BOS Case Study

Bank of Success Pvt LTD is setting up a bank and need a banking system module to be developed. Requirements for the first module has been mentioned below.

Bank is going to offer 2 types of accounts Savings and Current. Every account will have common information like

Level 1:

a. Account Number (Number should be an auto generated number with starting number as 1000)
b. Name
c. PinNumber (To secure access for every account)
d. Balance
e. Privilge – PREMIUM, GOLD, SILVER (This should be an enum)
f. IsActive
g. Activated Date
h. Closed Date
i. Savings account will have additionally.
    a. Date Of Birth
    b. Gender
    c. Phone No.
j. Current Account will have information like
    a. Company Name
    b. Website
    c. Registration No.

Level 2:

User should be able to Open, Close, Withdraw and Deposit from the account. The rule for Opening an account varies for Savings and Current types of accounts. Logic to Withdraw and Deposit from/into an account is the same for both types of accounts.

1. Open an Account.
    a. Savings:
        i. Validate if Age is greater than 18.
        ii. Set the IsActive status to true.
    b. Current
        i. Check if Registration No is not null.
        ii. Set the isActive status to true.

    Note: If not valid then throw an exception indicating the exact issue and why account could not be opened. AccountImpl Factory should be implemented as given in the design

Level 3:

2. Close
    a. Set the status isActive to false.
3. Withdraw
    a. Check If Account is Active
    b. Check If Sufficient Balance is available
    c. Deduct the amount to be withdrawn from Balance.
    d. Send the status.
    e. If any issue then raise a specific exception to understand the exact failure of rule
4. Deposit
    a. Check If Account is Active
    b. Add the amount to be deposited to Balance.
    c. Send the status.
    d. If any issue, then raise a specific exception to understand the exact failure of rule.

Level 3:

User should also be able to transfer funds which was introduced later by the bank. Transfer of funds should implement the following.

5. Transfer Funds

    Transfer Data:

    a. From Account
    b. To Account
    c. Transfer Amount
    d. Transfer Mode (NEFT, IMPS, RTGS – enum)
    e. Pin Number

    Rules

    a. Both accounts should be active in the system
    b. We should be able to withdraw funds from FromAccount holder.
    c. Deposit funds into ToAccount holder
    d. Limit for transfer in a day is dependent on the PrivilegeType
        a. PREMIUM – 1,00,000
        b. GOLD – 50,000
        c. SILVER – 25,000
    e. The limit must be checked before the transfer is made.
    f. A Transfer Log must be maintained to store the transfers made for a given day.
    g. When a transfer is initiated, we should query this collection for a given day, account and see the number of transactions made, validate the sum and then perform the transfer.

h. Appropriate exceptions should be raised when needed.

All the abstractions to be performed by an AccountManager to manage the operations of an account who will receive the necessary information. A LogManager should be available to store the transfer logs. Querying from the Log should be implemented using Lambda Expressions.

Need to implement.

1. Classes and Interfaces
2. Exceptions – Regular and Custom
3. Abstract Classes
4. Interfaces
5. Design Principles
6. Coding Practices – Best Principles
7. Static
8. Collections – Maps, Lists
9. Java 8 APIs
10. Unit Tests for Account Manager abstractions

Looking forward to a great Design and Implementation!!!