

# CS 611 - Final Project Design Document

Team:

Shivangi(U35642613)

Piyush Kathuria (U01034912)

Taishan Chen(U92817364)

Zhengxiong Zouxu(U85329166)



Please refer to the attached png file in the zip (cs611\_classDiagram.png)

In this project, our team created a GUI based ATM application from both a manager's and a customer's perspective. The customer can create checking and saving accounts in 3 supported currencies for depositing his/her money, and also create a Securities account from their Savings account for trading in the stock market.

## **PROJECT STRUCTURE AND CLASS DESCRIPTIONS**

In this project, we used three design patterns :

1. Model View Controller design pattern to ensure abstraction and proper code structure.
2. Factory Design Pattern to generate accounts for users.
3. Singleton Design Pattern to create Bank Manager

You can also find the description of the class structures below

### **The Front End(GUI):**

**FrontMain:** This is the main class used to drive the application and instrument each class.

**PanelLogIn:** This class is the login interface. Administrators and users log into accounts by entering id and password.

**PanelLogUp:** This class is to display the create account interface. Users can enter a new user ID and password to create a new bank account.

**PanelCustomer:** This class refers to the user interface. It includes all functions available to users. For example, deposit, withdraw, transfer, check balance, etc.

**PanelCreateAccount:** This class mainly supports customers to create new accounts by themselves and display the current account information.

**PanelDepositAndWithdraw:** This class is mainly to generate deposit and withdrawal interfaces. Customers can enter this interface through Panel, and can also enter through the menu menu of other interfaces. This interface supports entering numbers to confirm the amount and currency of deposits and withdrawals.

**PanelLoan:** This class is to display a UI interface about loan, customers can apply for a loan by filling in the collateral value and loan purpose.

**PanelStockCustomer:** This class is to generate a stock trading interface for customers. Customers can view the current status of the account, buy stocks and sell them.

**PanelTransfer:** This class is a customer transfer interface. Customers can transfer money to other customers, as well as transfer money between their own accounts.

**CustomerDetailPanel.java :** Class to generate the customer detail interface for views of the accounts that customer has.

**CustomerTransPanel.java :** Class to generate a JPanel for customer transactions window

**CustomerTransWindow.java :** Class to generate the customer transactions window where the user can see all the transactions.

**FrontMain.java :** Main class to generate a login panel which serves as the initial panel for all other panels.

**ManagerAccountWindow.java :** Class to create the window to show all the accounts of a particular user to the manager.

**ManagerStockMarketPanel.java :** Class to generate the Stock market view for the Manager, displaying all information for top 30 stocks according to market capitalization.

**ManagerWindow.java :** Main class that acts as the first screen in the Manager View of the application. Contains links to all other screens that a manager can access. Ex - StockMarketPanel and AccountWindow.

**PanelCreateAccount.java :** Class to create the JPanel that lets users create accounts of all types.

**PanelCustomer.java :** Class to create the main panel for the customer where the customer can perform various functions like view accounts, view loans etc.

**PanelDepositAndWithdraw.java :** Class that creates a panel where the user can withdraw or deposit an amount in an account of his/her choosing.

**PanelLoan.java :** Class that creates a panel where the user can see the status and all information for his/her loans.

**PanelLogin.java :** Class that contains the code that creates the login panel for Customers.

**PanelLogUp.java :** Class that displays the screen that a user sees after login is successful. Shows user the various login

**PanelStockCustomer.java** - Class that creates the panel for customer stock window

**PanelTransfer.java** - Class that creates the panel where customer can see transfer option between accounts.

## **EXTERNAL FILES**

**Org.json.jar** - Jar file used to parse json object we are getting from Stock API.

## **The BackEnd (Code Logic) :**

**Account.java** - Abstract Base Class that contains methods and variables common to Saving, Checking and Securities classes.

**AccountFactory.java** - Class that uses the Factory design pattern to generate the Account classes for the user.

**SavingsAccount.java** - Class that specifies all the logic for creating the Savings account. It also lets the user create a securities account.

**CheckingAccount.java** - Class that specifies all the logic for creating the Checking account for a customer.

**SecuritiesAccount.java** - Class that specifies all the logic for creating the Securities account for a customer.

**DataController.java** - A controller class that contains all the methods which allows other objects in our code to perform operations on our XML database.

**BackendController.java** - A controller class that contains all the methods which details all the interactions between objects of different classes.

**BankHuman.java** - Abstract base class that contains methods and variables common to BankManager and Customer classes.

**BankManager.java** - Child class of BankHuman that contains the methods and attributes relevant to a bank manager of an ATM.

**Customer.java** - Child class of BankHuman that contains methods and attributes relevant to the Customer of an ATM.

**Currency.java** - An Enum which contains all currencies currently supported by our program.

**CustomerAction.java** - Abstract base class that contains all the methods and attributes for all actions a customer can take.

**Loan.java** - Child class of CustomerAction class which contains logic to create a loan for the customer with an associated account.

**StockController.java** - Class that controls how stocks will interact with other classes

**StockData.java** - Class that retrieves Stock Data from API

**Trade.java** - Class that contains all logic related to trading and buying/selling of stocks.

**Transaction.java** - Class that contains all logic related to user transactions

**TransactionPanel.java** - Class that shows transactions done by the customer.