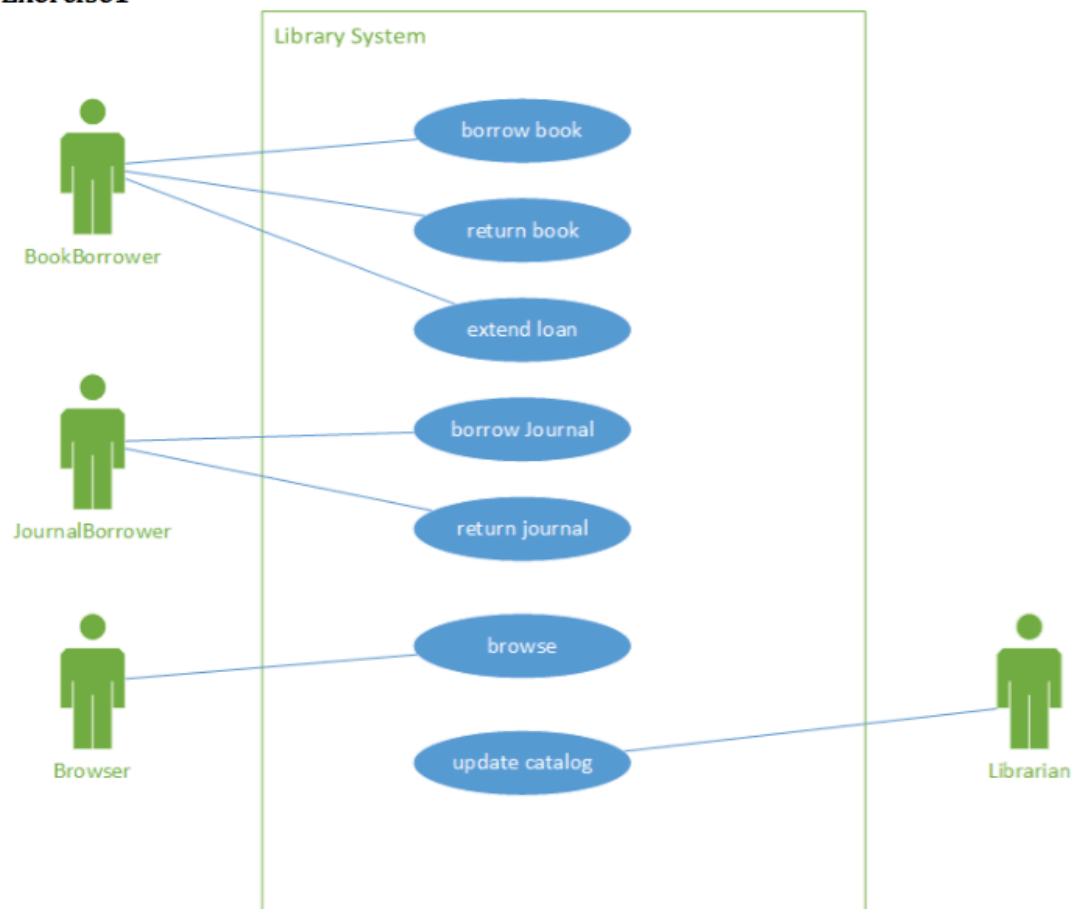
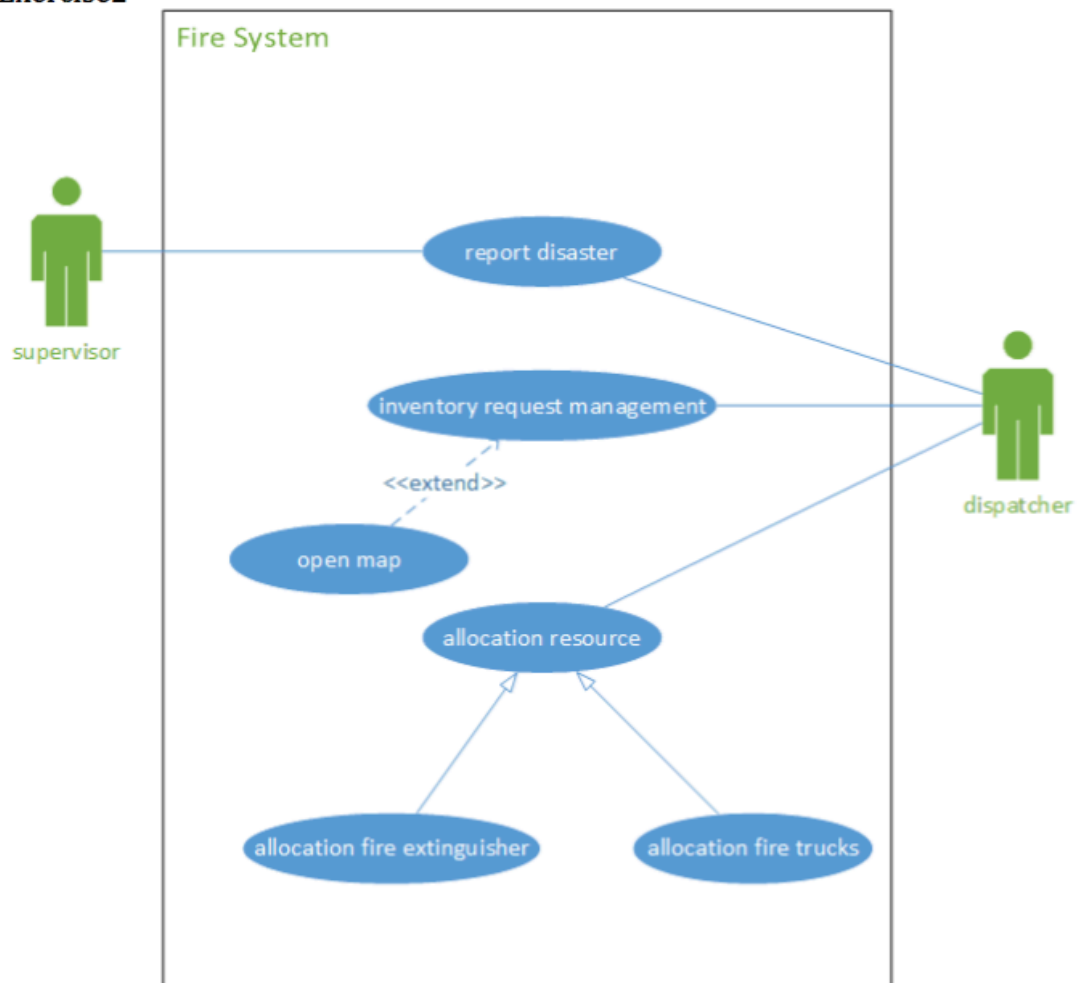
