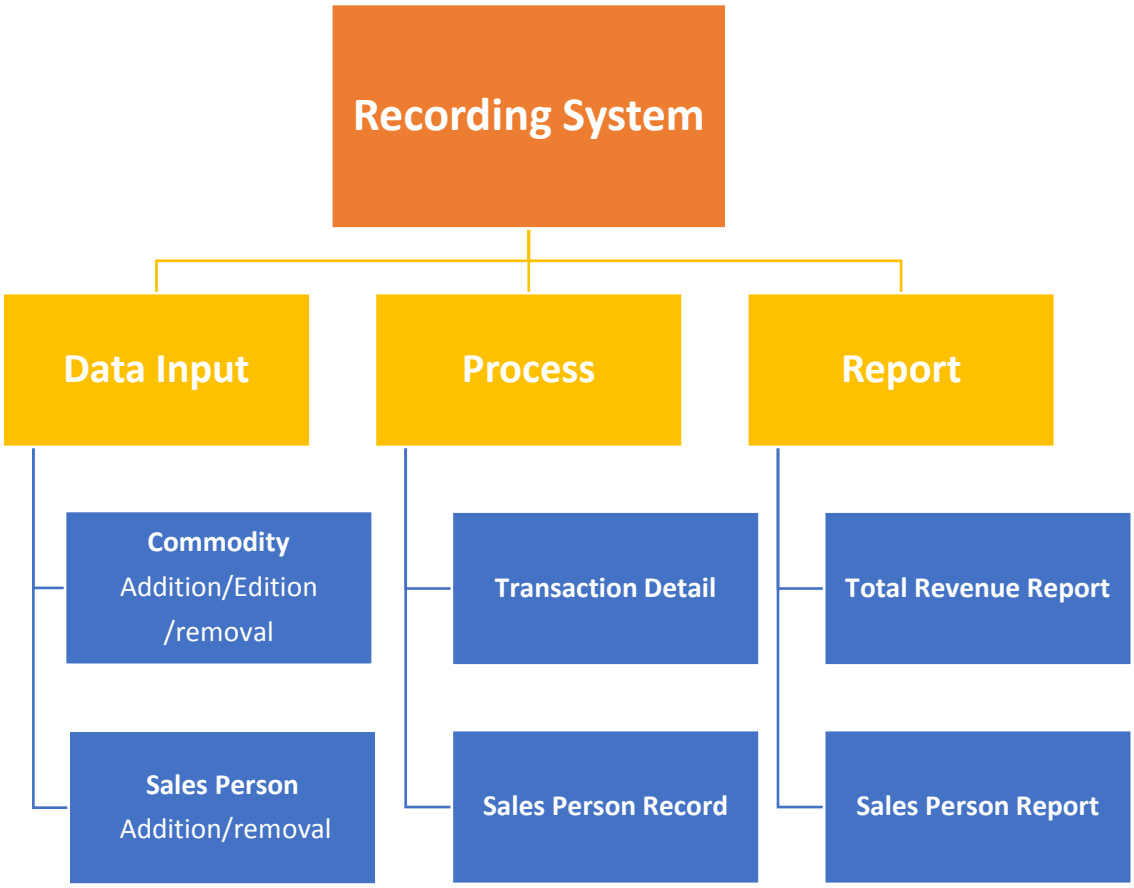
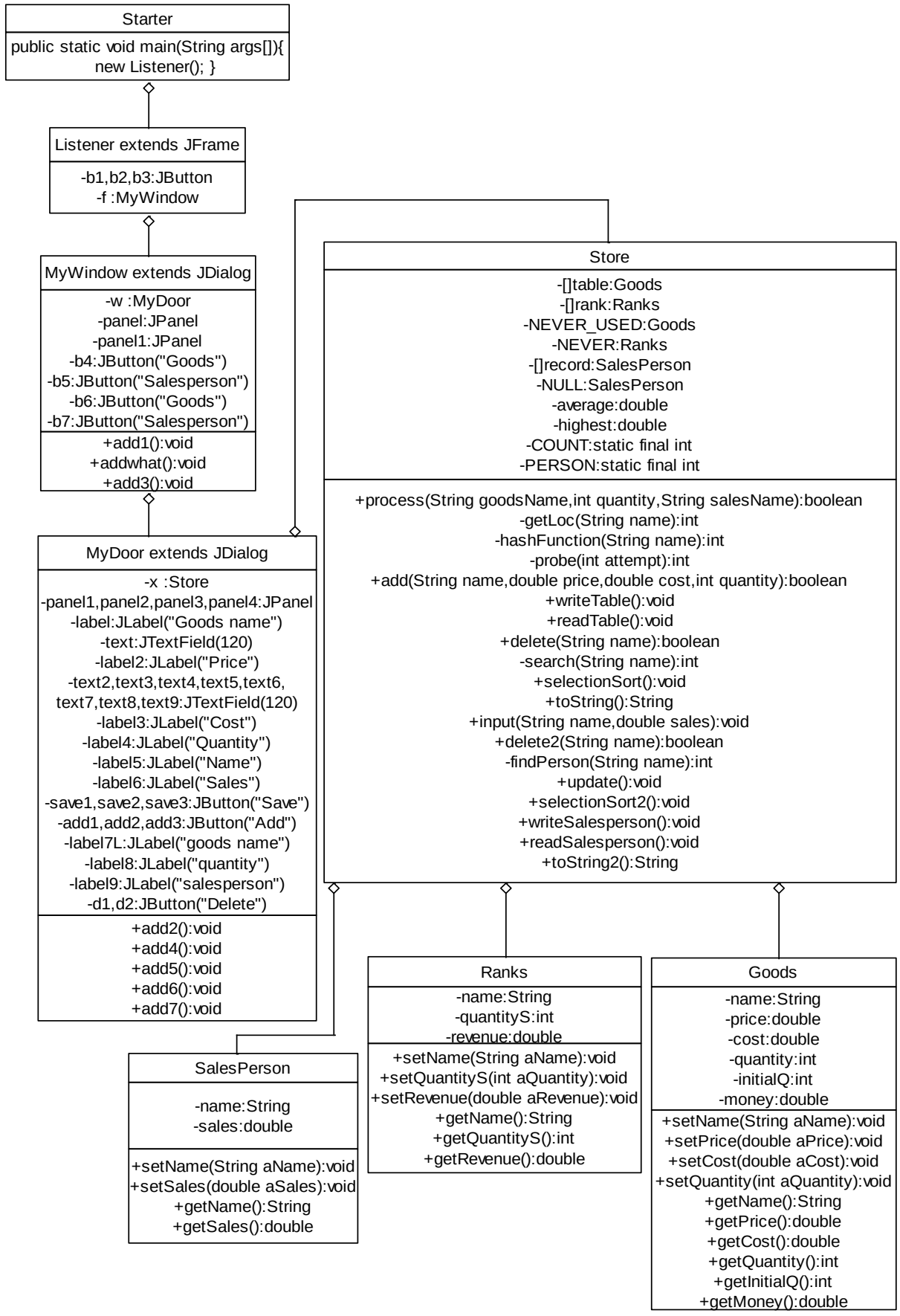
Criterion B: Design

Structure diagram



UML


Starter
public static void main(String args[]){
new Listener(); }
Listener extends JFrame
-b1,b2,b3:JButton
-f :MyWindow
MyWindow extends JDialog
-w :MyDoor
-panel:JPanel
-panel1:JPanel
-b4:JButton("Goods")
-b5:JButton("Salesperson")
-b6:JButton("Goods")
-b7:JButton("Salesperson")
+add1():void
+addwhat():void
+add3():void
Store
-[]table:Goods
-[]rank:Ranks
-NEVER_USED:Goods
-NEVER:Ranks
-[]record:SalesPerson
-NULL:SalesPerson
-average:double
-highest:double
-COUNT:static final int
-PERSON:static final int
+process(String goodsName,int quantity,String salesName):boolean
-getLoc(String name):int
-hashFunction(String name):int
-probe(int attempt):int
+add(String name,double price,double cost,int quantity):boolean
+writeTable():void
+readTable():void
+delete(String name):boolean
-search(String name):int
+selectionSort():void
+toString():String
+input(String name,double sales):void
+delete2(String name):boolean
-findPerson(String name):int
+update():void
+selectionSort2():void
+writeSalesperson():void
+readSalesperson():void
+toString2():String
MyDoor extends JDialog
-x :Store
-panel1,panel2,panel3,panel4:JPanel
-label:JLabel("Goods name")
-text:JTextField(120)
-label2:JLabel("Price")
-text2,text3,text4,text5,text6,
text7,text8,text9:JTextField(120)
-label3:JLabel("Cost")
-label4:JLabel("Quantity")
-label5:JLabel("Name")
-label6:JLabel("Sales")
-save1,save2,save3:JButton("Save")
-add1,add2,add3:JButton("Add")
-label7L:JLabel("goods name")
-label8:JLabel("quantity")
-label9:JLabel("salesperson")
-d1,d2:JButton("Delete")
+add2():void
+add4():void
+add5():void
+add6():void
+add7():void
Ranks
-name:String
-quantityS:int
-revenue:double
+setName(String aName):void
+setQuantityS(int aQuantity):void
+setRevenue(double aRevenue):void
+getName():String
+getQuantityS():int
+getRevenue():double
Goods
-name:String
-price:double
-cost:double
-quantity:int
-initialQ:int
-money:double
+setName(String aName):void
+setPrice(double aPrice):void
+setCost(double aCost):void
+setQuantity(int aQuantity):void
+getName():String
+getPrice():double
+getCost():double
+getQuantity():int
+getInitialQ():int
+getMoney():double
SalesPerson
-name:String
-sales:double
+setName(String aName):void
+setSales(double aSales):void
+getName():String
+getSales():double

Tests required

Test Type	Nature of test	Example
Add a product into the system	-Information of the product integrate into a Goods record -Find the key of this Goods record, and, according to it, add the record into the hash table -Write the updated hash table into the data file for Goods	-The Goods record: Goods name:“Chips”, Price:“2.50”, Cost:“1.00”, Quantity: “100” -“input”—“Goods”—“Add” Its key is 3, so this record is added in the 3rd place of the hash table -Click the “Save” button. The updated hash table is written into the data file
Delete a product from the system	-Locate and delete the Goods record of the product in the hash table -Write the updated hash table into the data file for Goods	-To delete “Chips”, “input”—“Goods”—“Delete”. According to its key of 3, delete the record in the 3rd place of the hash table -Click the “Save” button. The updated hash table is written into the data file
Add a salesperson into the system	-Information of the salesperson integrate into a SalesPerson record -Add the record into the array -Write the updated array into the data file for SalesPerson	-The SalesPerson record: Name: “Dave”, Sales: “54.7” -“input”—“Salesperson”—“Add”, and add this record into the array -Click the “Save” button. The updated array is written into the data file
Delete a salesperson from the system	-Use sequential search to find the record in the array and delete it -Write the updated array into the data file for SalesPerson	-To delete salesperson “Dave”, “input”—“Salesperson”—“Delete”, find and delete the record -Click the “Save” button. The updated array is written into the data file
Execute transaction	-Check if the input Goods is valid. If yes, locate it -Check if the saled quantity exceeds its original quantity. Yes, “sold out!”; or, do calculation -Check input SalesPerson. If valid, add sales to the record	-Click “transaction” button, and input “apple”-“3”-“Mike” -The Goods record “apple” is in the system -Remained quantity of “apple” is 10, greater than 3, Therefore calculate the profit=2x3=6, cost=1x3=3. Reminded quantity falls to 7, add 6-3=3 into Goods revenue

		-The SalesPerson record “Mike” is in the system, add “6” into the sales of this record
Generate reports	-Read data from data file of Goods/SalesPerson -Post on the ineditable TextArea of User Interface	