Receipt application 🧾

In this project I made a Java application that creates,deletes,print and save receipts for future usage i.e returning an item to a shop,replaceing an item.

Features 🚀

Multi threaded Server 🖧

I made a class called Server to handle the saving operation so every receipt can be saved and restored from the server by inserting the receipt id. The server class is multi threaded so it can serve as many cashiers as possible, but each receipt can be accessed by only one cashier at a time for security reasons
I used semaphores to make that happen.

Saving/Restoring 📝

In the Receipt class there is two static methods Save/Load. Save to save every Receipt object in an external folder, Load is used to load every object that has been saved by the Save method.
I used ObjectOutputStream, ObjectInputStream and the Serializable interface to save and load these objects .

Printed Receipts

each receipt will be printed with the necessary data. i.e date and time, shop name,receipt id,each and every item that have bought and total price.
this is a sample of a printed receipt

```
12/16/2021  11:34 AM
Sample
@Sa1639643664013
+----------------+------+----------+------------+
| NAME           | ID   | Quantity | Price      |
+----------------+------+----------+------------+
| Banana         | B6A  | 4        | 9.32       |
| Orange         | O6E  | 8        | 24.00      |
| Coke           | C4E  | 4        | 8.00       |
+----------------+------+----------+------------+
Total = 41.32
See you next time!
```

Unique id ☑️

Each receipt will have it's own unique id, this id consists of first two letters of the shop name and the number of milliseconds since January 1,1970.

Code ☕

in this part I will be talking about the program code.

Semaphore ⛔

An array of semaphores will be initialised with the size of (number of Receipts in the folder), everytime a user request to modify a receipt it will go to the index of the file in the semaphore array and the semaphore will be initilised with number 1 "Binary semaphore"

```
    //Deleting a receipt
    boolean found = false;
    while(!found) {
        id = dis.readUTF()+".txt";
        for (int i = 0; i < files.length; i++) {
            if (id.equals(files[i].getName())) {
                found = true;
                break;
            }
        }
        dos.writeBoolean(found);
    }
    int spot = findSpot(id);
    if(sem[spot]==null){sem[spot]=new Semaphore( permits: 1);}
    sem[spot].acquire();
    Receipt.Delete(Receipt.Load(id.substring(0,id.length()-4)));
    sem[spot].release();
    RefreshFiles();
```

Server 🖧

Like I said before the server is multi threaded so it's going to reuse the same address more than once and then run the class MT in different thread.

```
        ServerSocket ss = new ServerSocket( port: 1234);
        File folder = new File( pathname: "Receipts");
        if(!folder.exists()){ folder.mkdir();}
        ss.setReuseAddress(true); //Enable Multithreading
        while (true){ //as long as Server ready it will accept Clients
            Socket s = ss.accept();
            MT mt = new MT(s);
            Thread thread = new Thread(mt);
            thread.start();

        }
    }
```

Find A Receipt : 🔍

the cashier will input the receipt id, then the receipt id will be sent to the server and wait for boolean response, if it's true it will load the object then send it to the cashier + the boolean statement. If the response is false, it will only send the boolean statement and the cashier will print a message "invalid Receipt ID"
and it will ask you to input the receipt id once again or exit.

```
public static void FindAReceipt(DataInputStream dis,DataOutputStream dos,ObjectInputStream ois,ObjectOutputStream oos) throws Exception{
Scanner scan = new Scanner(System.in);
    String id;
    while (true) {
        System.out.print("Please Enter the Receipt ID or type 'exit':");
        id = scan.nextLine();
        if(id.equals("exit")){return;}
        dos.writeUTF(id);
        dos.flush();
        if(dis.readBoolean()){System.out.println("Connecting...."); break; }
        else System.out.println("Invalid Receipt ID");
    }

    Receipt r = (Receipt) ois.readObject();
    System.out.println("Connected");
    ReceiptOperations(r,oos,dos);
    oos.writeObject(r);
}
```

Create A New Receipt 🛒

Cashier will create a new object of Receipt and grant the cashier the ability to add items , delete items and print the receipt. Then send the object to the server to save later on and update the semaphore list.

```
public static void CreateAReceipt(DataInputStream dis,DataOutputStream dos,ObjectOutputStream oos)throws Exception{
Scanner scan = new Scanner(System.in);
String ShopName;
while(true){
    System.out.print("Please Enter the shop name:");
    ShopName = scan.nextLine();
    while(ShopName.length()<2){
    System.out.println("Invalid name");
    System.out.print("Please Enter the shop name:");
    ShopName = scan.nextLine();
    }


    Pattern p = Pattern.compile( regex: "^[a-z ]+$",Pattern.CASE_INSENSITIVE);
    Matcher m = p.matcher(ShopName);
    if(m.find()){break;}//if string contains only letters I leave loop else i will print invalid name and I will ask you again
    else{ System.out.println("invalid name");}
}
Receipt r = new Receipt(ShopName);
ReceiptOperations(r,oos,dos);
oos.writeObject(r);
}
```

DeleteAReceipt 🗑️

Cashier will input the receipt id and send it to the server if the receipt exists it gonna delete it and return true, if the server couldn't find the receipt it will return false and it will asks you to enter the receipt id or exit.