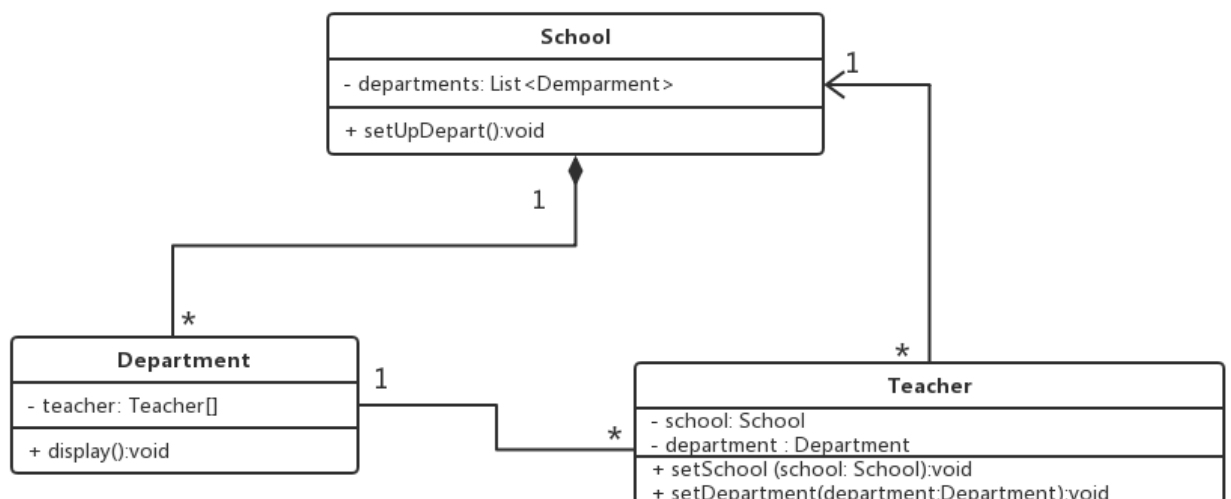


