**UCSC Extension, Summer 2016**

**Python For Programmers**

**Final Project: Banking Management Software**

**Student: Truc Ngo**

**I> Introduction**

This program can be used to in a banking environment on daily basis. It has some common features that can be found in any similar software. It has both user mode and admin mode. Admin mode requires a password.

**II> Requirements**

This software requires Python interpreter 2.x and SQLite installed in the host machine. The database file *bankingDB.sqlite* and the Python program *Banking.py* should be stored in the same directory. To execute the program, at console log, type > *python Banking.py*

**III> Descriptions**

This program has 10 features, which are listed below:

1: Open new account

2: Check balance

3: Deposit

4: Withdraw

5: Close an account

6: Check Promotions

7: Show active customer accounts (ADMIN)

8: Show \*\*ALL\* customer accounts (ADMIN)

9: List customers by joining date (ADMIN)

10: Save customer list to text file (ADMIN)

Feature#1: Open new account

Basic information for new customer will be acquired: SSN#, name, age, and balance. Each of these fields will be validated for sanity check using regular expression. Balance is of type float. A field called “joining date” will be created automatically and stored inside the record. Another Boolean field called “Active Account” will also be created; this field indicates that the account is currently active. Customer information will be stored in a class “Customer” before it is saved to database. Customer information is store as a tuple before it is passed to corresponding functions in which it will be un-packed. SSN# is used as primary key and it cannot be NULL.

The program will interact with customer if the provided SSN# is already existed in database.

*For the simplicity of the project, date of birth, phone number and address of customer will not be required.*

Feature#2: Check balance

SSN # will be required. This field is validated using regular expression. It returns customer’s name and account balance.

Feature#3: Deposit

Customer can deposit an amount of type float to his or her account. SSN # is required. SSN# and the deposit amount are validated using regular expression.

Balance of the account will be displayed after the deposit. This function also checks for existence of the account in database.

Feature#4: Withdraw

Customer can withdraw an amount of type float from his or her account. SSN # is required. SSN# and the withdrawal amount are validated using regular expression.

Customer cannot withdraw an amount greater than the current balance.

Balance of the account will be displayed after the withdrawal. This function also checks for existence of the account in database.

Feature#5: Close an account

SS# is required which is validated using regular expression. It does not delete the account from database. Instead, a Boolean flag is set in database is set to “False”. There are 3 steps to be followed: 1. Customer is provided a check for the amount of the current balance. 2. The balance is set to zero and 3. The “active” flag is set to “False”.

Feature#6: Check Promotions

This feature generates a random number, each of which corresponds to a promotion to be displayed to customer.

Feature#7: Show active customer accounts

This function requires ADMIN privilege. It asks for a password. It basically sends a query to the database to retrieve the information. It returns active customer accounts only.

Feature#8: Show ALL customer accounts

This function requires ADMIN privilege. It asks for a password. It returns both active and non-active customer accounts. This method utilized an overloaded function.

Feature#9: List customers by joining date

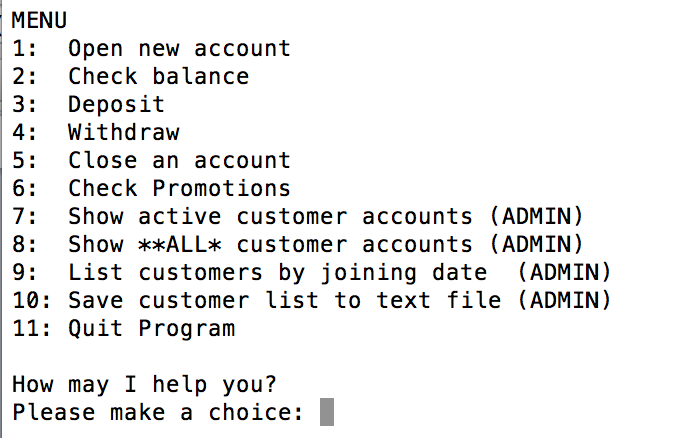
This function requires ADMIN privilege. User is required to provide two dates: start date and end date. This data is validated using regular expression. It returns all account: both active and closed. Returned data is sorted by joining date.

