Level 3 - Application Development

Instructions:

1. Go through the given application in detail and come up with a design
2. Discuss your design with the invigilators
3. Modify your design to incorporate their suggestions
4. Once the design is finalized, start implementation and show a working demo of the application. Once you start Step 4, you can take **3 hours** to complete the assignment.
5. Try to implement as many tasks as possible

Banking Application

~~~~~~~~~~~~~~~~~~~

Task : 1 - Initialization

We are going to write a software to perform a simple online banking application system.

The back bone for every bank is its customers. The initial customer details for our banking system will be given in a file bank_db.txt

The file contains multiple lines and every line will be of the form

CustId	**AccountNo**	**Name**	**Balance**	**EncryptedPwd**
11	11011	Kumar	10000	ApipNbjm
22	22022	Madhu	20000	Cboljoh
33	33033	Rahul	30000	dbnqvt
44	44044	Robin	40000	kbwb22

....

....

At first, the application will read the contents of this file and initialize its customer base
~~~~~~~~~~~~~~~~~~~

Task : 2 - Add new Customers

There must be provision to add new customers. While adding new customers, get the name and password as input. Ask the user to re-type password to avoid any typos. Ensure that the password and re-typed password match.
You can assume a default initial balance of 10000 and the application will generate a unique customer id and account number. Persist the details of this new customer into the bank_db.txt.

Task : 3 - Encryption

It's generally not advised to store passwords directly in files. So the passwords present in the file are obtained by applying a simple encryption technique to each and every individual character in the plain password.

The encryption rule is very simple - add +1 to original char to encrypt it
a will be encrypted as b
m as n, z as a (wrap around)...
1 as 2, 9 as 0, 0 as 1....
A as B, C as D, Z as A.....

Please take care of encrypting the password while writing to external storage.

Task : 4 - Authentication

Since all the transactions are going to be performed online, it requires proper authentication involving a correct customerid and password. As the users are not aware of and are abstracted from the encryptions applied to passwords, they will input the simple plain password which they once provided!

Task : 5 - Support for basic operations

Our banking application will support the following operations

a. ATM Withdrawal

b. Cash Deposit

c. Account Transfer

For Cash Deposit, on successful authorization, get the amount to be credited and add the same to the existing balance.

For ATM withdrawal, on successful authorization, get the amount to be debited and ensure sufficient funds to debit from the balance. Also by our bank's norms, every customer is expected to maintain a minimum balance of 1000. The application should throw proper exceptions whenever any of these criteria is violated.

For account transfer, on successful authorization, get the beneficiary's account number and amount, make a debit of that amount in the FROM account and perform a credit in the TO account. Ensure sufficient funds and minimum balances as stated earlier.

Task : 6 - Persistence of Transaction History

History of ATM withdrawal, Cash Deposit or Account Transfer should be persisted in the external storage for every user.

Any user after successful authorization, is allowed to check the transaction history. The sample format of history to be displayed is

Account Statement

Name - Kumar

Account No - 11011

Customer Id - 11

TransID	TransType	Amount	Balance
1	Opening	10000	10000
2	CashDeposit	2000	12000
3	ATMWithdrawal	3000	9000
4	TransferTo 11011	2500	6500
5	TransferFrom 22022	3000	9500

The TransID is an auto_incremented application generated value starting with 1 for the Opening Balance. After every successful transaction, persist the transaction details in external storage.

Task : 7 - Top n customers

Support to generate reports like fetching top 'n' customer details based on their current balance

Task : 8 - Password Complexity & Change Password

Password complexity of mandating at least 2 lower case, 2 upper case and 2 numbers with a minimum length of 6

The application should support change password. After authenticating with current password, ask the user to type and re-type the new password. Ensure the new password adheres to the password complexity criteria specified

Then update the new password in user database and hereafter, all authentications should work only with new password for this user

Task : 9 - Password History

The application will remember past 3 passwords for any user and while changing password, the new password should not be same as the last 3 passwords

Task : 10 - Force Password Change

If any user has performed 5 transactions, for security reasons, the application will force that user to change password

Task : 11 - Operational Fee

The bank will charge a nominal fee of 10 from the source account, for any cash transfer whose value exceeds 5000. This should appear in the source account transaction details.

Task : 12 - Maintenance Fee

For every 10 transactions a user performs, an account maintenance fee of 100 will be charged by the bank. The maintenance fee should also appear in the account transaction details. This fee will be waived off for the top 3 customers who has the highest balances at that time.

Ensure that your application continues to run until the user asks to quit