Analysis:

This is a simple bank teller program for the use of employees/employers at Tiger Bank. It is a fully functioning program that will allow the user to manage the accounts of the clients and employees of the bank. For the branch employees, it includes features such as add clients, adding accounts for clients, and editing every characteristic of each client (social security number, employer, and income). The admin has features that includes all those, but also includes adding and deleting branch member and displaying them all for themselves. The program will store all the client's and employee's/employer's data in .txt files in the folder that the program is contained in.

Use Case 1: Admin types in his name and password and successfully signs in. He then chooses to add a branch staff, then confirms and quits the program

Use case 2: User types in name and password, then the program responds saying password is incorrect. User then types in correct name and password and successfully logs in as branch employee. He then chooses to edit client account to change employer name. he then confirms and quits.

Use case 3: User types in name, then the program responds saying this user does not exist in our system. The user then realizes he typed his middle name, and tries his first name which logs him in as branch staff. He then ass a client to the system and add for the client two accounts, one for checking and savings. He then chooses to confirm and quit program.

Use case 4: User types in his admin credentials and successfully logs in. He chooses to display branch staff to make sure all his employees are at work. He notices one person did not show up and quits the program.

Use case 5: User types in name and password and logs in successfully as an admin. He deletes a branch employee after having to fire him for not showing up in use case 4.

Design:

- Interface User
    - Fields
        - name of type string
        - password of type string
    - Functions
        - getName()
            - returns name of user
        - setName(string nameIn)
            - takes in parameter of type string to set name of user
        - changePassword(string pwIn)
            - takes in parameter of type string to set password
- Class Client
    - Inherits from User
    - Struct Account
        - Fields
            - int number
            - string type
            - int balance
    - Fields
        - string ssn (social security number)
        - string employer
        - int income
        - Account accounts[]
    - Functions
        - confirm()
            - Returns boolean to confirm input of user
- Class BranchEmplyee
    - Inherits from User
    - No new fields
    - Functions
        - addClient()
            - function to add new client
        - addAccount(Client clientIn)
            - takes in parameter for client
            - function to add account to client
        - editAccount(Client clientIn)

                - takes in parameter for client
                - function to edit account of client
- Class Admin
    - Inherits from BranchEmployee
    - No new fields
    - Functions
        - addBranch()
            - Function to add new branch employee
        - deleteBranch()
            - function to delete branch employee
        - displayBranch()
            - function to display branch emplyees
- Class File
    - No fields
    - Functions
        - readFile(string filename)
            - will take in file name of type string and returns an array of strings to user
        - writeFile(string fileName, string objects[])
            - Takes in file name and an array of objects to write into the file
- Class System
    - Fields
        - menu of type Menu
        - currentUser of type User
        - file of type file
    - Functions
        - runSystem()
            - this function will initialize all the variables and run the entirety of the program

Test Cases

1. Check to see if readFile function will read every element stored in file
2. Check to see if writeFile function will successfully write every element of array in file
3. Check to see if the program correctly checks the username and password
4. Check to see if addBranch will add a branch emplyee
5. Check to see that a blank password will not work
6. Check to see that editAccount will make proper changes to account
7. Check to see if deleteBranch will successfully delete a branch emplyee

8. Check to see that the confirm function is properly functioning