Fancy ATM Design Documentation

Yufan Wen

U87145681

evanswen@bu.edu

Index

[Fancy ATM Design Documentation 1](#_Toc22335541)

[Introduction 3](#_Toc22335542)

[Object Oriented Design Specification 3](#_Toc22335543)

[Account 3](#_Toc22335544)

[CheckingAccount & SavingAccount 3](#_Toc22335545)

[Currency 4](#_Toc22335546)

[Loan 4](#_Toc22335547)

[Transaction 5](#_Toc22335548)

[Date 5](#_Toc22335549)

[guiATM 5](#_Toc22335550)

[guiLogin 6](#_Toc22335551)

[guiManager 6](#_Toc22335552)

[guiQuery 6](#_Toc22335553)

[Report 6](#_Toc22335554)

[ATMShell 6](#_Toc22335555)

[Examples 6](#_Toc22335556)

# Introduction

In this assignment, we are asked to implement a ATM terminal with design of Object Oriented Design and JAVA, in this assignment, I implemented function as transfer, deposit, create accounts, make loans and etc. Customer can check his/her own transaction records, and manager can view transaction records of all customers, and he can also check transaction filtered by date or customers.

# Object Oriented Design Specification

## Account

Meaning:

This abstract class extract common activities of an account, which contains different methods of an account:

Attributes:

String password;

String userName;

public Currency balance;

public boolean hasLoan;

public Loan loan;

public List<Transaction> transactions;

Methods:

Constructor():

public String getPassword(): get password of this account

public boolean setLoan(): check if this account has an loan from bank

public boolean payLoan(): method for account to pay his/her loan, if payment available, return true, else, return false

## CheckingAccount & SavingAccount

Meaning:

For different account type in this ATM, the activity is the same, so just put it in different classes, if there will be different activity for different classes, we can add them into these classes in order to expand function.

## Currency

Meaning:

This class defines different currency related activities including adding, delining, changing different currency types and change output to String type.

Attributes:

public Map<String,Double> rate = new HashMap<>();

private String Cur\_Cur;(means Current Currency)

private Double lastCharge;

private Double value;

Methods:

Constructor(): store different currency types and its rate in a map, so we can change Currency according to currency types.

public boolean setCurrency(): set different currency types for current account.

public boolean addValue(): add a number of money to current account

public boolean declineValue: decline a number of money from current account

public Currency mergeCur: while there are different currency types, call this method to merge them into main currency type the account is using.

public String toString()

public String getCur\_Cur()

public Double getLastCharge()

public Double getValue()

## Loan

Meaning:

To store the loan record, and it will present in guiQuery, this class will help calculating the interests of a loan. Loan is calculated by the term, whose unit is month.

Attributes:

public String userName;

public Currency currency;

public Date startDate;

public int termMonth;

public Currency interest;

public boolean isPaid;

Methods:

Constructor():

public Double calculateInterest(): calculate intrest the Bank might get from this loan.

## Transaction

Meaning:

This class stores information of current transaction performed by current account.

Attributes:

public Date date;

public String fromUser;

public String toUser;

public Currency value;

public Currency charge;

public Currency balanceLeft;

public String description;

Methods:

Constructor()

## Date

Meaning:

This class generate today’s date, for presentation in JTable.

Attributes:

private int day;

private int month;

private int year;

Methods:

public String toString(): change date parameters to Strings for the convenience of printing in table.

public int getDay()

public int getMonth()

public int getYear()

## guiATM

Meaning:

Checking if account has login, if login, create a guiQuery, if not, then create a login user interface.

## guiLogin

Meaning:

This GUI class present user interface for login and create account, with the help of JTextField, JPasswordField, JButton.

## guiManager

Meaning:

This GUI class is for manager to examine transaction records of different users, he can directly enter this user interface because he has the highest authorization.

## guiQuery

Meaning:

This is GUI class for current account to complete operation as transaction, deposit, withdraw, changing currency, and etc. whenever current user want to make some operation, it will open anther information window for user to input information required, and once user finished operation, it will provide feedback as information window.

## Report

Meaning:

This GUI class provide information of account’s activities, user can only see his/her own different transactions, but manager can view all users’ activities.

