Core Banking System

Introduction

XYZ Bank provides banking services to users. Once a person becomes a customer of the bank, he/she can perform basic banking operations such credit money, debit money, query transaction details etc.

It is proposed to develop an automated system to support typical operations/services of a bank namely a) user management operations, and b) money transactions.

Scope and Overview:

The scope of the Core Banking System will be to provide the functionality as described in Functional Requirements below. The system will be developed on a Linux box using CPP language and would provide a console-based user-interface.

Functional Requirements

The users of the application are

1. Bank Staff (Clerk) - Perform various functions related to bank operations
2. Customer – Views account and transaction details

Bank Clerk

The bank clerk should be able to Add a customer into the system to create an account
The bank clerk should be able to Delete a customer from the system
The bank clerk should be able to Modify a customer's details
The bank clerk should be able to Credit money to customer account
The bank clerk should be able to Debit Money from the customer account
The bank clerk should be able to Query/Report on customer account & transactions

Customer

The customer should be able to View Customer details
The customer should be able to Transfer money from one account to another
The customer should be able to Query/Report on the account & transactions

Authentication & Authorization

All users logging into the system should be authenticated using a unique login-id and password. (Operations to be supported based on type of user)

The build of the application should be possible using a make file.
The application should handle clean exits (close all open files, descriptors, etc.)
The Core Banking System should be developed as a console-based application.

Detailed Specifications

Bank clerk

The Bank clerk should be able to do the following operations once he has logged in with his unique user id and password:

1. Add a customer into the system to create an account
2. Delete a customer from the system
3. Modify a customer's details
4. Credit money to customer account
5. Debit Money from the customer account
6. Query/Report on customer account & transactions
7. Quit Application

Add a customer into the system to create an account:

As part of this operation, the bank clerk user should have the ability to enter all the details of the customer user (new user) / new account and get confirmation on successful entry.
(Refer to section Data Organization –> Customer User and Account)

Delete a customer from the system:

As part of this operation, the bank clerk should have the ability to delete a particular customer user record and get confirmation on successful deletion.
The system should accept the UserID from the admin to search the corresponding user details in the database.

Modify a customer's details:

As part of this operation, the bank clerk should have the ability to modify all the details of the customer user and get confirmation on the successful modification.
The system should accept the UserID from the bank clerk to search the corresponding user details in the database.
Customer user details to modify can include all the fields mentioned in the add operation.

Credit money to customer account:

Debit Money from the customer account:

As part of this operation, the bank clerk should have the ability credit money into a customer's account and debit money from a customer's account.

(Refer to section Data Organization –> Transaction)

Query/Report on customer account & transactions:

As part of this operation, the bank clerk should have the ability to view all the existing customers' users in the system. He should be able to get the report of a customer's account and transactions.

(Refer to section Data Organization –> User)

Quit Application: As part of this operation bank clerk should have the ability to quit the application completely.

Customer

The Customer User should be able to do the following operations once he has logged in with his unique user id and password.

1. View customer details
2. Transfer money from one account to another
3. Query/Report on the account & transactions
4. Quit Application

View Customer Details: As part of this operation, the customer user should have the ability to view his existing customer details in the system.

(Refer to section Data Organization –> Customer User)

Transfer money from one account to another: As part of this operation, the customer user should have the ability to make amount transfer.

(Refer to section Data Organization –>Transaction)

Query/Report on the account & transactions:

As part of this operation, the customer user should have the ability to key in his user ID (or customer identification number) and password to retrieve his bank account transaction details.

(Refer to section Data Organization –> Account and Transaction)

Quit Application:

As part of this operation, the customer user should have the ability to quit the application completely, saving any active data as necessary.

Data Organization

User Information

Field Name	Data Type	Description
UserID	int	Unique user id in numeric format. It will be a numeric value sequentially generated by the system and need not be passed from the client. It will be added when the admin user creates the new user account.
CustomerID	char	Unique login ID
Password	char	Encrypted Password
UserType	char	U – Customer User A – Administrator
LoginStatus	int	Count of Login Instance can be either 0 or 1. Will disallow any second login attempt using the same Customer ID.

Customer Information

Field Name	Data Type	Description
CustomerID	String	Unique User ID
Name	String	Full Name (First Name, Last Name)
Type	String	Type of the customer like individual, corporate
Address-1	String	Address part-1
Address-2	String	Address part-2
Address-3	String	Address part-3
City	String	
State	String	
Country	String	
Pin Code	String	

Phone Number	String	
Email	String	
Pan	string	

Account Information

Field name	Type	Description
Account Number	Int	
Type	String	Account type – SB Account or Current Account
Customer ID	String	
Status	String	Account status – Enabled or Disabled
Balance	Double	
Opening Date	String	

Transaction Information

Field Name	Data Type	Description
TransID	Int	Primary key for the table
Type	char	Type of transaction – Credit, Debit
AccountIDSrc	Int	Source Account ID
AccountIDDst	int	Destination Account ID
Amount	double	Amount involved in the transaction
Account Number	Int	