

Possible examples include:

Purchase ordering system\*

Banking application\*

Collectable card game tracker\*

Employee management system

Banking App:

Driver:

getUser()

* First of all: new user or existing user?
  + If new user: register account, role, password, confirm pwd, balance, then proceed to login page
  + Existing: login()
* Login():
  + Input userID
    - Check if user exists,
      * If not, asks user to enter another one.
      * Existing user: Input password
        + Check pwd
        + If matches. Return user.
        + Else: exit program. / asks user to enter id again.

UserUI(user)

Shows UI for user role determined.

User can then: deposit, withdraw, getAccountInfo(), transaction log.

Clerk can check transaction log, ???

Admin: do everything the others can do, plus more. Like assign clerk role to customers. Or make clerks customers again.

FULL backlog:

User, admin, bank clerk.

User: userId, balance, deposit, withdraw, check account info, transaction log

deposit(): user money--, account money++, transaction log: add new log

transactionLog: obj contains userId, clerk id, date&time, amount of money in or out, comment.

withdraw(): user money++, account money--, transactionlog add new withdraw log.

checkInfo(userId, account number): get balance, list transactionlog.

Clerk: clerkId, getAccountInfo(userId, account number), createsAccounts and CancelAccounts

admin: have all the functions, plus

Purchase system backlog:

list all available products with product id with prices

purchase with id and $account, seller get money from customer

product: product id, name, price, stock.

buyer: id, money, role: buyer

buyer can spend money to change stock, and seller's money

purchase(itemId, quantity of purchase):

buyer money - item price \* quantity of purchase; seller money + item price \* quantity of purchase; stock - quantity of purchase;

print receipt: seller money amount, quantity of purchase, current stock, money spent, tracking number, transactionID

if buyer not enough money: throw exception/catch and fail purchase.

if out of stock: throw out of stock message.

seller: id, money, role: seller

seller can change item price, stock, name, id

can add new item, remove existing item.

admin: can change everything.

given transaction ID can find a purchase, which contains

transaction: contains list of itemids purchased, their quantity, money spent and tracking number, as well as buyer id seller id.