Feature#10: Save customer list to text file

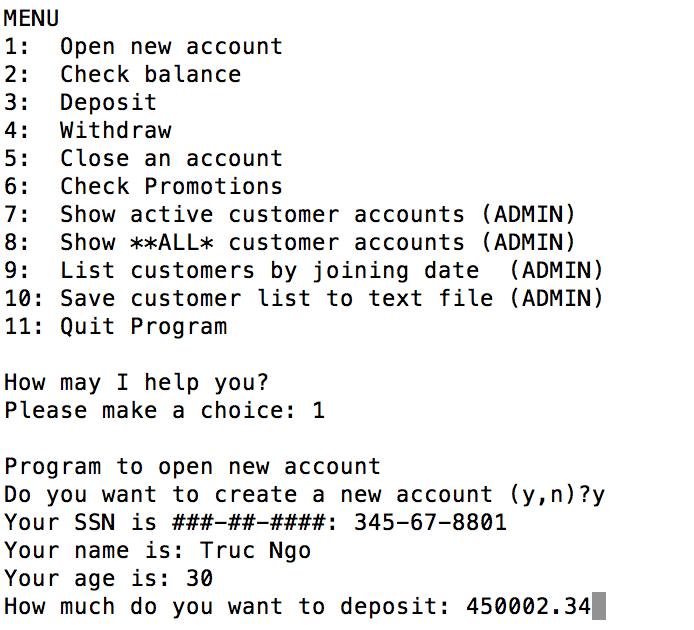
This function requires ADMIN privilege. It saves both active and closed accounts to a text file. This file can be further processed using Excel’s text-to-column feature.

Other support functions are also developed to construct and populate the database.

