BIOS6642-Final Report

Contents

0.	Login()	2
	Login()	2
	MENU	2
	EXIT	2
1.	Display()	2
	Display()	2
	Exit()	3
2.	Statistics()	3
	Statistics()	3
	Exit()	4
3.	Subset()	4
	31. ByAge()	4
	32. ByResidency()	4
	33. ByBalance()	5
	34. ByDate()	5
	Exit()	5
4.	Search()	5
	41. ByAccNum()	6
	42. ByName()	6
	Exit()	6
5.	Signup	6
	Signup()	6
	Exit()	7
6.	Cancellation()	7
	Cancellation()	7
	Exit()	8
7.	Operation()	8
	72. Withdraw()	9
	73. Transfer()	9
	74. Show()	. 10
	75. Change()	. 10
	76. History()	. 10
	Exit()	. 11
8.	Record()	. 11
	Record()	. 11
	Exit()	. 11
9.	Menu()	. 11
	Menu()	. 11
99	9. Logout()	. 12
	Logout()	.12

O. Login()

Login()

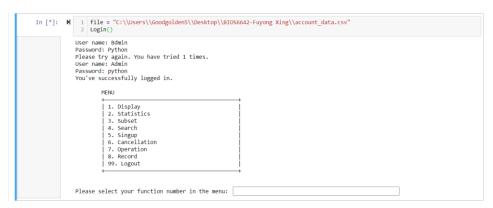
A required condition for the Login() function is to provide the directory location of the "account_data.csv" file. The successful Login() requires one predefined "UserName" and a matching "password" in case sensitive condition. The function provides the maximum three times of failure trails and lockdown after that. The function also provides how many times you have failed. Here assume all of us use the same UserName (i.e., Admin) and password (i.e., python).

```
In [4]: | H | 1 | file = "C:\\Users\\Goodgolden5\\Desktop\\BIOS6642-Fuyong Xing\\account_data.csv" | 2 | Login() |

User name: 1 | Password: 1 | Please try again. You have tried 1 times. | User name: 1 | Password: 1 | Please try again. You have tried 2 times. | User name: 1 | Password: 1 | Please try again. You have tried 2 times. | User name: 1 | Password: 1 | Please try again. You have tried 3 times. | Please restart the system.
```

MENU

After successful user login, the system should automatically initialize the customer account database by using the file "account_data.csv" and then the system will ask the user to provide inputs for account management, the MENU shows up and requires your action to perform next time. Each integer number represent its function on the menu. Input of wrong type data will cause ValueError(several functions are prepared with ValueError and Exception term, but not all of them).



FXIT

This is one special menu for the user to proper Logout() of the system. Normally the EXIT menu will show up after each function; if the user any other "input" besides "Y" or "N", the system will return to EXIT. In this EXIT, the user can select either to return to functional MENU or Logout().

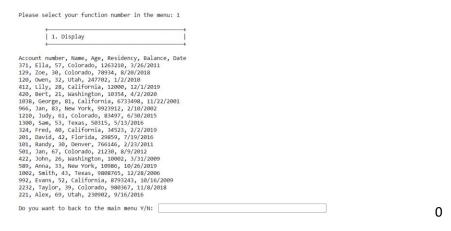
9. Menu 99. Logout	
Please select your function number in the menu:	

1. Display()

Display()

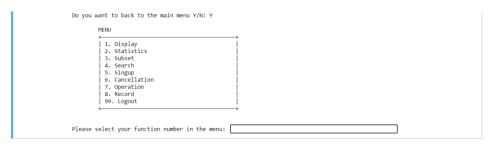
Let us look at the first function, Display() in the menu. It is a simple function, as long as you enter 1, the function will return all the information contained in the file. (1) account number, (2) account

owner's names, (3) account owner's age, (4) account owner's residencies, (5) current balance, or (6) the date when the account is opened.



Exit()

After the Display(), the program will ask you whether to return to the main MENU. If you choose "Y", the program will return to the MENU. Otherwise the program goes to EXIT and requires further action for properly exit the program.



2. Statistics()

Statistics()

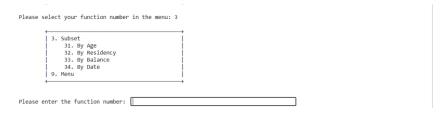
The next function in the MENU is Statistics(). This operation displays (1) total balance, average balance and median balance of all accounts, and (2) total balance, average balance and median balance of accounts for each following age group: (a) under 35, (b) 35 - 44, (c) 45 - 54, (d) 55 - 64, (e) 65 - 74, and (f) 75 and older.

Exit()

After the display the statistics, the system will ask the user's action: "Y" to return back to MENU and "N" to return to the EXIT.

