I will analysis the classes with 3 respects:

(1) purpose of the class

(2) benefit of this class to our existing design

(3) benefit of this class, if any to a future bank (not for every class)

User:

(1) The User class has 3 attributes which are name, email, password. User class is the super class of Customer and Manager
(2) Both customer and manager have name and they need email (also known as username in my FancyBank) and password to login the FancyBank. we can make code of Customer and Manager class simpler by inheriting User class.
(3) Maybe in other bank system, there is a position whose priority is higher than manager, it is easy for us to create such position by inheriting User class and add specific attributes that only available to this position.

Manager:

(1) Except name, email and password, Manager has a transaction daily report attributes that is used to store the recording of transactions made by customers for that day.
(2) Manager will be initialed by FancyBank class and delivered through whole process of this application. Once there is a transaction happening, system can capture manager variable and update the transactionDailyReport attributes immediately and conveniently.

Customer:

(1) Except name, email and password, Customer has many lists such as accounts which can contain many accounts like checking account, saving account and loan account, transactions that record the history of transactions of customer itself, collateralsOwn that contains the collaterals that can be used to loan money for customer and collateralsLoan that contains the collaterals that have been used for loan and expected to be pay back.
(2) The reason why I use list is the number of accounts, transactions, collaterals are not stable. The add and remove function in list provide me a good solution

to this problem. Customer can easily create an account and transactions history can be easily added to the customer.

Account:

(1) An account includes account number, account type(checking account, saving account, loan account), interest rate(0 for checking account, 0.05 for saving account, 0.1 for loan account) and currency list(RMB, dollar, pound).
(2) The differences between accounts are the type and the interest rate correspond to the type. Because there is no other attribute should be added to a specific account, so there is no need to regard Account class as a super class and make other accounts inherit Account class. By assign different values to type and interest rate, we can easily create different accounts.
(3) We can increase the number of type of account for other banks by assigning a different integer to type attribute and creating functions that work for this specific type of account.

Currency:

(1) Currency class plays as a role of balance. It contains type and balance. Type is to identify this currency is RMB, dollar or pound and so on. Balance represents the balance of this specific type of currency.
(2) When a list of currency is added to the account of customer, customer will have an account with different type of currencies. Customer can do Save, Withdraw and Transact operations based on the currency.
(3) There are only 3 types of currencies in my FancyBank. But it is convenient to create another type of currency like HKD by “new Currency(“HKD”, 0)”for other banks.

Collateral:

(1) Collateral includes label(also known as name), origin value and new value. Collaterals are part of customer, customer can use collaterals to get loan.
(2) System will evaluate the value of collateral which is origin value, that’s the money we can loan by depositing this collateral. But when manager starts to charge interest, the amount you need to pay back this collateral will rise, that

becomes the new value.