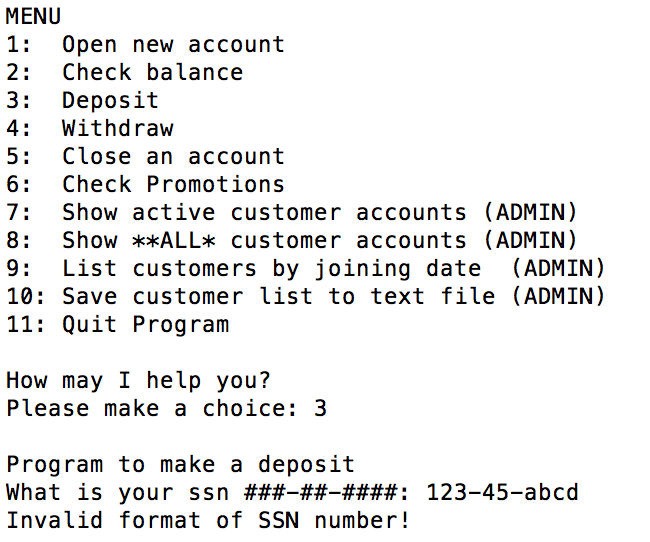
**IV> Screenshots for Programs**



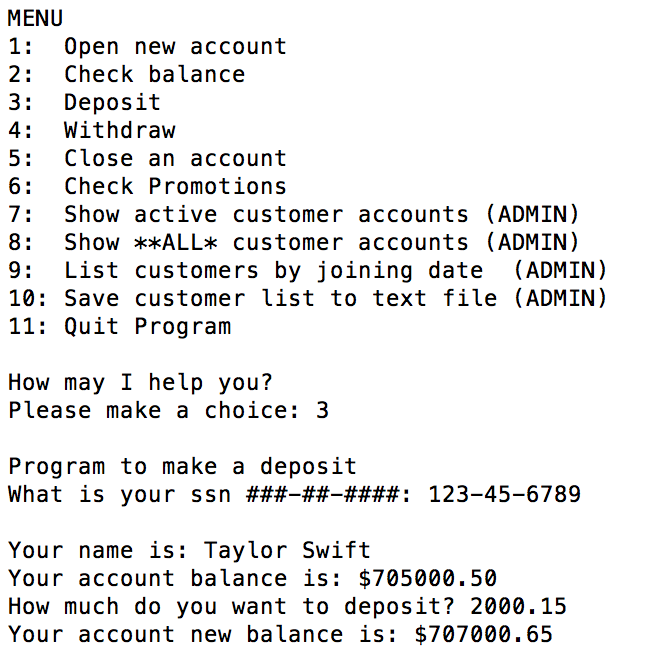
**Photo1: Main Menu**



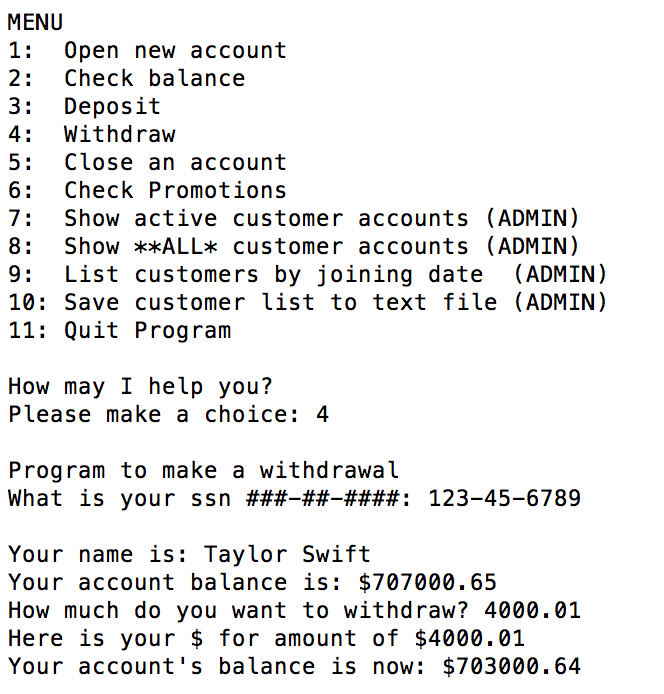
**Photo2: Create new account**



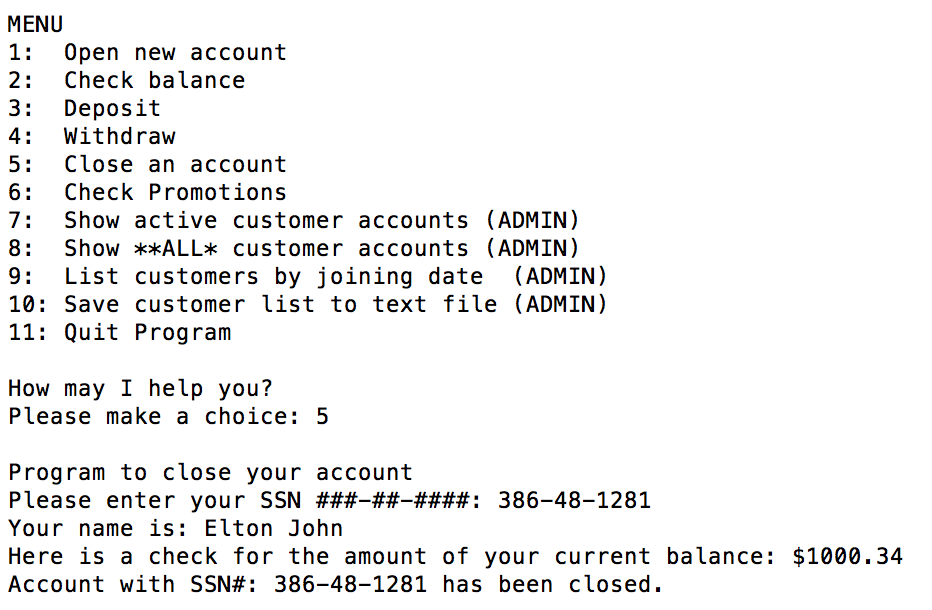