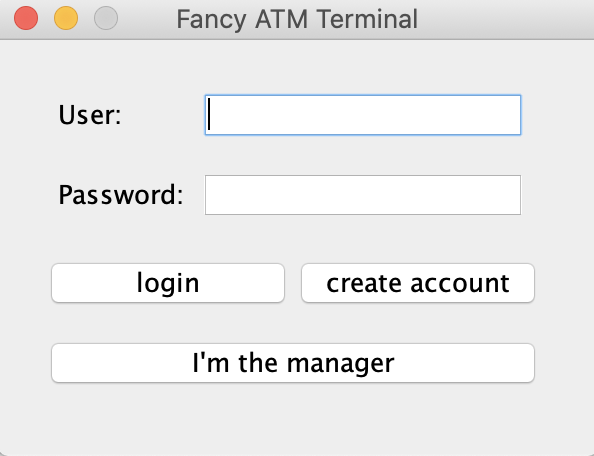
## ATMShell

Meaning:

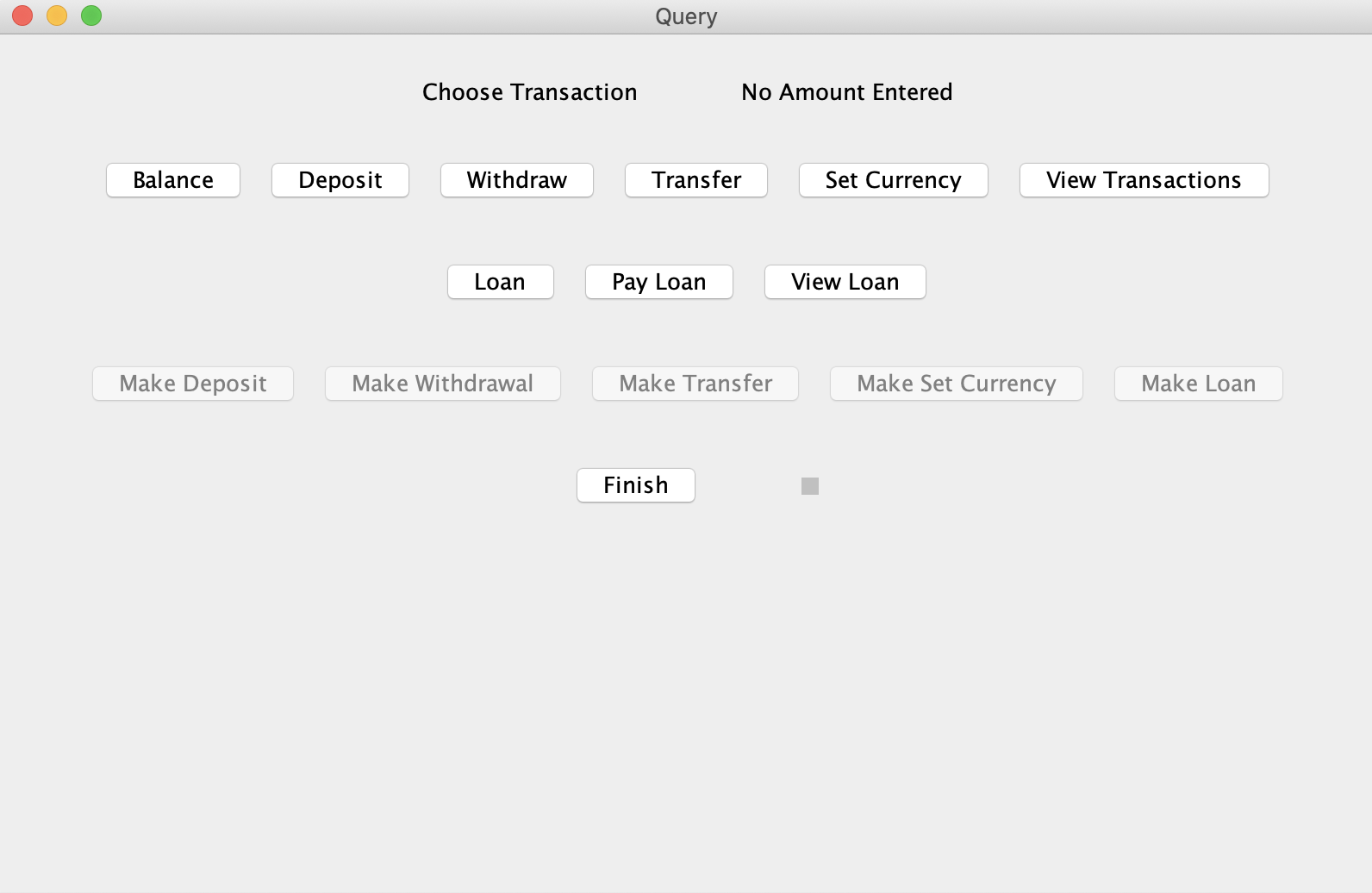
The main class that initialize the whole project.