3. Subset()

Display a subset of accounts on the screen, given certain criteria. After select Subset() function, the program will present you a SUBMENU of the Subset() options:



31. ByAge()

This function displays the accounts with the owners having ages between *min age* and *max age*, which are positive integers given by the user during runtime.

32. ByResidency()

This function displays the accounts with the owners having a residency, residency, which is a string provided by the user during runtime.

```
Please enter a state residency: Colorado
Account number, Name, Age, Residency, Balance, Date
371, Ella, 57, Colorado, 1263210, 3/26/2011
129, Zoe, 30, Colorado, 18934, 8/20/2018
1210, Judy, 61, Colorado, 83497, 6/30/2015
591, Jan, 67, Colorado, 21230, 8/9/2012
2232, Taylor, 39, Colorado, 980367, 11/8/2018
Do you want to check another subset Y/N:
```

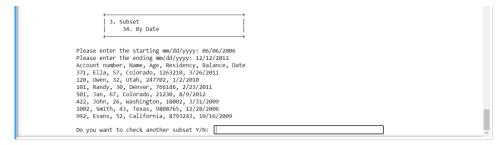
33. ByBalance()

This function displays the accounts with the owners having current balances between *min balance* and *max balance*, which are positive integers given by the user during runtime.



34. ByDate()

This function displays the accounts with opening dates between *min date* and *max date*, which are strings provided by the user during runtime.

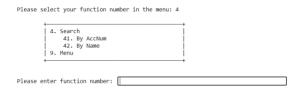


Exit()

After each operation in Subset(), the project will require your action: "Y" to try another option in Subset(); "N" to return back to MENU page; or any other type of input to EXIT page.

4. Search()

This function will also provide the user a SUBMENU as shown below and require your next option.



41. ByAccNum()

Query and display an account given an AccNum. If the given AccNum does not exist in the current account database, the system provides potions to Search(). The user can reenter ByAccNum() and enter another AccNum.

```
4. Search
41. By AccNum

Please enter your AccuNum: 999
Account number, Name, Age, Residency, Balance, Date
AccNum does not exist

4. Search
41. By AccNum

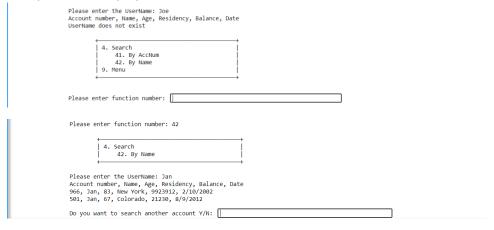
Please enter your AccuNum: 201
Account number, Name, Age, Residency, Balance, Date
201, David, 42, Florida, 29859, 7/19/2016

Do you want to search another account Y/N:
```

If multiple accounts are found given an account owner's name (note that the owner's name is not unique), please display them all on the screen.

42. ByName()

Query and display owner's Name. If the given account number or owner's name does not exist in the current account database, the system should print a proper message and return back to Search(). If multiple accounts are found given an account owner's name (note that the owner's name is not unique), the system will display them all on the screen.



Exit()

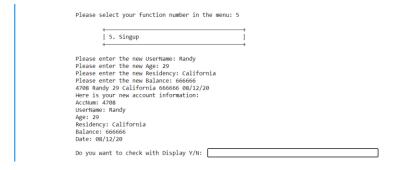
After each option in Search() functions, the system will require the user's action: "Y" to search another account; "N" to return back to MENU; or any other input will return back to EXIT.

5. Signup

Signup()

This function opens a new account. The account owner's Name, Age, Residency and Balance are provided by the user during runtime. The account number is randomly generated between 100 and 9999. Please note that each account number is unique, and it must be

different from the others in the current account database. The date of opening the account is the day when the system runs (i.e., current date).



Exit()

If the account is successfully created, the system automatically updates the account database, i.e., add this newly opened account to the current database, and also updates the file "account_data.csv" on your disk. And the system will ask the user to whether: "Y" to Display() the data file; "N" to return to MENU; or any other input to EXIT.

```
Account number, Name, Age, Residency, Balance, Date
371, Ella, 57, Colorado, 1263218, 3/26/2011
129, Zoe, 30, Colorado, 78934, 8/26/2018
120, Owen, 32, Utah, 247702, 1/2/2010
412, Lily, 28, California, 12000, 12/1/2019
420, Bert, 21, Washington, 10354, 4/2/2020
1038, George, 81, California, 6733498, 11/22/2001
966, Jan, 83, New York, 9029312, 2/10/2002
1210, Judy, 61, Colorado, 83407, 6/30/2015
1300, Sam, 53, Texas, 50315, 5/13/2016
324, Fred, 40, California, 34523, 2/2/2019
201, David, 42, Florida, 29859, 7/19/2016
101, Randy, 30, Denver, 766146, 2/23/2011
501, Jan, 67, Colorado, 21230, 8/9/2012
422, John, 26, Washington, 10002, 3/31/2009
589, Anna, 33, New York, 10986, 10/26/2019
1002, Smith, 43, Texas, 9080765, 12/28/2006
992, Evans, 52, California, 87931231, 30/16/2009
2222, Taylor, 39, Colorado, 980367, 11/8/2018
221, Alex, 69, Utah, 230002, 3/16/2016
4708, Randy, 39, California, 87931231, 30/16/2009
2222, Taylor, 39, Colorado, 980367, 11/8/2018
221, Alex, 69, Utah, 230002, 3/16/2016
4708, Randy, 39, California, 87931231, 30/16/2009
2232, Taylor, 39, Colorado, 980367, 11/8/2018
221, Alex, 69, Utah, 230002, 3/16/2016
```