**Photo3: Sanity validation of inputs**



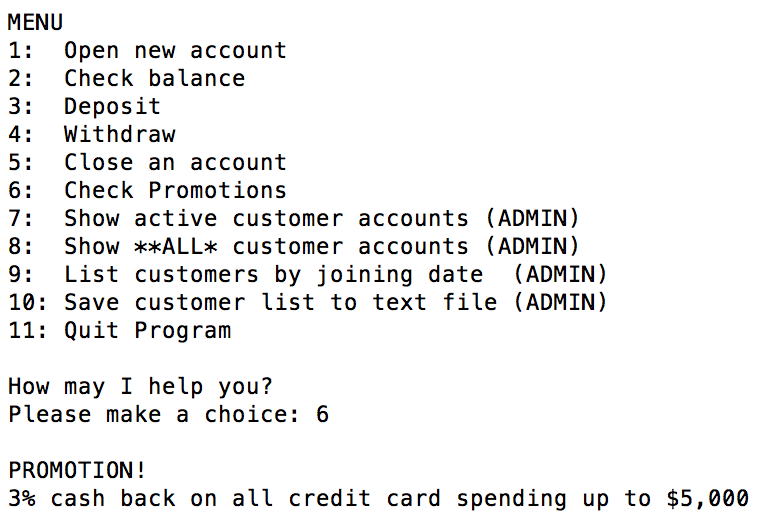
**Photo4: Deposit**



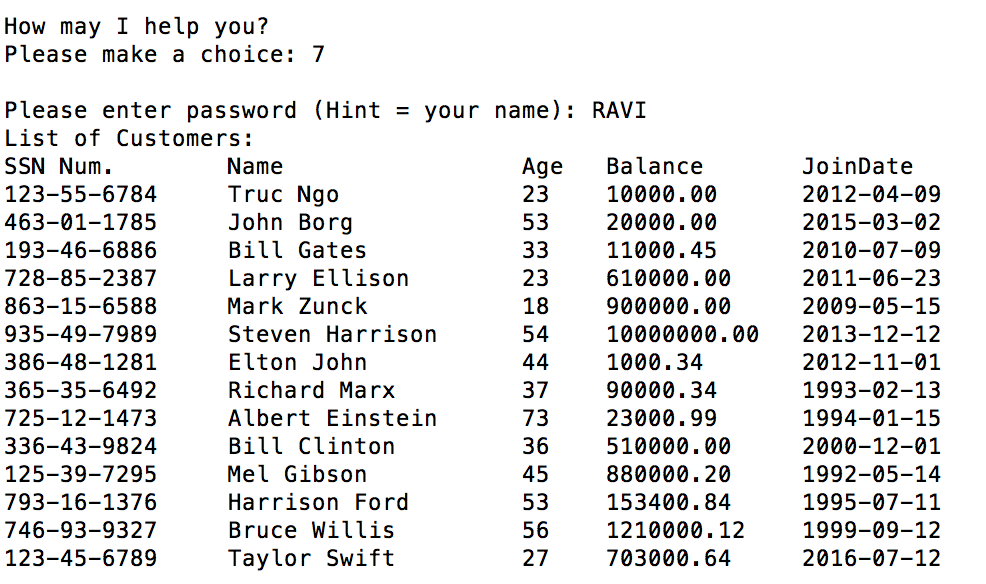
**Photo5: Withdrawal**



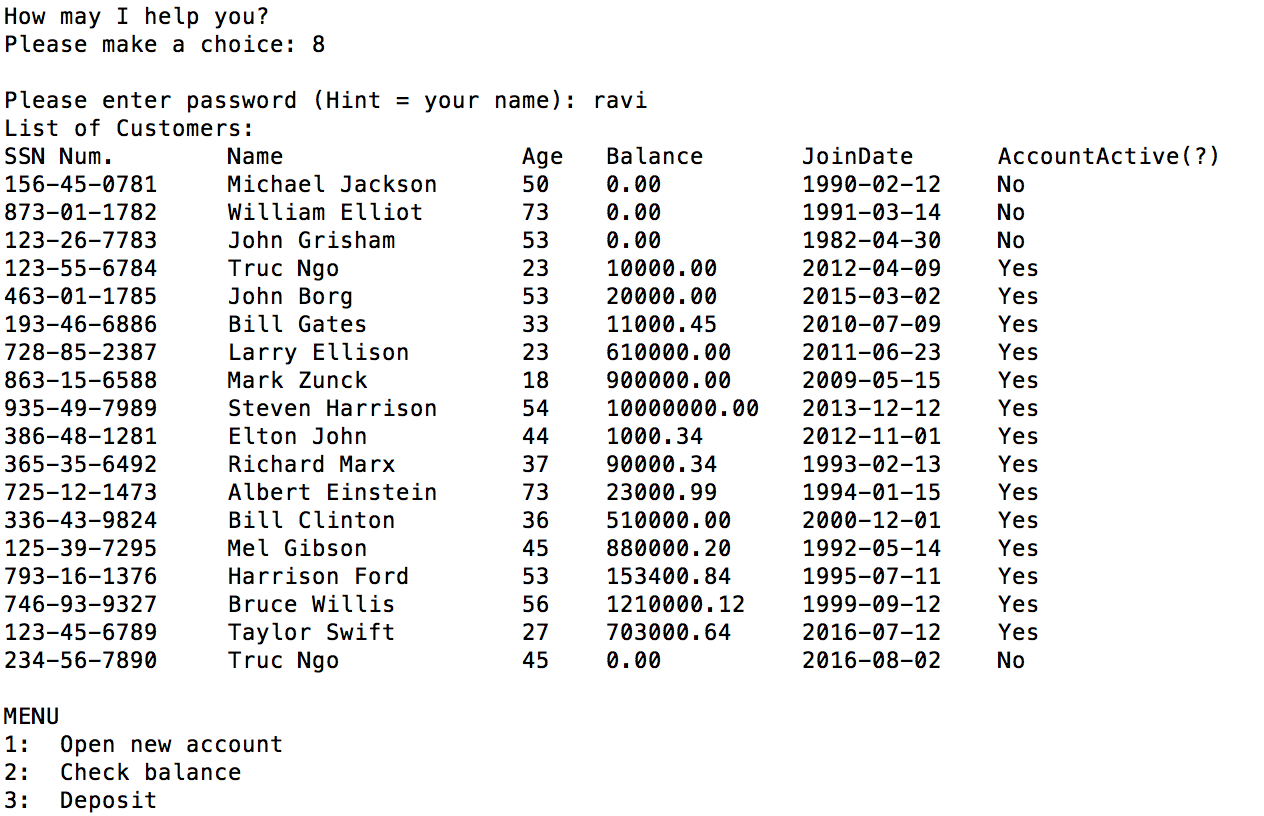
**Photo6: Close account**



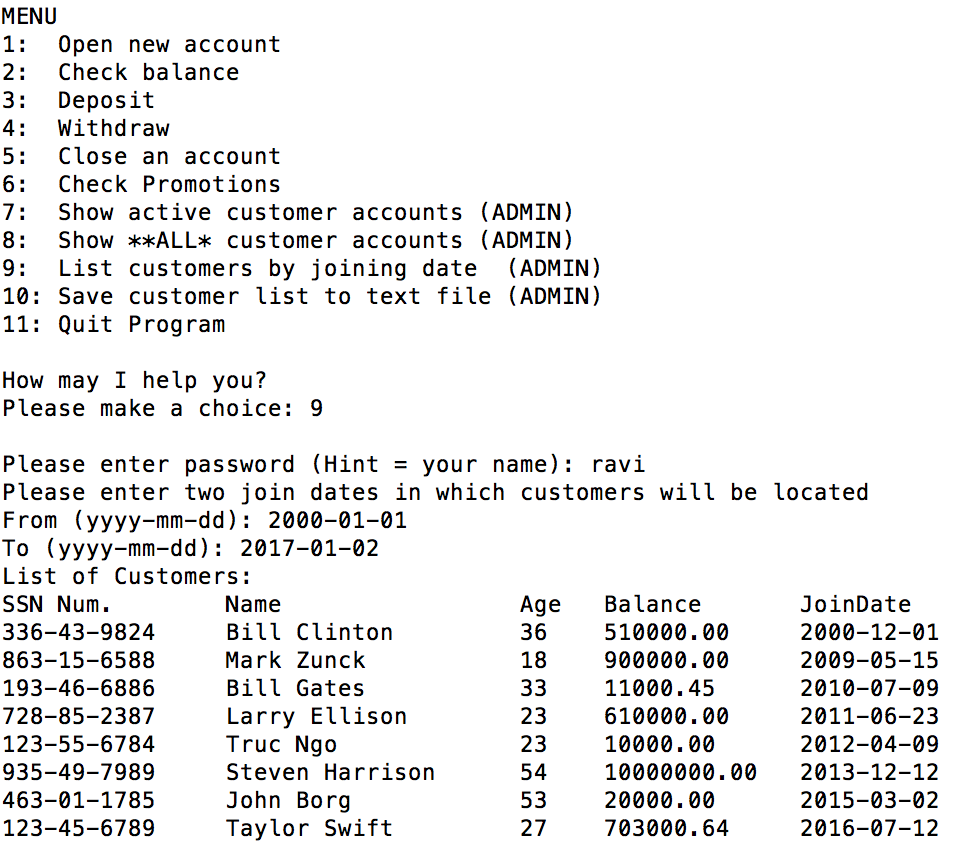
