Group 6

Banking

Top Requirements

- There are 3 types of users all with user authentication
- The 3 types are Customer, Teller, Banker, each will have different functions/purposes
- Bankers can create bank accounts for customers
- Tellers deposit or withdraw money from customer bank accounts
- Customers can create Customer Accounts and request to make a bank account
- There will be an ATM portion where customers can deposit or withdraw money from their bank accounts
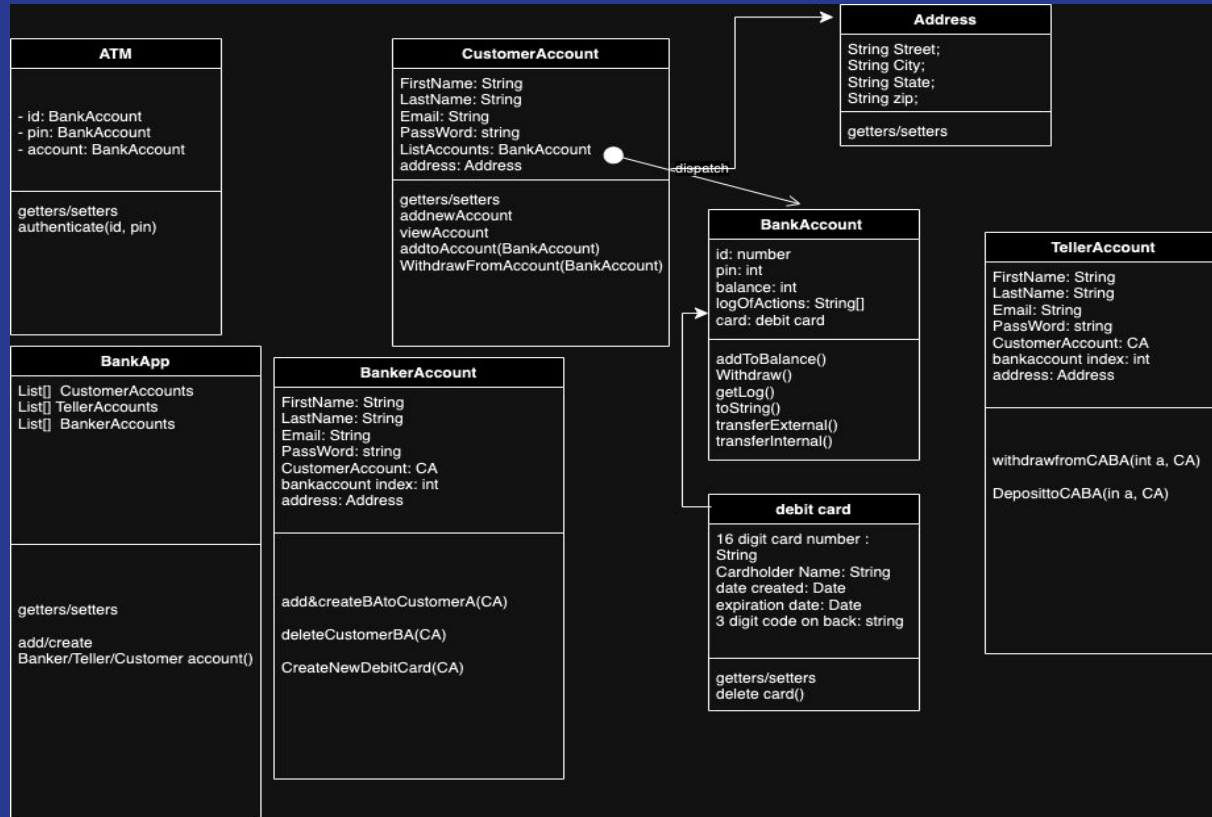
More Requirements

- Customer can request to delete Bank accounts or cancel debit cards
- Banker can delete bank accounts for Customer or create new debit cards
- Withdrawing money has limits on it like you can not withdraw more than what is in the balance of the accounts
- There will be a log of all actions for each bank account.
- Customer can view their account and bank accounts when logged in
- There will be a GUI
- Must Use TCP/IP

The system is organized into 3 parts

- First module is the user management module that allows us to create instances of different types of User Accounts. They also allow users to log in
- The 3 accounts we have are Customer, Teller, and Banker
- The second module is the functional module that allows different users with different functionality like teller can deposit money into Customers Bank accounts
- The Last Module is the ATM module that interacts with the customer user to deposit and withdraw money from their bank accounts

Top Class candidates

ATM

- id: BankAccount
- pin: BankAccount
- account: BankAccount

getters/setters
authenticate(id, pin)

CustomerAccount

FirstName: String
LastName: String
Email: String
PassWord: string
ListAccounts: BankAccount
address: Address

getters/setters
addnewAccount
viewAccount
addtoAccount(BankAccount)
WithdrawFromAccount(BankAccount)

dispatch

Address

String Street;
String City;
String State;
String zip;

getters/setters

BankAccount

id: number
pin: int
balance: int
logOfActions: String[]
card: debit card

addToBalance()
Withdraw()
getLog()
toString()
transferExternal()
transferInternal()

TellerAccount

FirstName: String
LastName: String
Email: String
PassWord: string
CustomerAccount: CA
bankaccount index: int
address: Address

withdrawfromCABA(int a, CA)

DeposittoCABA(in a, CA)

BankApp

List[] CustomerAccounts
List[] TellerAccounts
List[] BankerAccounts

getters/setters

add/create
Banker/Teller/Customer account()

BankerAccount

FirstName: String
LastName: String
Email: String
PassWord: string
CustomerAccount: CA
bankaccount index: int
address: Address

add&createBAtoCustomerA(CA)

deleteCustomerBA(CA)

CreateNewDebitCard(CA)

debit card

16 digit card number :
String
Cardholder Name: String
date created: Date
expiration date: Date
3 digit code on back: string

getters/setters
delete card()

A Use-case

Use Case ID: JBA 003
Use Case Name: create bank accounts for customer accounts
Relevant Requirements: 3.1.3
Primary Actor: User Banker
Pre-Conditions: Banker user must be logged in and select a customer account to view then create a bank account for them
Post-Conditions: the customer user account now has a new bank account with generated id and credentials also a debit card
Basic Flow or Main Scenario:

1. Banker is logged in to system
2. System shows list of customer accounts
3. Banker selects a customer account and selects create a bank account for it
4. If allowed there is new bank account with generated attributes for the customer account

Extensions Or Alternate Flows: If login failure direct back to login or sign up page or if there are no customer accounts then do nothing.
Exceptions: if credentials are incorrect they are not logged in but redirected to sign in, also there might be a maximum amount of bank accounts a customer account can have.
Related Use Cases: JBA 004

Thank you

Top 10 Requirements

1.) Application will have a GUI
2.) Banker- should be able to open checking/savings accounts for customers including generating account numbers/id and debit cards
3.) Teller - should be able to deposit or withdraw from accounts and locate accounts
4.) Atm - section to withdraw money and deposit money directly into customer account
5.) User can have and view multiple accounts
6.) Security - user can create an account with username and password
7.) Data collection- collect usage data of customers like transaction, last accessed, and identity
8.) Cancel transactions if the account is negative balance
9.) User can close accounts
10.) User can cancel debit card/submit to get a new one
11.) Must use TCP/IP