Austin Keelin

Dr. Ku

COMP 2710

30 April 2017

1. **Analysis**

Branch Staff Member: Branch staff members can manage client and account information, and change his/her password.

Login: Branch staff members log in with the username and password initialized by the system administrator.

Staff Menu: Menu used for user to choose the branch staff member actions

Change password: The user is authenticated using their current password then, enter their new password. The new password is valid if it is not equal to the old password and is non-empty.

Manage: Brings up menu run through the Client class. Has options for the branch staff to add a client, add an account, edit client info, manage an account, save client and account info, and exit the system.

Save: Function ran automatically during use, saves the staff info to the staff.txt file

Read File: Function ran automatically during use, reads the info from the staff.txt file

System Administrator: A system administrator can manage client and account information, add/delete/display users, and change his/her password.

Login: System administrator can login with a default username and password. When a system administrator logs into the system, the system administrator can do any of the above.

Admin Menu: Menu in which the user can choose administrator actions for the system to perform

Change password: The user is authenticated using their current password then, enter their new password. The new password is valid if it is not equal to the old password and is non-empty.

Add branch staff: Add new staff members to the teller terminal system. Username and password of the new staff member must be initialized. Empty values will not be accepted.

Display branch staff: User names and roles of all employees including administrators are displayed.

Delete branch staff member: Username is used to identify which staff should be deleted. After the username is entered, the administrator needs to confirm the deletion, and then the staff member is deleted. If the user is not in the system, a warning message will pop up.

Manage: Brings up menu run through the Client class. Has options for the admin to add a client, add an account, edit client info, manage an account, save client and account info, and exit the system.

Client: In this class, the user can add a client, add an account, edit client info, manage an account, and save client info

Add a client: The new client’s name, address, social security number, employer, and annual income must be entered. Client names are used as client identifiers.

Add an account: The client’s name is entered. If the client is not in the system, an error message will pop up. If the client is in the system, then the administrator has to enter the account number, account type, and account balance.

Edit client info: The client’s name is entered. If the client is not found in the system, an error message will pop up. If the client is in the system, then the administrator can edit the client’s information including address, social security number, employer, and annual income. Clients cannot be deleted and client names cannot be changed.

Manage an account: The client’s account number must be entered. If the account is not in the system, an error message will pop up. The administrator can either deposit or withdraw funds

Save client info: The teller system writes all current accounts to a file called account-info.txt and writes all client info to a file called client-info.txt.

1. **Design**

System/Bank

-menu

-login()

-quit()

Admin

Staff

Admin

-admin\_menu()

-changePassword()

-addStaff()

-displayStaff()

-deleteStaff()

-manage()

-save()

-readFile()

System

Client

Staff

-staff\_menu()

-changePassword()

-manage()

-save()

-readFile()

System

Client

Client

-clientMenu()

-addClient()

-addAccount

-editClient()

-manageAccount()

-saveClient()

-saveAccounts()

-readClient()

-readAccounts()

Admin

Staff

Account

**Note: There are a large number of variables, so I found it better to list them instead of including them in the class diagram.**

Bank:

Public: String user, string admin, string password, string role, int index, string array[100], string users[100], string passwords[100], string roles[100]

Private: int option

Client:

Public: int read\_count

Private: string array[100], string names[100], string addresses[100], string ssNumebers[100], string employers[100], string incomes[100], int accIndex, string accArray[100], string accNumbers[100], string accTypes[100], string balances[100], int client\_op, Bank bn.

Admin:

Public: string admin\_user, string admin\_pass

Private: int admin\_op, int index, string array[100], string users[100], string passwords[100], string roles[100], Bank bn, Client cln

Staff:

Public: string staff\_user, string staff\_pass

Private: int staff\_op, int index, string array[100], string users[100], string passwords[100], string roles[100], Bank bn, Client cln

1. **Testing**

addStaff:

1.1: Run addStaff with normal and correct values for name, username, and password

deleteStaff:

2.1: Run deleteStaff with the correct username

2.2: Run deleteStaff with a nonexistent username returning an error message

displayStaff:

3.1: Return the username and role of just the administrator

3.2: Return the usernames and roles of one administrator and three normal staff members

changePassword\_admin:

4.1: Correctly change the password of the administrator

4.2: Enter the current password when attempting to change the password, return error message

changePassword\_staff:

5.1: Correctly change the password of the branch staff member

5.2: Enter the current password when attempting to change the password, return error message

addClient:

6.1: Normal values for client’s name, address, social security number, employer, and annual income are entered

addAccount:

7.1: Create a new account using the correct values of a current client

7.2: Attempt to create an account for a non-existent client, returns error message

editClient:

8.1: Edit the information of a current client

8.2: Attempt to edit the information of a non-existent client, returns error message

manageAccount:

9.1: Deposit amount into the account of a client

9.2: Withdraw money from an account

9.3: Attempt to withdraw more than in an account, returns error message

9.4: Attempt to manage a non-existent account, returns error message