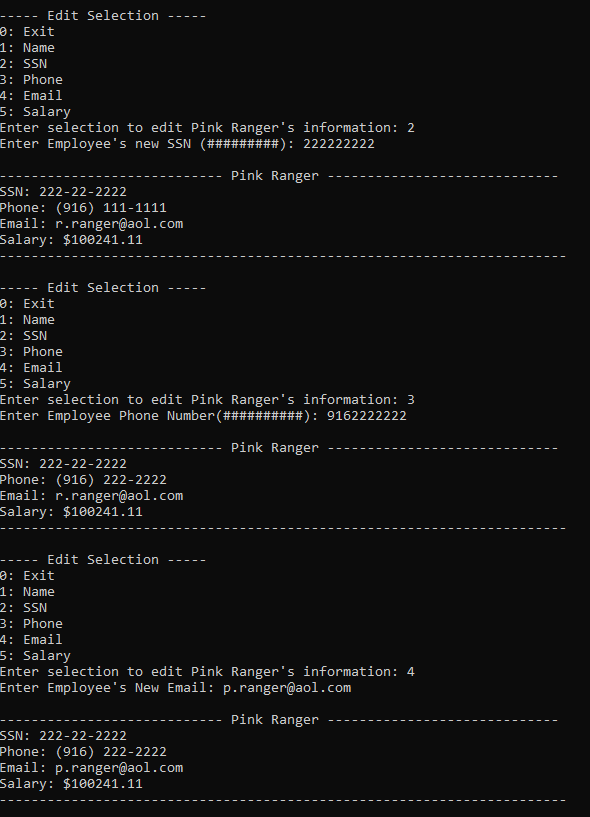


Functionality three allows the EMS user to lookup and display employee information by entering the employee’s SSN. Once the SSN is entered, the system will retrieve and display the located employee information.

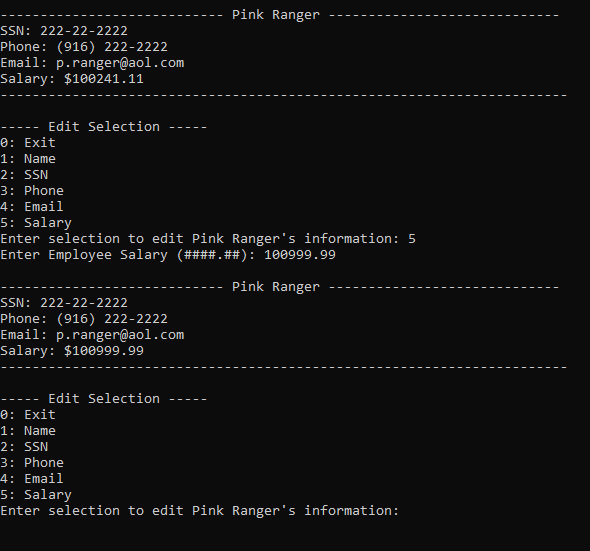
Screenshot of functionality four (Name Change):



Screenshot of functionality four (SSN, Phone, & Email Change):

