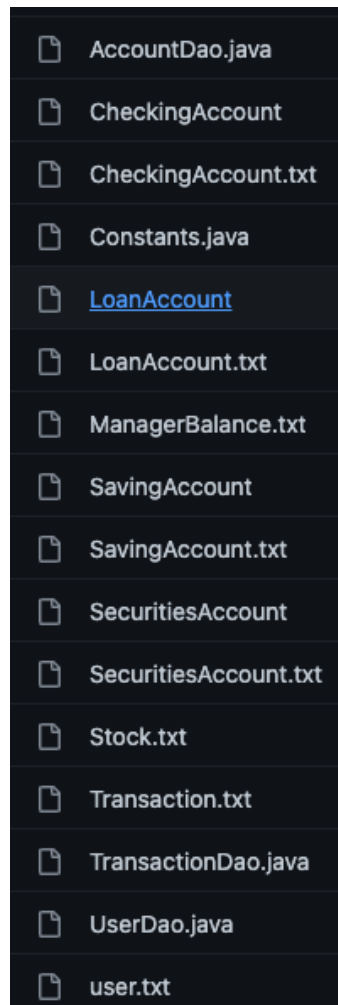


Design Choice

We decide to implement several parts included in this bank application. We have the bank manager which can check-up a specific customer/all customers/the poor ones who owe me, get report on transactions for one day, charge a fee, pay interest on saving accounts, charge interest on all loans, maintaining the list of stocks, updating current price of each stock.

We also have a service-oriented bank object which serves as an online ATM. Additionally, we provide stock market only for the rich customers. On customer end, they can create accounts, maintain deposit up to 3 currencies, take out/request loans, view transactions/current balances, buy/sell stocks and view their current open positions/realized/unrealized profit. For our databases, we have several txt files corresponding to each requirement of different accounts.



Design Architecture & Brief Description of Each File

- Data

AccountDao.java: Interact with data files when account/loan/customers' stock activities occur.

TransactionDao.java: Interact with transaction record file when a new transaction happened.

UserDao.java: Interact with user record file when a new user signs up or is used when looking up a user based on username or ID.

User.txt: Stores user record. [ID, Username, FName, LName, Password, Birthdate, Saving, Checking, Loan, Securities, CardNumber]

ManagerBalance.txt: Stores manager(bank)'s balance.

Stock.txt: Stores stock record. [stock name, price(USD), company name]

Transaction.txt: Stores transaction record. [ID, UID, FromAcclID, ToAcclID, Date, Currency, Amount]

CheckingAccount.txt: Stores checking account record. [ID, UID, Currency, Balance, Data]

SavingAccount.txt: Stores saving account record. [ID, UID, Currency, Balance, Date]

LoanAccount.txt: Stores load account record. [ID, UID, Currency, Balance, Data, PaidLoan, UnpaidLoan]

SecuritiesAccount.txt: Stores securities account record. [ID, UID, Currency, Balance, Date]

StockList:

stores users' stock activity records. Each user owns a txt file named after userID.

- Model

ATM.java: ATM object.

Bank.java: Basic functions for the bank.

Accounts

AccountType.java: 4 types of accounts.

Account.java: Parent class for other Account type classes.

| **CheckingAccount.java:** Extend from account.java for checking account.

| **LoanAccount.java:** Extend from account.java for a loan account.

| **SavingAccount.java:** Extend from account.java for saving account.

| **SecuritiesAccount.java:** Extend from account.java for securities account.

AccountBehavior

IntrestBehavior.java: An interface for interest behavior.

for checking account. | **CheckingAccountIntrestBehavior.java:** Implement interface intrestBehavior

loan account. | **LoanAccountIntrestBehavior.java:** Implement interface intrestBehavior for

| **SavingAccountIntrestBehavior.java:** Implement interface intrestBehavior for saving account.

| **SecuritiesAccountIntrestBehavior.java:** Implement interface intrestBehavior for securities account.

Currency

CurrencyType.java: Three types of currency.

Money.java: Implements comparable<Money> for balance comparison and calculation.

Currency.java: Parent class for other currency type classes.

| **USD.java:** Extends currency class for USD.

| **CNY.java:** Extends currency class for CNY.

| **INR.java:** Extends currency class for INR.

Stock

Stock.java: Stock class, defines stock information. (Stock name, price, currency, company name)

StockList.java: Defines a list that stores information of a customer's own stock.

Transactions

TransactionType.java: 6 different transaction types.

Transaction.java: Defines transaction objects. Parent class of 6 transaction behavior class.

| **Transfer.java:** Extends from Transaction class for transfer transaction.

- | **Deposit.java:** Extends from Transaction class for deposit transaction.
- | **Withdraw.java:** Extends from Transaction class for withdrawing transaction.
- | **PayLoan.java:** Extends from Transaction class for loan paying transaction.
- | **StockBuy.java:** Extends from Transaction class for stock buying transaction.
- | **StockSell.java:** Extends from Transaction class for stock selling transaction.

Users

User.java: Defines all user objects. Parent class of customer and manager class.

| **Customer.java:** Extends from User.java. Contains methods that a customer can do.

| **Manager.java:** Extends from User.java. Contains methods that a manager can do. e.g. modify the stock list, update stock price, and get customers who have the most unpaid loans.

- Service

CreateAccount.java: Create saving/checking/securities account.

MakeTransactions: A user makes a transaction.

Register.java: User register behavior.

SingIn.java: User sign-in class.

- Util

Checker.java: Some checker methods.

Constants.java: Some constant helper values and methods.

Factory.java: Follow the factory pattern.

ID.java: methods related to ID issues.

Reader.java: Methods related to reading files and returning contents from files.

Writer.java: Methods related to writing or modify on files.

GUI Design

(GUI) Java Swing Components used for consistency throughout the bank:

swingComponents:

BackButton.java: the commonly used back button display component

Button.java: For consistent size,color and shape of JButtons

Frame.java: for common size, color, functionality of JFrames

Label.java: for common styling of JLabel

LogoutButton.java: the commonly used logout button display component

Panel.java: for common styling of JPanel

Frames:

Customer:

Account:

AccountFrame.java: Frame used to display all 3 account and a loan request

CreateAccount.java: To create a new account

CustomerDashboard.java: shows all of the users accounts

CustomerRegistration.java: registers a new customer to the bank

PersonallInfo.java: displays the personal information of the user

StartFrame.java: To select between a existing user and a new user

Manager

CustomerLookUp.java: To lookup specific user

ManagerDashboard.java: can update socks and can view transactions

Login.java: for logging into the user/manager's account

LogoFrame.java: The opening frame of the bank

UserType.java: (To select between user and manager

Manager and User Data:

Manager:(to change the data, refer to the constants file in the data folder)

username:"CPK"

password:1