

# Design Doc Of Bank ATM

Qi Yin U31787103

For each class, I listed 3 essential points of my design:

1. purpose of the class
2. benefit of this class to the existing design
3. benefit of this class, if any to a future game

## Controller

This package contains the back end classes of the project.

---

### *BankATM*

1. This is the entrance of the project.
2. It helps set an entrance to start the project.
3. In the future, it can be changed the name and be the entrance of another project.

---

### *SystemInterface*

1. This interface defines functions that a system should have. Here it means the functions that user and manager have in common.
2. It specifies the methods that the system must implement.
3. In the future, other system can be implemented which must register, login and logout.  
Some other methods can be added if need.

---

### *BankController*

1. This is the controller for bank or manager. It contains the methods that bank system used.
2. It deals the operations that front end called in bank system.
3. This is designed for this bank ATM specifically. But some methods can be used in other manage systems.

---

### *UserController*

1. This is the controller for user. It contains the methods that user system used.
2. It deals the operations that front end called in user system.
3. This is designed for this bank ATM specifically. But some methods can be used in other user systems.

# Model

This package contains the data structure classes of the project.

---

## *Account*

1. This class defines an account.
  2. It records an account for a user and contains the elements an account has.
  3. In the future, it can be used anywhere there are accounts.
- 

## *Bank*

1. This class defines a bank.
  2. It contains the elements a bank has.
  3. In the future, it can be used by other bank systems.
- 

## *Currency*

1. This class defines a currency.
  2. It contains the elements a currency has.
  3. In the future, it can be used anywhere there are currencies.
- 

## *CurrencyConfig*

1. This class defines the configurations of a currency.
  2. The configuration can be changed by manager and helps the manager to accumulate money.
  3. In the future, it can be used anywhere there are currencies.
- 

## *DailyReport*

1. This class defines the daily report.
  2. This shows the daily report and helps the manager to manage the bank well.
  3. In the future, it can be used anywhere there are transactions and wants to show the details of the report.
- 

## *Date*

1. This class defines the date.
2. It contains the elements date has.
3. In the future, it can be used anywhere date exists.

---

## *Loan*

1. This class defines loan.
2. It contains the elements a loan has.
3. In the future, it can be used anywhere loans are needed.

---

## *Name*

1. This class defines the name of a person.
2. It contains the elements a name has.
3. In the future, it can be used anywhere names are needed.

---

## *Person*

1. This class defines a person.
2. It contains the elements a person has.
3. In the future, it can be used anywhere there is a person.

---

## *PhoneNumber*

1. This class defines a phone number.
2. It contains the elements a phone number has.
3. In the future, it can be used anywhere phone numbers are needed.

---

## *Transaction*

1. This class defines a transaction.
2. It contains the elements a transaction has.
3. In the future, it can be used anywhere there are transactions.

---

## *User*

1. This class defines a user and extends from person.
2. It contains the elements a user in bank has.
3. In the future, it can be used in other bank systems.

## **Utils**

This package contains some global functions and variables.

---

## *Config*

1. This class defines some global variables, constants, default configurations.
2. It prevents specific number from appearing in the project and instead replaces them with variables for easily maintenance.
3. The variables and constants are defined for this project.

---

## *ErrCode*

1. This class defines the error code for the project.
2. It can easily show the error message to users.
3. The variables and constants are defined for this project.

---

## *UtilFunction*

1. This class defines some utility methods that used in the whole project.
2. The same as 1.
3. The methods are defined for this project. But some of the methods can be reused in other projects.

## **View**

This package contains all the views for this project.

---

## *AccountDetail*

1. This class draws the account detail page. It is used by user system. It shows the details of the accounts for a user.
2. The same as 1.
3. The page is designed for this project.

---

## *CheckBalance*

1. This class draws the check balance page. It is used by manager system. It shows the details of the balance in bank.
2. The same as 1.
3. The page is designed for this project.

---

## *CheckCustomer*

1. This class draws the check customer page. It is used by manager system. It shows all users who registered for the system.
2. The same as 1.
3. The page is designed for this project.

---

## *ChooseIdentity*

1. This class draws the choose identity page. It is the home page of the project. People should choose their identity here.
2. The same as 1.
3. The page is designed for this project.

---

## *CurrencyConfig*

1. This class draws the currency configuration page. It is used by manager system. Manager can set currency configuration at the page.
2. The same as 1.
3. The page can be reused anywhere currency configurations need to be set.

---

## *GetDailyReport*

1. This class draws the daily report page. It is used by manager system. It shows the details of a day's transaction details in the bank.
2. The same as 1.
3. The page is designed for this project.

---

## *Login*

1. This class draws the login page. It is used by both user system and manager system. It is used to login to the system.
2. The same as 1.
3. It can be used anywhere login is needed.

---

## *ManagerInterface*

1. This class draws the manager system page. It is the home page of manager system. It is used by manager system. Manager can take actions at this page.
2. The same as 1.
3. The page is designed for this project.

---

## *Register*

1. This class draws the registration page. It is used by user system. Users should register for the system first.
2. The same as 1.
3. It can be reused any where registration is needed.

---

## *SetConfig*

1. This class draws the set configuration page. It is used by manager system. Manager can set all configurations here.
2. The same as 1.
3. The page is designed for this project.

---

### *TakeLoan*

1. This class draws the take loan page. It is used by user system. User can take a loan here.
2. The same as 1.
3. The page is designed for this project.

---

### *Transact*

1. This class draws the transaction page. It is used by user system. User can do transactions such as deposit, withdraw, transfer here.
2. The same as 1.
3. The page is designed for this project.

---

### *TransactionDetail*

1. This class draws the transaction detail page. It is used by both manager system and user system. It shows the details of a transaction for a user.
2. The same as 1.
3. The page is designed for this project.

---

### *UserDetail*

1. This class draws the user detail page. It is used by manager system. It shows all the details of a user.
2. The same as 1.
3. The page is designed for this project.

---

### *UserInterface*

1. This class draws the user system page. It is used by user system and is the home page for user system. Users can take actions here.
2. The same as 1.
3. The page is designed for this project.