6. Cancellation()

Cancellation()

This function is to delete an account given an account number, which is provided by the user during runtime. If the given account number exists in the current account database, the system deletes the corresponding account from the database and also from the file "account_data.csv" on your disk.

Please select your function number in the menu: 6	
6. Cancellation	
Please enter the AccNum to delete: 371 Successfully deleted: 371 True	
Do you want to check with Display: Y/N	
Please select your function number in the menu: 6	
6. Cancellation	
Please enter the AccNum to delete: 999 The account does not exist	
Do you want to check with Display: Y/N	

If the given account number is not found in the current database, the system provides a notice and ask the user to re-enter another account number.

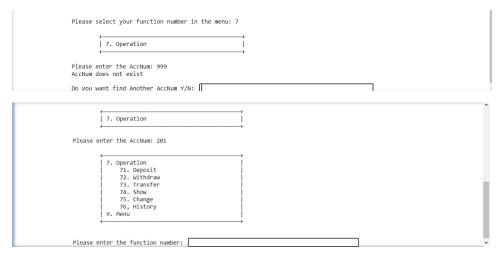
Exit()

If the account is successfully created, the system automatically updates the account database, i.e., add this newly opened account to the current database, and also updates the file "account_data.csv" on your disk. And the system will ask the user to whether: "Y" to Display() the data file; "N" to return to EXIT.



7. Operation()

This function conduct transactions for a specific account. In this operation, the system provides the following options to perform transactions for a given account, which is provided by the user during runtime. After the user provide a valid existing AccNum, the function will directly show the option in a SUBMENU, otherwise the system will ask the user to enter the AccNum again.



71. Deposit()

Deposit money into the given account. The amount of money is provided by the user during runtime. If the deposit is successfully performed, the system automatically updates the account database and also the file "account_data.csv" on your disk. Also, the new information will be added into their "personal history.txt" file and an "overall record.txt" file (if not exist will be created).

If the input is negative or not integer, the system will provide a notice "Please enter a valid deposit amount". The ValueError is overwritten by a warning notice.

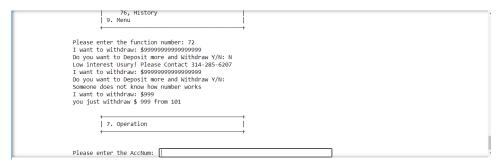
```
75. Change
76, History
9. Menu

Please enter the function number: 71
I want to deposit: $-99
Please enter a valid deposite amount
I want to deposit: $t
Please enter a valid deposite amount
I want to deposit: $99
you just deposit $99
you just deposit $ 99 into 201
```

72. Withdraw()

Withdraw money from the given account. The amount of money is provided by the user during runtime. In order to get a successful withdrawal, the current balance (before the withdrawal) of the account must be higher than the amount that the user intends to withdraw.

If the current balance is lower than the withdraw amount, the system will ask the user for more options For example, the program will ask whether the user need to deposit enough money and then withdraw: "Y" to go to Deposit(); "N" to confirm not enough money; and other input to return to Withdraw() and try again.



If the withdrawal is successfully performed, the system automatically updates the account database and also the file account_data.csv on your disk. Also, the new information will be added into their "personal history.txt" file and an "overall record.txt" file (if not exist will be created).

73. Transfer()

Transfer money from the given account (sender) to another account (receiver). The receiver account and the amount of money to transfer are provided by the user during runtime. In order to get a successful transfer, the current balance (before the transfer) of the sender account must be higher than the amount that the user intends to transfer. If the transfer is successfully performed, the system automatically updates the account database and also the file "account_data.csv" on your disk (note that both sender and receiver accounts will be updated).