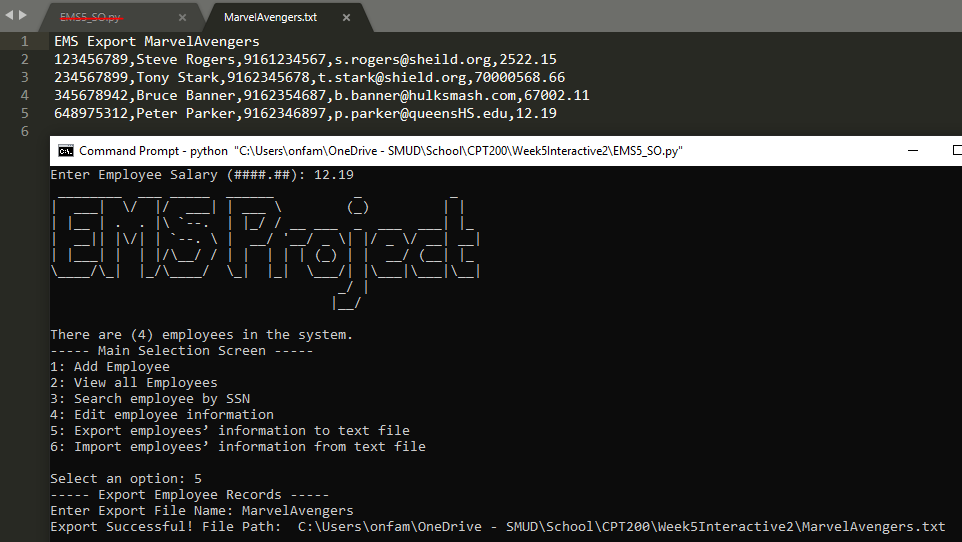


Screenshot of functionality four (Salary Change):



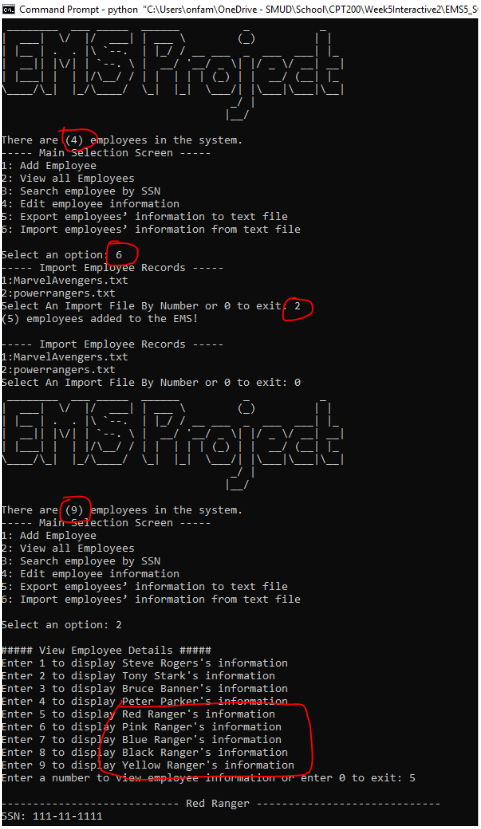
Functionality four utilizes the search by SSN feature from functionality three. After the EMS user finds the employee by SSN, the user is then presented with options to edit the employee’s: Name, SSN, Phone, Email, or Salary.

Screenshot of Functionality five:



Functionality five allows the user to export all employee data that is currently stored in the EMS. The EMS user is prompted to enter the name of the text file that will be generated in the same location of the Python script. Exporting employee data into a text file will allow the user to save the employee information that has been already entered in the EMS. By exporting, the EMS user will be able to use functionality six, importing at any time.

Screenshot of functionality six:



Functionality six allows the EMS user to import an exported text file. This functionality will allow the EMS user to work with previously entered employee data.

I did some scouting on my assignments for this course and looked at week five during week one. I saw that we were working toward building an EMS which contains six functionalities. After each weekly assignment, I kept appending code to my EMS project script. Below, you will find the outline of my EMS code:



As new functionalities were rolled out each week, I’ve appended new option selections for the user and added “Elif” statements based on the user’s selection. This was not as simple as putting a jigsaw puzzle together, I had to think heuristically as I started to uncover new things that Python can do such as: dictionaries, functions, classes, working with files, and error handling. Each week my entire code changed slightly to match what I wanted the EMS to perform based on each week’s assignment prompt. I now understand the main purpose of an EMS.

An EMS allows an organization to enter, view, edit, export, or import employee information. It’s a barebones repository to store and access employee information for use. In a sense, we’ve built our own Database Management System (DBMS) for employee data. If this system was running continuously and, on a timer, to export files for redundancy purposes this can be used as a fast file system database. And it makes me think that this EMS is like NoSQL databases such as MongoDB (MongoDB, 2019). Did I just build my own MongoDB?

In conclusion, the completed script is located in the .zip file contained in my week five, final project. I couldn’t find a clean way to present my script in word due to screenshot sizing restraints. Thank you for this opportuity to allow me to lean Python programming!

References

mongoDB. (Date Accessed: 2019, June 25). The database for modern applications. Retrieved from <https://www.mongodb.com/>