**Photo7: Check for promotions**



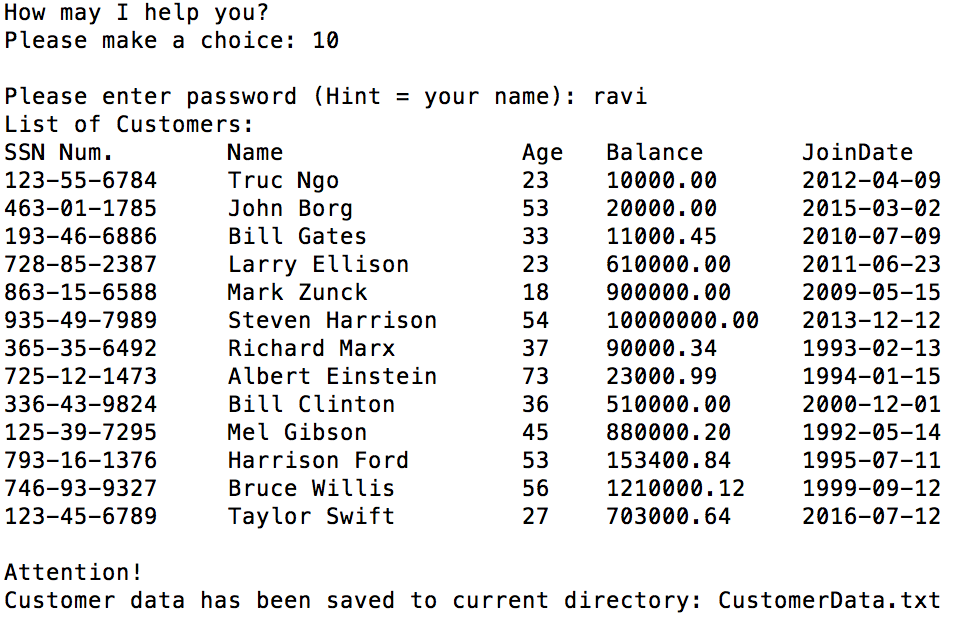
**Photo8: List active customers**



**Photo10: List all customers**



**Photo11: List customers by joining dates**



**Photo12: Save customer data to text file**

**V> Conclusion**

This project utilizes basic knowledge of Python data structure, class, database, file handling and regular expression. It helps me solidify my learning for the language.

**VI> Source codes**

This program was written by student Truc Ngo at UCSC-Extension in Summer 2016.

Source code and database file are included.