Object-Oriented-Design Document

The program is divided into 6 layers: The **GUI** layer displaying the required information and guiding the user to input the actions. The **Controller** layer receieve the input information from GUI layer, forward the data to corresponding functions in **Service** layer, and return the result to the GUI for displaying. The Service layer is where every action's logic is really executed. Whenever the action requires data from database, the service layer will forward the request to the **Data Access Object** (DAO) layer, only in this layer the program can access to the database.

Using the action "log in" as an example. When user input the name, password and select the type of user, the GUI will forward these input to the `LoginController.signIn()` function in the Controller layer. This function will further forward the information to the `LoginService.signIn()` function in the service layer, where the log in action is really execute. When executing the action, we need to compare the username, password, and user type with our database to see whether there is a match, therefore, the `LoginService.signIn()` will call the `userDao.selectUserById()` function, where the program access the database serching for matching user. Each layer is returning whether log in or not to its upper layer, and if receieving response of success, the GUI will forward to the home page of user, otherwise the information doesn't match warning will be displayed.

Apart from the 5 layers above, the program also contains a model layer storing the defination of each object in the ATM system. The instances of these madels are created based on the data retrieved from the database in the DAO layer and return to the Service layer. The Service layer functions retrieve the required information from these instances and process these information based on the corresponding action's logic.

Each GUI file match with single business defined in model, which is pluggable for further increasement.

The detail description of each layers is as follows:

The GUI Layer in the Package "view":

1. GUIRegistry: The interface to register a new user.
2. GUILoginInWindow: The interface for user to log in.
3. GUICustomerHomePage: The homepage of customer, where customer can choose to managing user information, account information or taking actions related to money.
4. GUICustomerAccountWindow: The interface for customer to view, open and close their accounts. ?? need double check
5. GUICustomerCloseAccount: The interface for customer to close an account.
6. GUICustomerOpenAccount: The interface for customer to open a new account.
7. GUICustomerMoneyWindow: The interface for user to choose action related to money.
8. GUIUserProfileWindow: The interface to display customer profile.
9. GUIDailyReport: The interface for manager to choose which date report to check.
10. GUIDailyReportTable: The interface for specific date daily report.
11. GUIDeposit: The interface for customer to make deposit.
12. GUITransfer: The interface for customer to transfer monsy to another account.
13. GUIWithdraw: The interface for customer to withdraw money.
14. GUILoan: The interface for customer to take actions related to loan.
15. GUIStockManagement: The interface for manager to take actions related to stock.
16. GUIDisplayLoan: The interface displaying all the loans that customer have.
17. GUIRequireLoan: The interface for customer to require a loan.

18. GUIPayForLoan: The interface for customer to pay for loan.
19. GUIStock: The interface for user to take actions related to stock.
20. GUIDisplayStock: The interface displaying all the stocks that user hold or in the market depending on cases.
21. GUIBuyOrSellStock: The interface for customer having activated security account to buy or sell stocks.
22. GUITransactionHistory: The interface to display transactions.
23. GUITranscurrency: The interface for user to buy other currency.
24. GUIManagerHomePage: The interface for manager to choose action.
25. GUIManagerCheckCustomer: The interface for manager to check specific customer information.
26. GUIAddStock: The interface for manager to add stocks.
27. GUIUpdateStockPrice: The interface for manager to update stock price.
28. GUICustomerChecked: The interface to show checked customer.

The Controller Layer in the Package "controller":

1. LoginController: Handling all input when user log in.
2. AccountController: Handling all input when taking actions related to account management.
3. LoanController: Handling all input when taking actions related to loan.
4. StockController: Handling all input when taking actions related to stock.
5. TransactionController: Handling all input when taking actions related to transaction.