If the Receiver AccNum does not exist, the system will provide a notice and let the user enter the AccNum again. If the Transfer amount is negative or not in integer form, the system will provide message and Show() the information of current user; then the system will return back to Operation() to enter their own AccNum agian.

```
7. Operation
71. Deposit
72. Withdraw
73. Transfer
74. Show
75. Change
76, History
9. Menu

Please enter the function number: 73
I want to send to AccRum: 100
AccRum does not exist
I want to send to AccRum: 11
I want to send to AccRum: 11
I want to send to AccRum: 12
I want to send to AccRum: 13
I want to send to AccRum: 14
I want to send to AccRum: 15
I want to send to AccRum: 16
I want to send to AccRum: 16
I want to send to AccRum: 17
I want to send to AccRum: 18
I want to send to AccRum: 19
I want to
```

Also, the new information will be added into their "personal history.txt" file and an "overall record.txt" file (if not exist will be created).

74. Show()

Display all the current information for the given account on the screen.

```
74. Show
75. Change
76, History
9. Menu

Please enter the function number: 74
Account number, Name, Age, Residency, Balance, Date
['420', 'Bert', '21', 'Washington', '11254', '4/2/2020']
```

75. Change()

Account settings such as changing owner's name, age and/or residency for the given account. If this change is successfully performed, the system automatically updates the account database and also the file *account_data.csv* on your disk. The account number, current balance and the date of account opening cannot be changed manually.

```
Please enter the function number: 75
Account number, Name, Age, Residency, Balance, Date
['120', 'Owen', '32', 'Utah', '247702', '1/2/2010']
Please enter a new name: Randy
Please enter a new age: 28
Please enter a new residency: Missouri
Successfully change you indentity
Account number, Name, Age, Residency, Balance, Date
120, Randy, 28, Missouri, 247702, 08/12/2020
```

76. History()

Display all recent transactions for the given account on the screen (not exactly 5) transactions of money deposit, withdrawal, and transfer as well as the activities in account settings, such as changing owner's name, age and/or residency.

Please note that these transactions include the events/activities of money deposit, withdrawal, transfer, and owner's name/age/residency changes that occurred during the last running of the bank account management system. These files are the "personal"

history.txt" saved independently in the directory. When you rerun the bank account management system and want to display the previous transactions for a specific account on the screen, you would need to read the file to get the information of the events/activities.

```
Please enter the function number: 76
AccNum: 120, Name: Owen, Time: 08/12/2020, Action: Receive, Amount: 122, Balance: 247824
AccNum: 120, Name: Owen, Time: 08/12/2020, Action: Receive, Amount: 1230, Balance: 249054
AccNum: 120 Name: Owen Time: 08/12/2020, Action: Showing, Balance: 247702
AccNum: 120, Name: Owen, Time: 08/12/2020, Action: ChangeName, Name: Owen, into: Randy, Action: ChangeAge, Age: 32, into: 28, Action: ChangeRes, Residency: Utah, into: Missouri

AccNum: 101, Name: Dean, Time: 08/12/2020, Action: Withdraw, Amount: 30, Balance: 768303
AccNum: 101 Name: Dean Time: 08/12/2020, Action: Showing, Balance: 768303
AccNum: 101, Name: Bert, Time: 08/12/2020, Action: ChangeName, Name: Bert, into: Randy, Action: ChangeAge, Age: 32, into: 30, Action: ChangeRes, Residency: California, into: Denver
AccNum: 101, Name: Randy, Time: 08/12/2020, Action: Withdraw, Amount: 63, Balance: 766146
AccNum: 101 Name: Randy Time: 08/12/2020, Action: Withdraw, Amount: 9999, Balance: 666147
AccNum: 101, Name: Randy, Time: 08/12/2020, Action: Withdraw, Amount: 999, Balance: 666148
AccNum: 101, Name: Randy, Time: 08/12/2020, Action: Withdraw, Amount: 999, Balance: 665148
AccNum: 101, Name: Randy, Time: 08/12/2020, Action: Withdraw, Amount: 999, Balance: 665148
AccNum: 101, Name: Randy, Time: 08/12/2020, Action: Withdraw, Amount: 999, Balance: 665148
AccNum: 101, Name: Randy, Time: 08/12/2020, Action: Withdraw, Amount: 100, Balance: 665148
```

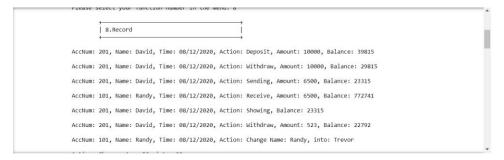
Exit()

After the perform each option, the program will return to Operation(). The user has to properly exit the system through Operation() SUBMENU.

8. Record()

Record()

Different from the personal History() function in Operation(), this function provides the manager to check all the users' events in the records. The function will display all the same texture in each and every individual user's history.



Exit()

After display all the Record() from the "history.txt" file, the system will ask whether to return to the MENU: "Y" to return to MENU; "N" and other input to EXIT.

9. Menu()

Menu()

This is the main MENU display function after each selection of "Return to Menu" or the EXIT choice.

99. Logout()

Logout()

This the function for the user to properly Logout() the system and never return back until next time Login(). The function can be chosen either through EXIT or MENU.