Semester Project

**Due: See the detailed schedule in the Project progress plan**

**Goal of this assignment:** To be able to bring together all the learning from the course to build something substantial

**Project:** A bank is building a new software in C. The requirements for this software are the following:

1. There should be 2 types of accounts, an administrator account and a customer account. To start off the administrator account user must be “admin” (login), “admin” (password)
2. The login screen should allow both administrator and Customer to login to the system

Welcome to Online Banking/ATM System

====================================

Enter your Customer/Admin ID: 12345 OR admin

Enter your Customer/Admin Password: XXXXX OR admin

1. The administrator shall be able to login to the system. The system should detect when an administrator has logged in. Once logged in she/he shall be able to see a menu as shown below

Welcome to Online Banking/ATM System

====================================

-------------------------

Administrator Menu

-------------------------

1. Create Customer account
2. Change Password
3. View Customer Info
4. Change Customer Info
5. Delete Customer account
6. Show Top 5 accounts
7. Show customer accounts alphabetically
8. Exit
9. Administrator shall be able to create a customer account
10. Administrator shall be able to change his/her own password
11. Administrator shall be able to view customer information
12. Administrator shall be able to change/edit a customer information
13. Administrator shall be able to delete a customer/account
14. Administrator shall be able to view the top 5 accounts (ordered by max balance)
15. Administrator shall be able to view accounts by alphabetical choice. E.g. list all customers whose last name begins with “s”
16. Administrator shall be able to exit the system
17. The Customer should be able to login to the system. The system should be able to detect that a Customer has logged in. Once logged in he should be able to see a menu as shown below

Welcome to Online Banking/ATM system

====================================

--------------------

Customer Menu

--------------------

1. Change Password
2. View Customer information
3. View Balance
4. Make a Deposit
5. Transfer Money
6. Withdraw Money
7. Exit
8. Customer shall be able to change his/her own password
9. Customer shall be able to view his/her information
10. Customer shall be able to view account balance
11. Customer shall be able to deposit into his account
12. Customer shall be able to withdraw from his/her account
13. Customer shall be able to transfer money from his account to another customer’s account using customer ID
14. Customer shall be able to exit the system

**Details:**  All the customer data will reside on a single line in a CustomerData.txt file with the following format. Each field is separated by a single blank.

Customer First Name: 8 characters

Customer Last Name: 8 characters

Customer City: 10 characters

Customer State: 2 characters

Customer Phone: XXX-XXXX (7 characters)

Account Number/Customer ID: XXXXX (5 digits)

Password: XXXXX (6 characters)

Account Balance: (Double)

Every time an administrator or a customer accesses the information from the menu, the data from the file will be downloaded into the program into an array of structure.

The specific customer record will be located and a local copy made for any changes that need to be made. Once the changes are made, upon exit from the menu, the local record will be updated into the main array of records and written back to the file.

When a customer/administrator accesses the data, a lock (implemented as a file) will be activated which will disallow anyone from accessing the file till such a time that the lock is released. When customer/administrator exits the menu, the lock is released.

Administrator can add customer by entering details for new customers or an initial file version may contain all of the information which is a starting data for the system.

You will use structured programming techniques to design a solution to this problem Care must be taken to have a detailed low level design/pseudocode ready before any code is written for this project.

You will have 3 source code files at a minimum. All of the Administrator functionality must be written in administrator.c source file. All of the Customer functionality must be written in customer.c file. You will also create a main.c file that will be responsible for serving the menu for the Customer and the Administrator.

You will compile and build the “banksystem” object. You will run your program by entering the command line on the Linux/Unix system as below:

“banksystem CustomerData.txt”

Thus, you will also design the main function with argc and argv[] as parameters.

**Exception Handling:** You must implement all possible exception handling that is needed to ensure that the program does not crash on invalid inputs. You must also ensure that the bank does not go into a loss because you allow customer to withdraw an amount greater than the balance.

**NOTE**: As queries/omissions are brought to instructor’s notice by students, the project description is subject to tweaking. For many details not specified, it is OK to make reasonable assumptions and mention those in your pseudocode/design.

Students should work in teams of 2. To be able to do so, please inform your instructor and TA on who your team mate is. As a team, you must start working on a menu item and implement it completely as a function before going on to implement the next menu function. Measure your own progress in terms of functions completed and test each function thoroughly.

You must create an initial data for the CustomerData.txt file that has at least 10 customers and one administrator. You may add any field to the CustomerData.txt file to designate that an entry is an Administrator. Mention your strategy in the pseudocode.

**Project Progress Plan:**

**Week 1:** Starting Tuesday April 18th, write a detailed pseudocode/design to solve the given problem, submit the pseudocode to Blackboard. – Due Sunday, April 23.

**Week 2:** Starting Monday, April 24th, Create the framework for the Menus for Customer and the Administrator, submit main.c, customer.c, customer.h, administrator.h and administrator.c to Blackboard – Work it in the lab time and submit by Sunday April 30th to Blackboard.

**Week 3:** Starting May 1, Implement all of the Administrator menu functions, and Customer menu functions and submit administrator.h/administrator.c and customer.h/customer.c to blackboard, Work during lab time and submit by Sunday May 7th to Blackboard. Demo in Lab time

**Week 4:** Test, debug and finalize your code. Demonstrate the functioning code to the instructor in the Lab during this week and complete submission to Blackboard by Thursday, May 11.

Submit the final version of your program to the blackboard. If you have to modify any other code, then do so and submit all changed versions to blackboard.