The Service Layer in the Package "service":

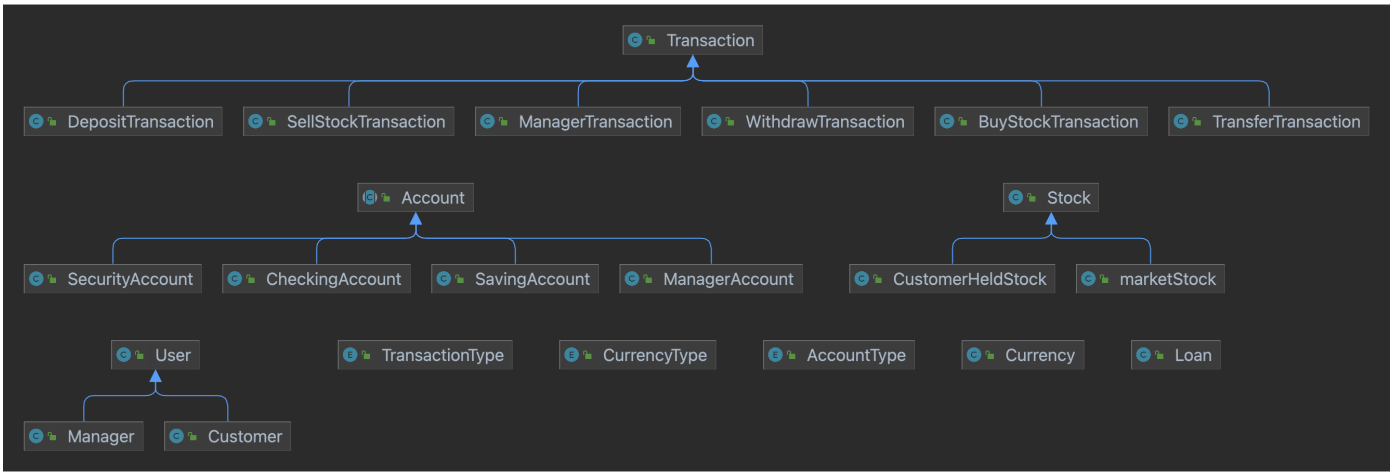
1. AccountService: The service layer to execute the actions related to account.
2. DbService: ?????
3. LoanService: The service layer to execute the actions related to loan.
4. LoginService: The service layer to execute the actions related to user log in.
5. StockService: The service layer to execute the actions related to stock.
6. TransactionService: The service layer to generate transactions.

The DAO Layer in the Package "dao":

1. AccountDao: The DAO layer access to database to retrieve information related to accounts.
2. ConnectDao: The MySQL database connection built-up setting, including connecting and closing functions.
3. CustomerHoldStocksDao: The DAO layer access to database to retrieve information related the the stocks hold by customer.
4. LoanDao: the DAO layer access to database to retrieve information about loans.
5. StockDao: the DAO layer access to database to retrieve information about loans.
6. TransactionDao: the DAO layer access to database to retrieve information about transactions.
7. UserDao: the DAO layer access to database to retrieve information about user information.

The Model Layer in the package "model":

UML:



1. User: The class contains the basic information of a user.
2. Customer: Extends from user, contains some additional information of customer.
3. Manager: Extends from user, contsins some additional information of manager.
4. CurrencyType: The enumration of type of currency.
5. Account: The abstract class contains attributes and information getters of account object.
6. AccountType: The enumration of type of accounts.
7. CheckingAccout: Extends from Account, contains some additional information of Checking Account.
8. SavingAccount: Extends from Account, contains some additional information of Saving Account.
9. SecurityAccount: Extends from Account, contains some additional information of Security Account.
10. ManagerAccount: Extends from Account, contains some additional information of Manager Account.
11. Stock: The class contains the basic information of a stock.
12. marketStock: Extends frm stock, contains additional information of stocks in stock market.
13. CustomerHoldStock: Extends from stock, contains additional information of stocks hold by customer.
14. Transaction: The class contains the basic information of a transaction.
15. TransactionType: The enumration of type of transaction.
16. TransferTransaction: Extends from Transaction, contains some additional information of transaction recording a transfer.
17. SellStockTransaction: Extends from Transaction, contains some additional information of transaction recording selling a stock.
18. BuyStockTransaction: Extends from Transaction, contains some additional information of transaction recording buying a stock.
19. DepositTransaction: Extends from Transaction, contains some additional information of transaction recording making deposit.
20. WithdrawTransaction: Extends from Transaction, contains some additional information of transaction recording withdrawing money.
21. Loan: The class contains the basic information of a loan.