Exercise1

Exercise2



Exercise3



Exercise4

```
public interface graph {
    void draw();
}

public class polygon implements graph {
    private double perimeter;
    private double area;

    @Override
    public void draw() {}
}

public class triangle extends polygon{
    public double side1;
    public double side2;
    public double side3;

    public void draw(){}
}

public class rectangle extends triangle {
    public double side1;
    public double side2;
    public double side3;
    public double side4;

    public void draw(){}
}
```

WeChar Payment

Mobile payment is very popular in China. Many people use WeChat as their third-party payment platform. The following description is a simple design of its payment function.

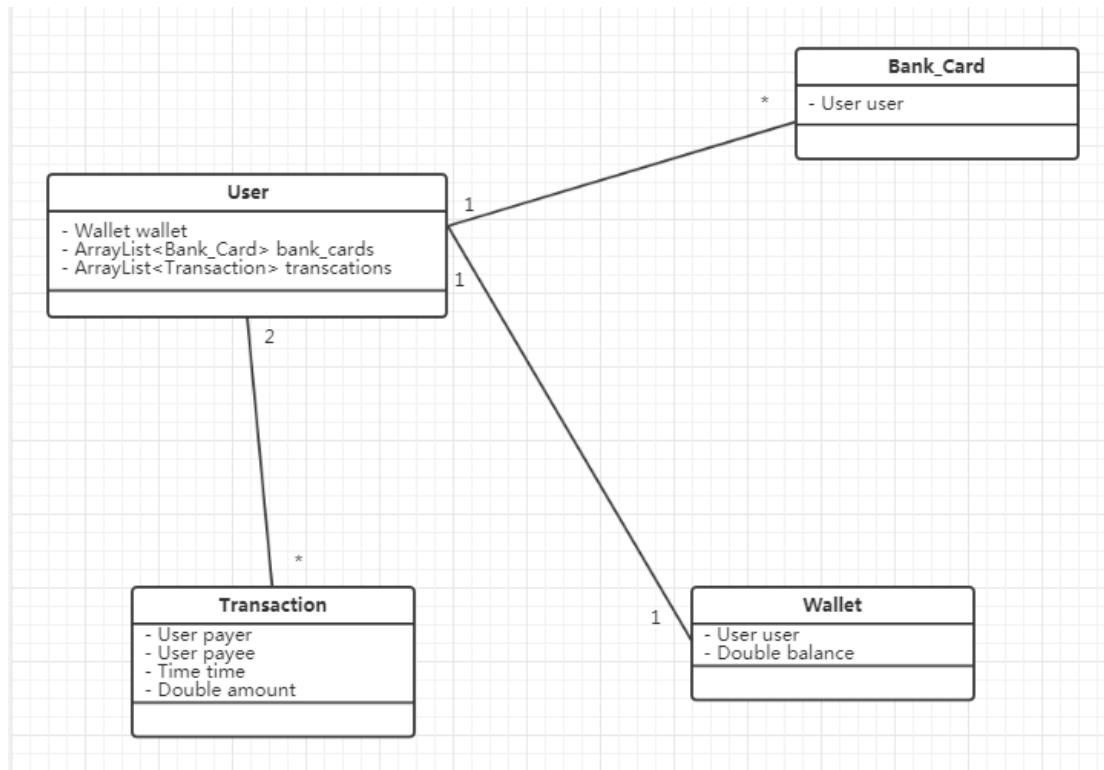
You can do multiple operations in WeChat pay, including checking how much balance you have in your wallet, adding bank cards to recharge or withdraw, looking up transaction history, sending lucky money to your contacts, and making payment to strangers by a QR code. In last case, we have two ways of transaction, either the payer scanning the QR code of payee's or the payee scanning the QR code of payer's.

Question 1: Draw a use case diagram according to the above scenario. [30 points]

Answer 1:



Question 2: Class diagram: Finding out entity class according to your design, and given the class diagram. In this section, you can only provide the class name, necessary attributes, and the indicate the relationship between those classes. [20 points]



Question 3: Please given the code structure according to your class diagram. [10 points]

Answer 3:

```

Public class User {
    Private Wallet wallet;
    Private ArrayList<BankCard> bankcard;
    Private ArrayList<Transaction> transactions;
}

Public class BankCard {
    Private User user;
}

Public class Wallet {
    Private User user;
    Private Double balance;
}

Public class Transaction {
    Private User payer;
    Private User payee;
    Time time;
    Double amount;
}
    
```