# Examples

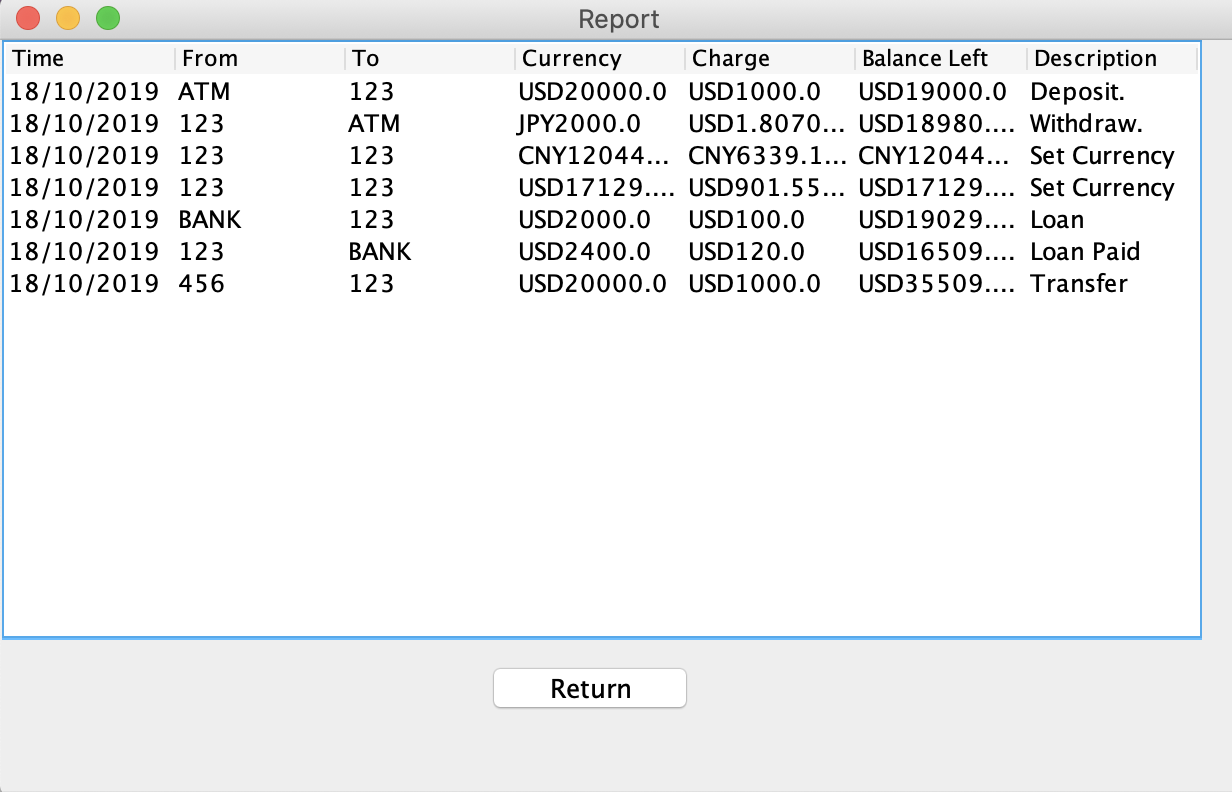
ATM terminal:



Query interface:



Personal report



Manager report:

