Hochschule für Angewandte Wissenschaften Hamburg



*Hamburg University of Applied Sciences*

### Fakultät Technik und Informatik

### Studiendepartment Informations- und Elektrotechnik

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2** | Self Service Drink Dispenser | | | | | | | |
| Semester / Group ***IE4/ 2*** | | |  | | | Record editor ***Liliia Kolchenko*** | | |
| Date  ***18.05.2015,***  ***08.06.2015*** | | | Participant  ***Hasan Hasanli*** | | |
| Professor  ***Prof. Dr. Hotop*** | | |
| Approvals | |  |  |  |  |  |  |  |

**Instructions:**

The task of this lab is to develop the control software for a self service drink dispenser (SSDD). Money insertion and selected drink output is to be simulated via GUI. There are ten different drinks available, each one can be selected and then deselected. After each pressing of a button the output screen is to be updated and show the current state of selection, i.e. the new selection and the total cost of the selection. Payment is realized via coin and/or banknotes insertion (cash button) and the possibility of payment by money card.

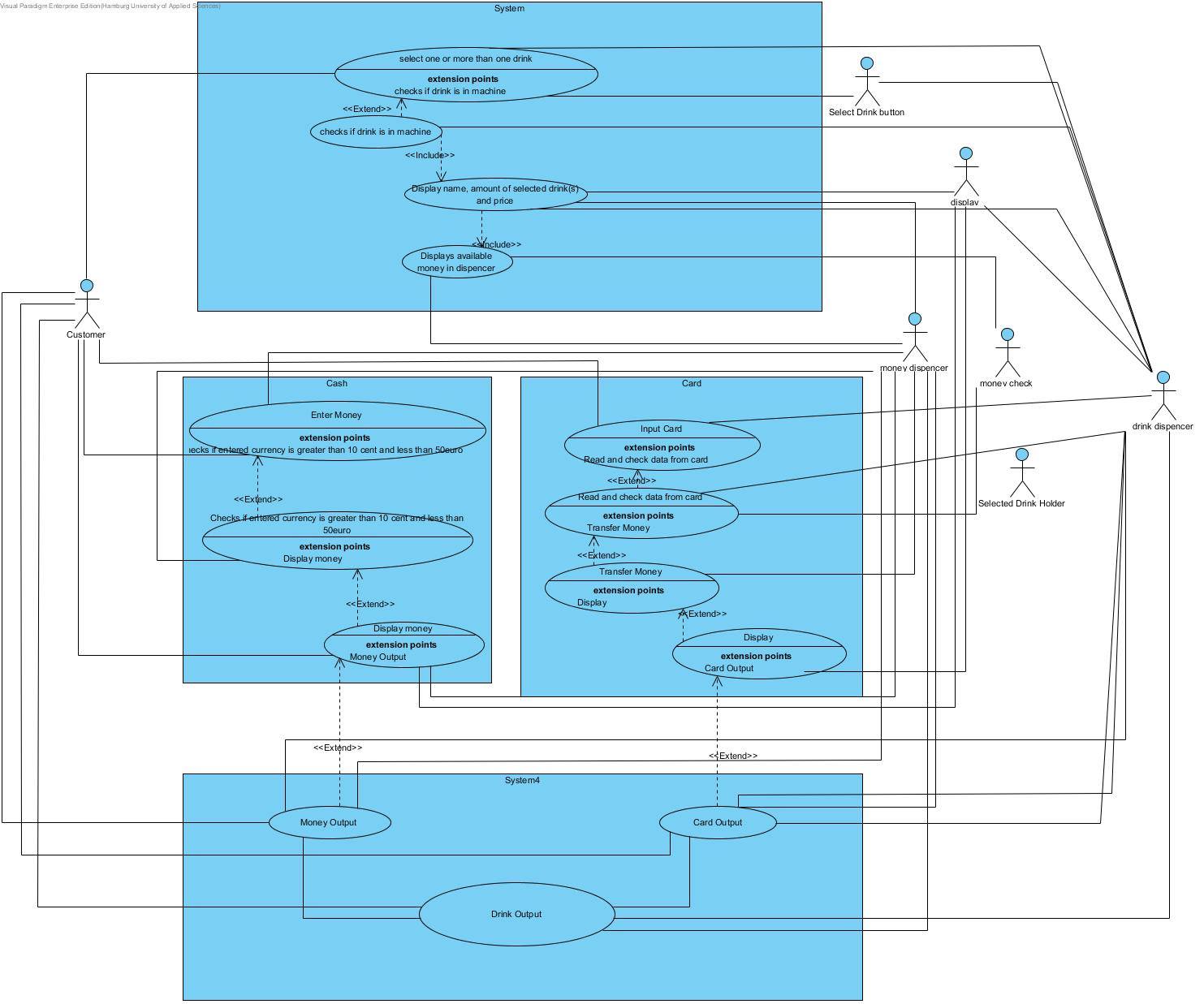
For the first step the task is analyzed, as well as the functionality of this application. The use case diagram, the class diagram, the activity diagram, the sequence and collaboration diagrams are created to document and visualize the flow of the program.

For the next step JAVA GUI programming is used to visualize the output. The final step is creating the documentation.

**Algorithm of SSDD:**

1. User has a choice either to start selecting drink(s) or input money via money card or cash.
2. For the situation when user selects drink(s) at first, in order to display the selected drink(s) the button “Select” needs to be pressed.
3. Quantity of drinks and total price is displayed.
4. After this the choice of form of payment needs to be made. For cash payment the buttons of corresponding amount of money (coins, banknotes) need to be pressed. For money card payment the amount of money to be paid is entered in text area next to “Insert card”.
5. After successful payment the user receives drinks and (if needed) the change.
6. If there is not enough money entered, then the entered money is returned and SSDD returns to initial state.
7. If before payment user decides to deselect drink(s), the button “deselect” needs to be pressed (for drinks to be deselected).
8. For the situation when user selects payment at first, the money needs to be entered. The steps 2 and 3 are followed, as well as step 5.

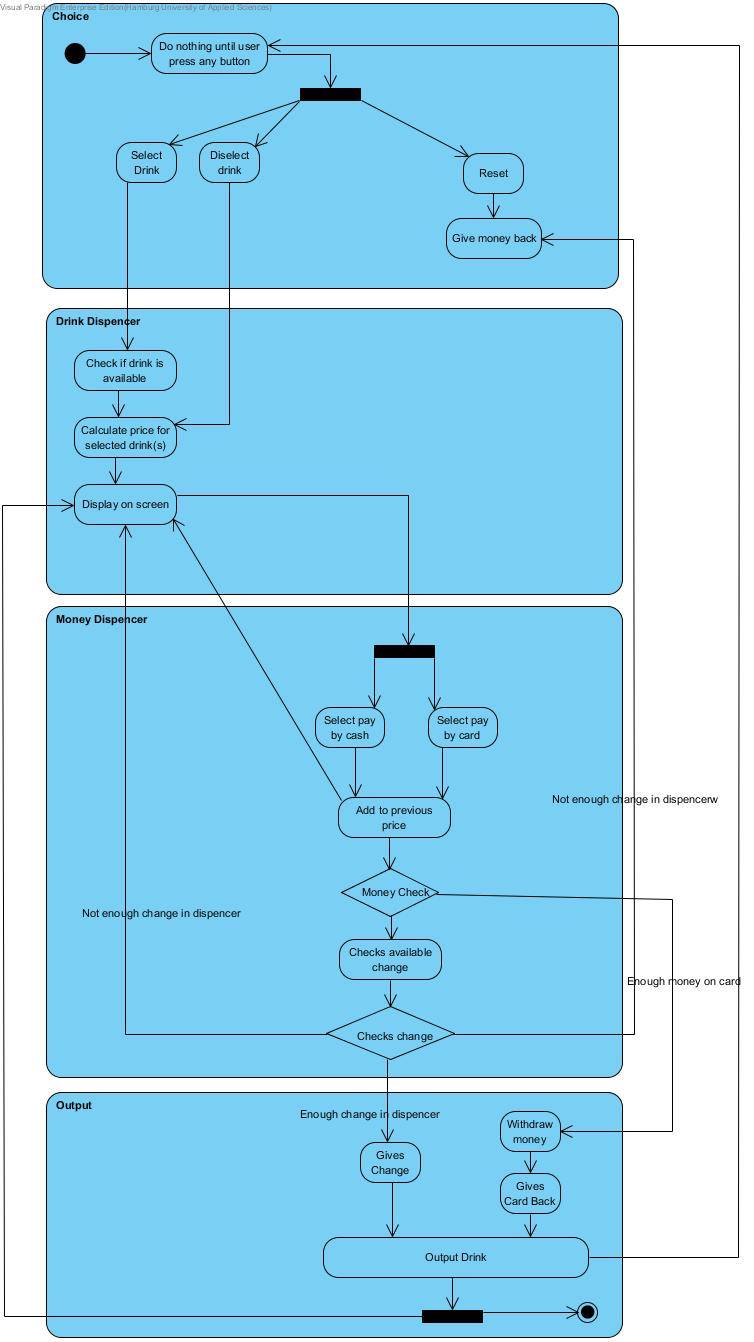
**Use case diagram:**



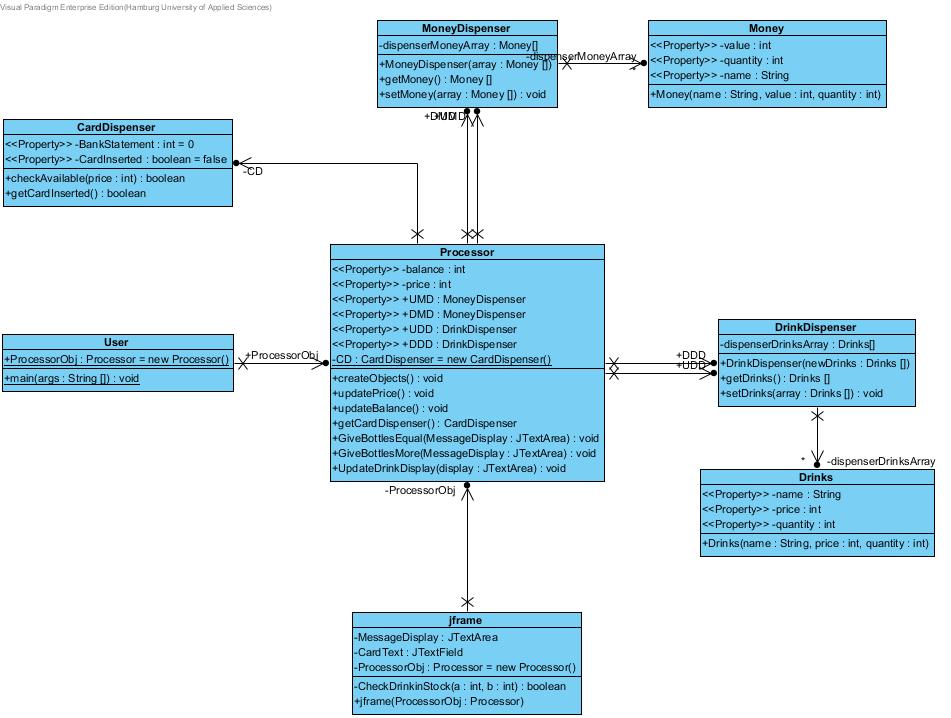
From the Use case diagram above we can see that the actors are:

Customer, Select Drink Button, Selected Drink Holder, Money Check, Drink Dispenser, Money Dispenser and Display.

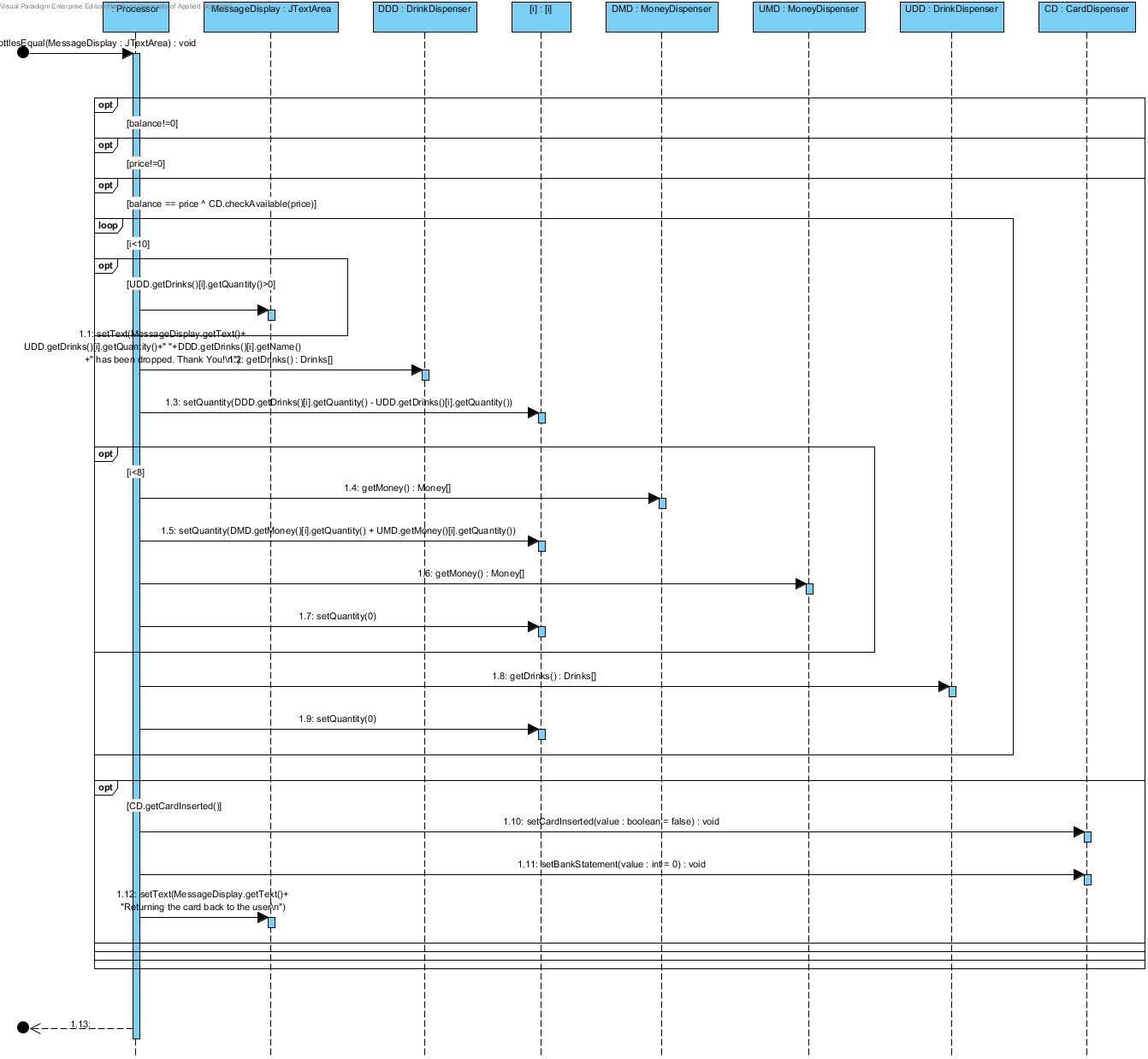
**Activity diagram:**



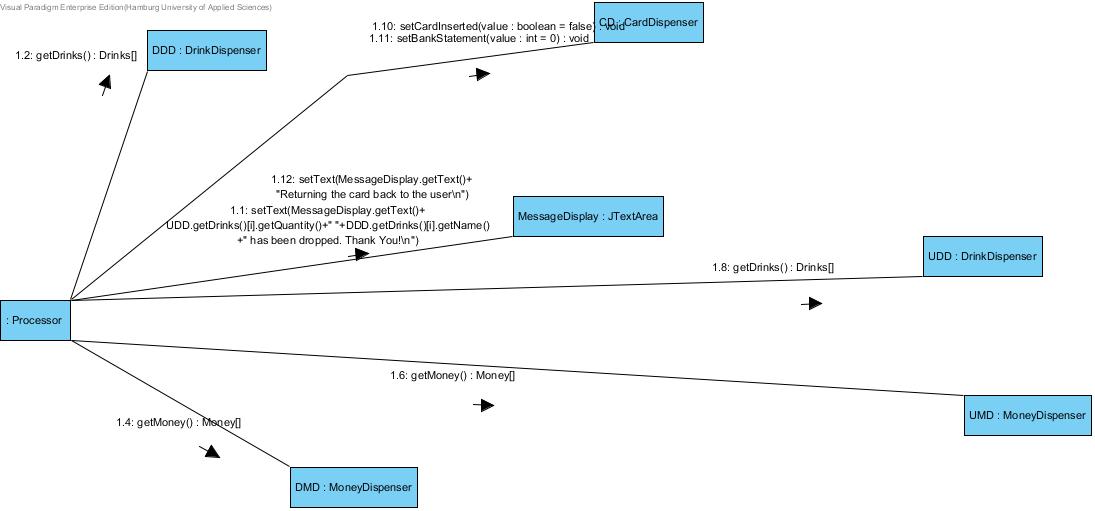
**Class diagram:**



**Sequence diagram:**



**Collaboration diagram:**



**Source code:**

Processor.java:

**import** javax.swing.JTextArea;

**public** **class** Processor {

**private** **int** balance;

**private** **int** price;

**public** MoneyDispenser UMD; // USER MONEY DISPENSER

**public** MoneyDispenser DMD; // DISPENSER MONEY DISPENSER

**public** DrinkDispenser UDD; // USER DRINK DINSPENSER

**public** DrinkDispenser DDD; // DISPENSER DRINK DISPENSER

**private** **static** CardDispenser *CD* = **new** CardDispenser();

**public** **void** createObjects(){

Money MoneyArray[] = **new** Money[8];

MoneyArray[0] = **new** Money("10cent",10,10);

MoneyArray[1] = **new** Money("20cent",20,10);

MoneyArray[2] = **new** Money("50cent",50,10);

MoneyArray[3] = **new** Money("1Euro",100,10);

MoneyArray[4] = **new** Money("2Euro",200,10);

MoneyArray[5] = **new** Money("5Euro",500,10);

MoneyArray[6] = **new** Money("10Euro",1000,10);

MoneyArray[7] = **new** Money("20Euro",2000,10);

DMD = **new** MoneyDispenser(MoneyArray);

Drinks DrinksArray[] = **new** Drinks[10];

DrinksArray[0] = **new** Drinks("Coke",110,20);

DrinksArray[1] = **new** Drinks("Sprite",110,20);

DrinksArray[2] = **new** Drinks("Beer",230,20);

DrinksArray[3] = **new** Drinks("Light Beer",240,20);

DrinksArray[4] = **new** Drinks("Fun Beer",210,20);

DrinksArray[5] = **new** Drinks("Mineral Water",100,20);

DrinksArray[6] = **new** Drinks("Apple Juice",180,20);

DrinksArray[7] = **new** Drinks("Orange Juice",190,20);

DrinksArray[8] = **new** Drinks("Tomato Juice",190,20);

DrinksArray[9] = **new** Drinks("Wine",280,20);

DDD = **new** DrinkDispenser(DrinksArray);

Money UserMoneyArray[] = **new** Money[8];

UserMoneyArray[0] = **new** Money("10cent",10,0);

UserMoneyArray[1] = **new** Money("20cent",20,0);

UserMoneyArray[2] = **new** Money("50cent",50,0);

UserMoneyArray[3] = **new** Money("1Euro",100,0);

UserMoneyArray[4] = **new** Money("2Euro",200,0);

UserMoneyArray[5] = **new** Money("5Euro",500,0);

UserMoneyArray[6] = **new** Money("10Euro",1000,0);

UserMoneyArray[7] = **new** Money("20Euro",2000,0);

UMD = **new** MoneyDispenser(UserMoneyArray);

Drinks UserDrinksArray[] = **new** Drinks[10];

UserDrinksArray[0] = **new** Drinks("Coke",110,0);

UserDrinksArray[1] = **new** Drinks("Sprite",110,0);

UserDrinksArray[2] = **new** Drinks("Beer",230,0);

UserDrinksArray[3] = **new** Drinks("Light Beer",240,0);

UserDrinksArray[4] = **new** Drinks("Fun Beer",210,0);

UserDrinksArray[5] = **new** Drinks("Mineral Water",100,0);

UserDrinksArray[6] = **new** Drinks("Apple Juice",180,0);

UserDrinksArray[7] = **new** Drinks("Orange Juice",190,0);

UserDrinksArray[8] = **new** Drinks("Tomato Juice",190,0);

UserDrinksArray[9] = **new** Drinks("Wine",280,0);

UDD = **new** DrinkDispenser(UserDrinksArray);

}

**public** DrinkDispenser getUDD(){

**return** **this**.UDD;

}

**public** MoneyDispenser getUMD(){

**return** **this**.UMD;

}

**public** DrinkDispenser getDDD(){

**return** **this**.DDD;

}

**public** MoneyDispenser getDMD(){

**return** **this**.DMD;

}

**public** **void** updatePrice(){

price=0;

Drinks Drink[] = UDD.getDrinks();

**for**(**int** i=0;i<10;i++)

price = price + Drink[i].getQuantity()\*Drink[i].getPrice();

}

**public** **void** updateBalance(){

balance=0;

Money Money[] = UMD.getMoney();

**for**(**int** i=0;i<8;i++)

balance = balance + Money[i].getQuantity()\*Money[i].getValue();

}

**public** **int** getBalance() {

**return** **this**.balance;

}

**public** **void** setBalance(**int** balance) {

**this**.balance = balance;

}

**public** **int** getPrice() {

**return** **this**.price;

}

**public** CardDispenser getCardDispenser(){

**return** *CD*;

}

**public** **void** GiveBottlesEqual(JTextArea MessageDisplay){

**if** (balance!=0)

**if** (price!=0)

**if** (balance == price ^ *CD*.checkAvailable(price)){

**for**(**int** i=0;i<10;i++){

**if** (UDD.getDrinks()[i].getQuantity()>0)

MessageDisplay.setText(MessageDisplay.getText()+

UDD.getDrinks()[i].getQuantity()+" "+DDD.getDrinks()[i].getName()

+" has been dropped. Thank You!\n");

DDD.getDrinks()[i].setQuantity(DDD.getDrinks()[i].getQuantity() - UDD.getDrinks()[i].getQuantity());

**if**(i<8){

DMD.getMoney()[i].setQuantity(DMD.getMoney()[i].getQuantity() + UMD.getMoney()[i].getQuantity());

UMD.getMoney()[i].setQuantity(0);

}

UDD.getDrinks()[i].setQuantity(0);

}

price=0;

balance=0;

**if** (*CD*.getCardInserted()){

*CD*.setCardInserted(**false**);

balance=0;

*CD*.setBankStatement(0);

MessageDisplay.setText(MessageDisplay.getText()+

"Returning the card back to the user\n");

}

}

}

**public** **void** GiveBottlesMore(JTextArea MessageDisplay){

**if** ((balance > price)& price!=0){

**for** (**int** i=0;i<8;i++){

DMD.getMoney()[i].setQuantity(DMD.getMoney()[i].getQuantity()+UMD.getMoney()[i].getQuantity());

}

Money[] tDMD = DMD.getMoney();

**int** value = balance-price;

**for**(**int** i=0; i<8;i++){

**if**((tDMD[7-i].getQuantity()!=0) && (tDMD[7-i].getValue()<=value)){

**if** (tDMD[7-i].getQuantity()<value/tDMD[7-i].getValue())

{

MessageDisplay.setText(MessageDisplay.getText()+tDMD[7-i].getQuantity()+" piece(s) of "+tDMD[7-i].getName()+" has been output\n");

tDMD[7-i].setQuantity(0);

value -= tDMD[7-i].getQuantity() \* tDMD[7-i].getValue();

}

**else**

{

MessageDisplay.setText(MessageDisplay.getText()+(value/tDMD[7-i].getValue())+" piece(s) of "+tDMD[7-i].getName()+" has been output\n");

tDMD[7-i].setQuantity(tDMD[7-i].getQuantity()-value/tDMD[7-i].getValue());

value -= (value/tDMD[7-i].getValue())\*tDMD[7-i].getValue();

}

DMD.getMoney()[7-i].setQuantity(tDMD[7-i].getQuantity());

}

}

**if** (value!=0){

MessageDisplay.setText(MessageDisplay.getText()+"Not enough money to return output, all the coins will be returned\n");

**for**(**int** i=0;i<8;i++){

**if** (UMD.getMoney()[i].getQuantity()>0)

MessageDisplay.setText(MessageDisplay.getText()+

"Returning "+UMD.getMoney()[i].getQuantity()+"piece(s) of "

+UMD.getMoney()[i].getName()+" back to user\n");

UMD.getMoney()[i].setQuantity(0);

}}

**else**{

**for**(**int** i=0;i<10;i++){

**if** (UDD.getDrinks()[i].getQuantity()>0)

MessageDisplay.setText(MessageDisplay.getText()+

UDD.getDrinks()[i].getQuantity()+" "+DDD.getDrinks()[i].getName()

+" has been dropped. Thank You!\n");

DDD.getDrinks()[i].setQuantity(DDD.getDrinks()[i].getQuantity() - UDD.getDrinks()[i].getQuantity());

**if**(i<8){

DMD.getMoney()[i].setQuantity(DMD.getMoney()[i].getQuantity() + UMD.getMoney()[i].getQuantity());

UMD.getMoney()[i].setQuantity(0);

}

UDD.getDrinks()[i].setQuantity(0);

}

price=0;

balance=0;

}

}

}

**public** **void** UpdateDrinkDisplay(JTextArea display){

display.setText("NAME\t\tQUANTITY\tPRICE\n");

Drinks array[] = UDD.getDrinks();

**for**(**int** i=0;i<10;i++)

**if** (array[i].getQuantity()>0)

display.setText(display.getText()

+"\n"+array[i].getName()+"\t\t "

+String.*valueOf*(array[i].getQuantity())+"\t"

+String.*valueOf*((**float**)(array[i].getQuantity()\*array[i].getPrice())/100)+"\u20AC");

display.setText(display.getText()+"\n\n\n\t\t\t Total: "+String.*valueOf*((**float**)(price)/100)+"\u20AC");

display.setText(display.getText()+"\n\t\t\t Balance: "+String.*valueOf*((**float**)(balance)/100)+"\u20AC");

}

}

Drinks.java:

**public** **class** Drinks {

**private** String name;

**private** **int** price;

**private** **int** quantity;

**public** Drinks(String name, **int** price, **int** quantity){

**this**.name = name;

**this**.price = price;

**this**.quantity = quantity;

}

**public** String getName() {

**return** **this**.name;

}

**public** **int** getPrice() {

**return** **this**.price;

}

**public** **int** getQuantity() {

**return** **this**.quantity;

}

**public** **void** setQuantity(**int** quantity) {

**this**.quantity = quantity;

}

}

Money.java:

**public** **class** Money {

**private** **int** value;

**private** **int** quantity;

**private** String name;

**public** Money(String name, **int** value, **int** quantity){

**this**.name = name;

**this**.quantity = quantity;

**this**.value = value;

}

**public** **int** getValue() {

**return** **this**.value;

}

**public** **int** getQuantity() {

**return** **this**.quantity;

}

**public** **void** setQuantity(**int** quantity) {

**this**.quantity = quantity;

}

**public** String getName() {

**return** **this**.name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

MoneyDispenser.java:

**public** **class** MoneyDispenser {

**private** Money[] dispenserMoneyArray;

**public** MoneyDispenser(Money[] array){

**this**.dispenserMoneyArray = array;

}

**public** Money[] getMoney()

{

**return** **this**.dispenserMoneyArray;

}

**public** **void** setMoney(Money[] array)

{

**this**.dispenserMoneyArray = array;

}

}

User.java:

**import** java.awt.EventQueue;

**public** **class** User {

**public** **static** Processor *ProcessorObj* = **new** Processor();

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

*ProcessorObj*.createObjects();

GUI frame = **new** GUI(*ProcessorObj*);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

}

DrinkDispencser.java:

**public** **class** DrinkDispenser {

**private** Drinks[] dispenserDrinksArray;

**public** DrinkDispenser(Drinks[] newDrinks){

**this**.dispenserDrinksArray = newDrinks;

}

**public** Drinks[] getDrinks()

{

**return** **this**.dispenserDrinksArray;

}

**public** **void** setDrinks(Drinks[] array)

{

**this**.dispenserDrinksArray = array;

}

}

CardDispenser.java:

**public** **class** CardDispenser {

**private** **int** BankStatement = 0;

**private** **boolean** CardInserted = **false**;

**public** **boolean** checkAvailable(**int** price) {

**if** (price!=0)

**if**(price <= **this**.BankStatement) **return** **true**;

**return** **false**;

}

**public** **void** setBankStatement(**int** value){

**this**.BankStatement = value;

}

**public** **void** setCardInserted(**boolean** value){

**this**.CardInserted = value;

}

**public** **boolean** getCardInserted(){

**return** CardInserted;

}

}

GUI.java:

**import** java.awt.EventQueue;

**import** javax.swing.JFrame;

**import** javax.swing.JTextField;

**import** javax.swing.JToggleButton;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** java.awt.Color;

**import** javax.swing.JButton;

**import** java.awt.Font;

**import** javax.swing.JTextArea;

**import** javax.swing.JScrollPane;

**public** **class** GUI **extends** JFrame {

**private** JTextArea MessageDisplay;

**private** JTextField CardText;

**private** Processor ProcessorObj = **new** Processor();

**private** **boolean** CheckDrinkinStock(**int** a, **int** b){

**for**(**int** i=0;i<10;i++) **if** (a<b) **return** **true**;

**return** **false**;

}

**public** GUI( **final** Processor ProcessorObj) {

**this**.ProcessorObj = ProcessorObj;

JFrame GUIArea = **new** JFrame();

GUIArea.setLocationByPlatform(**true**);

GUIArea.setResizable(**false**);

GUIArea.setName("Drink Vending Machine");

GUIArea.setTitle("Drink Vending Machine");

GUIArea.setBounds(0, 0, 800, 700);

GUIArea.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

GUIArea.getContentPane().setLayout(**null**);

GUIArea.setVisible(**true**);

**final** javax.swing.JButton btnSelect = **new** javax.swing.JButton();

btnSelect.setBackground(Color.*YELLOW*);

JScrollPane scrollPane = **new** JScrollPane();

scrollPane.setBounds(125, 350, 400, 65);

GUIArea.getContentPane().add(scrollPane);

MessageDisplay = **new** JTextArea();

scrollPane.setViewportView(MessageDisplay);

MessageDisplay.setRequestFocusEnabled(**false**);

MessageDisplay.setForeground(Color.*BLACK*);

MessageDisplay.setFont(**new** Font("Malgun Gothic", Font.*PLAIN*, 11));

MessageDisplay.setFocusable(**false**);

MessageDisplay.setEditable(**false**);

MessageDisplay.setBackground(Color.*YELLOW*);

btnSelect.setFocusable(**false**);

**final** JToggleButton btnDrink1 = **new** JToggleButton();

btnDrink1.setFocusable(**false**);

btnSelect.setText("Select");

GUIArea.getContentPane().add(btnSelect);

btnSelect.setBounds(565, 166, 80, 25);

JButton btnDeselect = **new** JButton();

btnDeselect.setBackground(**new** Color(220, 20, 60));

btnDeselect.setText("Deselect");

btnDeselect.setFocusable(**false**);

btnDeselect.setBounds(565, 204, 80, 25);

GUIArea.getContentPane().add(btnDeselect);

JButton btnCard = **new** JButton();

btnCard.setText("Insert Card");

btnCard.setFocusable(**false**);

btnCard.setBounds(562, 270, 100, 25);

GUIArea.getContentPane().add(btnCard);

btnDrink1.setText("Coke");

GUIArea.getContentPane().add(btnDrink1);

btnDrink1.setBounds(40, 30, 80, 25);

**final** JToggleButton btnDrink2 = **new** JToggleButton();

btnDrink2.setText("Sprite");

btnDrink2.setFocusable(**false**);

btnDrink2.setBounds(40, 80, 80, 25);

GUIArea.getContentPane().add(btnDrink2);

**final** JToggleButton btnDrink3 = **new** JToggleButton();

btnDrink3.setText("Beer");

btnDrink3.setFocusable(**false**);

btnDrink3.setBounds(40, 130, 80, 25);

GUIArea.getContentPane().add(btnDrink3);

**final** JToggleButton btnDrink4 = **new** JToggleButton();

btnDrink4.setText("Light Beer");

btnDrink4.setFocusable(**false**);

btnDrink4.setBounds(40, 180, 80, 25);

GUIArea.getContentPane().add(btnDrink4);

**final** JToggleButton btnDrink5 = **new** JToggleButton();

btnDrink5.setText("Fun Been");

btnDrink5.setFocusable(**false**);

btnDrink5.setBounds(40, 230, 80, 25);

GUIArea.getContentPane().add(btnDrink5);

**final** JToggleButton btnDrink6 = **new** JToggleButton();

btnDrink6.setText("Mineral Water");

btnDrink6.setFocusable(**false**);

btnDrink6.setBounds(40, 280, 80, 25);

GUIArea.getContentPane().add(btnDrink6);

**final** JToggleButton btnDrink7 = **new** JToggleButton();

btnDrink7.setText("Apple Juice");

btnDrink7.setFocusable(**false**);

btnDrink7.setBounds(40, 330, 80, 25);

GUIArea.getContentPane().add(btnDrink7);

**final** JToggleButton btnDrink8 = **new** JToggleButton();

btnDrink8.setText("Orange Juice");

btnDrink8.setFocusable(**false**);

btnDrink8.setBounds(40, 380, 80, 25);

GUIArea.getContentPane().add(btnDrink8);

**final** JToggleButton btnDrink9 = **new** JToggleButton();

btnDrink9.setText("Tomato Juice");

btnDrink9.setFocusable(**false**);

btnDrink9.setBounds(40, 430, 80, 25);

GUIArea.getContentPane().add(btnDrink9);

**final** JToggleButton btnDrink10 = **new** JToggleButton();

// btnDrink10.setFont(new Font("Arial", Font.BOLD, 10));

btnDrink10.setText("Wine");

btnDrink10.setFocusable(**false**);

btnDrink10.setBounds(40, 480, 80, 25);

GUIArea.getContentPane().add(btnDrink10);

**final** JTextArea DrinkDisplay= **new** JTextArea();

DrinkDisplay.setColumns(15);

DrinkDisplay.setForeground(**new** Color(0, 150, 255));

DrinkDisplay.setBackground(**new** Color(46, 50, 20));

DrinkDisplay.setFont(**new** Font("Dialog", Font.*BOLD*, 10));

DrinkDisplay.setFocusable(**false**);

DrinkDisplay.setBounds(125, 30, 325, 295);

GUIArea.getContentPane().add(DrinkDisplay);

DrinkDisplay.setEditable(**false**);

DrinkDisplay.setRequestFocusEnabled(**false**);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

CardText = **new** JTextField();

CardText.setFocusTraversalKeysEnabled(**false**);

CardText.setBounds(475, 269, 80, 25);

GUIArea.getContentPane().add(CardText);

CardText.setColumns(10);

JButton btn10c = **new** JButton();

btn10c.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money1 = ProcessorObj.getUMD().getMoney()[0];

Money1.setQuantity(Money1.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn10c.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn10c.setText("10c");

btn10c.setFocusable(**false**);

btn10c.setBounds(475, 30, 50, 25);

GUIArea.getContentPane().add(btn10c);

JButton btn20c = **new** JButton();

btn20c.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money2 = ProcessorObj.getUMD().getMoney()[1];

Money2.setQuantity(Money2.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn20c.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn20c.setText("20c");

btn20c.setFocusable(**false**);

btn20c.setBounds(535, 30, 50, 25);

GUIArea.getContentPane().add(btn20c);

JButton btn50c = **new** JButton();

btn50c.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money3 = ProcessorObj.getUMD().getMoney()[2];

Money3.setQuantity(Money3.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn50c.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn50c.setText("50c");

btn50c.setFocusable(**false**);

btn50c.setBounds(595, 30, 50, 25); //change

GUIArea.getContentPane().add(btn50c);

JButton btn1E = **new** JButton();

btn1E.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money4 = ProcessorObj.getUMD().getMoney()[3];

Money4.setQuantity(Money4.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn1E.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn1E.setText("1\u20AC");

btn1E.setFocusable(**false**);

btn1E.setBounds(655, 30, 50, 25); //change

GUIArea.getContentPane().add(btn1E);

JButton btn2E = **new** JButton();

btn2E.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money5 = ProcessorObj.getUMD().getMoney()[4];

Money5.setQuantity(Money5.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn2E.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn2E.setText("2\u20AC");

btn2E.setFocusable(**false**);

btn2E.setBounds(715, 30, 50, 25);

GUIArea.getContentPane().add(btn2E);

JButton btn5E = **new** JButton();

btn5E.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money6 = ProcessorObj.getUMD().getMoney()[5];

Money6.setQuantity(Money6.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn5E.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn5E.setText("5\u20AC");

btn5E.setFocusable(**false**);

btn5E.setBounds(475, 70, 50, 25);

GUIArea.getContentPane().add(btn5E);

JButton btn10E = **new** JButton();

btn10E.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money7 = ProcessorObj.getUMD().getMoney()[6];

Money7.setQuantity(Money7.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn10E.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn10E.setText("10\u20AC");

btn10E.setFocusable(**false**);

btn10E.setBounds(535, 70, 50, 25); //change

GUIArea.getContentPane().add(btn10E);

JButton btn20E = **new** JButton();

btn20E.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money Money8 = ProcessorObj.getUMD().getMoney()[7];

Money8.setQuantity(Money8.getQuantity()+1);

ProcessorObj.updateBalance();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

});

btn20E.setFont(**new** Font("Dialog", Font.*BOLD*, 9));

btn20E.setText("20\u20AC");

btn20E.setFocusable(**false**);

btn20E.setBounds(595, 70, 50, 25);

GUIArea.getContentPane().add(btn20E);

JButton btnReset = **new** JButton();

btnReset.setFont(**new** Font("Dialog", Font.*BOLD*, 10));

btnReset.setText("Reset");

btnReset.setFocusable(**false**);

btnReset.setBounds(565, 372, 80, 25);

GUIArea.getContentPane().add(btnReset);

**final** JToggleButton DrinkButtonArray[] = **new** JToggleButton[10];

DrinkButtonArray[0]=btnDrink1;

DrinkButtonArray[1]=btnDrink2;

DrinkButtonArray[2]=btnDrink3;

DrinkButtonArray[3]=btnDrink4;

DrinkButtonArray[4]=btnDrink5;

DrinkButtonArray[5]=btnDrink6;

DrinkButtonArray[6]=btnDrink7;

DrinkButtonArray[7]=btnDrink8;

DrinkButtonArray[8]=btnDrink9;

DrinkButtonArray[9]=btnDrink10;

**for**(**int** i=0;i<10;i++) DrinkButtonArray[i].setText(ProcessorObj.getDDD().getDrinks()[i].getName());

btnSelect.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**for**(**int** i=0;i<10;i++)

**if**(DrinkButtonArray[i].isSelected()){

Drinks Drink = ProcessorObj.getUDD().getDrinks()[i];

Drink.setQuantity(Drink.getQuantity()+1);

**if** (CheckDrinkinStock(ProcessorObj.getDDD().getDrinks()[i].getQuantity(),ProcessorObj.getUDD().getDrinks()[i].getQuantity()))

{MessageDisplay.setText(MessageDisplay.getText()+

"Not Enough bottles of type "+ProcessorObj.getDDD().getDrinks()[i].getName()

+". Drink not added to list.\n");

Drink.setQuantity(Drink.getQuantity()-1);}

DrinkButtonArray[i].setSelected(**false**);

}

ProcessorObj.updatePrice();

ProcessorObj.GiveBottlesEqual(MessageDisplay);

ProcessorObj.GiveBottlesMore(MessageDisplay);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

**if** (ProcessorObj.getCardDispenser().getCardInserted()){

**if** (ProcessorObj.getCardDispenser().checkAvailable(ProcessorObj.getPrice())){

ProcessorObj.getCardDispenser().setCardInserted(**false**);

ProcessorObj.getCardDispenser().setBankStatement(0);

ProcessorObj.setBalance(0);

MessageDisplay.setText(MessageDisplay.getText()+

"Returning the card back to the user\n");}

**else** {

MessageDisplay.setText(MessageDisplay.getText()+

"Not enough money on card. Returning the card back to the user\n");

ProcessorObj.setBalance(0);

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}

}

}});

btnDeselect.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

MessageDisplay.setText("");

**for**(**int** i=0;i<10;i++)

**if**(DrinkButtonArray[i].isSelected()){

Drinks Drink = ProcessorObj.getUDD().getDrinks()[i];

**if** (Drink.getQuantity()>0)

Drink.setQuantity(Drink.getQuantity()-1);

**else** MessageDisplay.setText(MessageDisplay.getText()+">>> No "+Drink.getName()+" to deselect\n");

DrinkButtonArray[i].setSelected(**false**);

}

ProcessorObj.updatePrice();

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}});

btnCard.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

ProcessorObj.getCardDispenser().setCardInserted(**true**);

ProcessorObj.getCardDispenser().setBankStatement(Integer.*valueOf*(CardText.getText()));

**if**(ProcessorObj.getCardDispenser().checkAvailable(ProcessorObj.getPrice())){ //if enough money on card

**for**(**int** i=0;i<8;i++) {

**if** (ProcessorObj.getUMD().getMoney()[i].getQuantity()>0)

MessageDisplay.setText(MessageDisplay.getText()+

"Returning "+ProcessorObj.getUMD().getMoney()[i].getQuantity()+"piece(s) of "

+ProcessorObj.getUMD().getMoney()[i].getName()+" back to user\n");

ProcessorObj.getUMD().getMoney()[i].setQuantity(0);

}

ProcessorObj.GiveBottlesEqual(MessageDisplay);

}

**else**

ProcessorObj.setBalance(Integer.*valueOf*(CardText.getText()));

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

}});

btnReset.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Money[] marray = ProcessorObj.getUMD().getMoney();

Drinks[] darray = ProcessorObj.getUDD().getDrinks();

**if** (ProcessorObj.getCardDispenser().getCardInserted()){

ProcessorObj.getCardDispenser().setCardInserted(**false**);

MessageDisplay.setText(MessageDisplay.getText()+

"Returning the card back to the user\n");

}

**for**(**int** i=0;i<8;i++){

**if** (marray[i].getQuantity()>0)

MessageDisplay.setText(MessageDisplay.getText()+

"Returning "+marray[i].getQuantity()+"piece(s) of "

+marray[i].getName()+" back to user\n");

marray[i].setQuantity(0);

}

**for**(**int** i=0;i<10;i++){

**if** (DrinkButtonArray[i].isSelected())

DrinkButtonArray[i].setSelected(**false**);

darray[i].setQuantity(0);

}

ProcessorObj.updateBalance();

ProcessorObj.updatePrice();

ProcessorObj.UpdateDrinkDisplay(DrinkDisplay);

MessageDisplay.setText("");

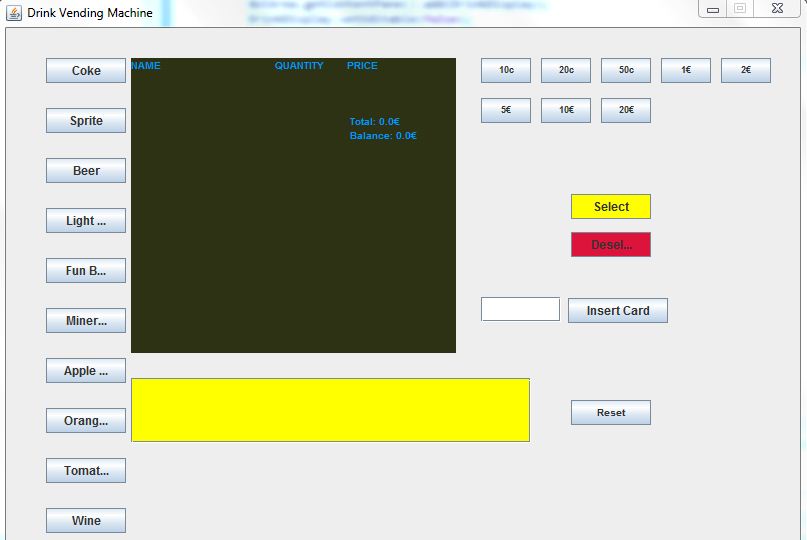
}

});

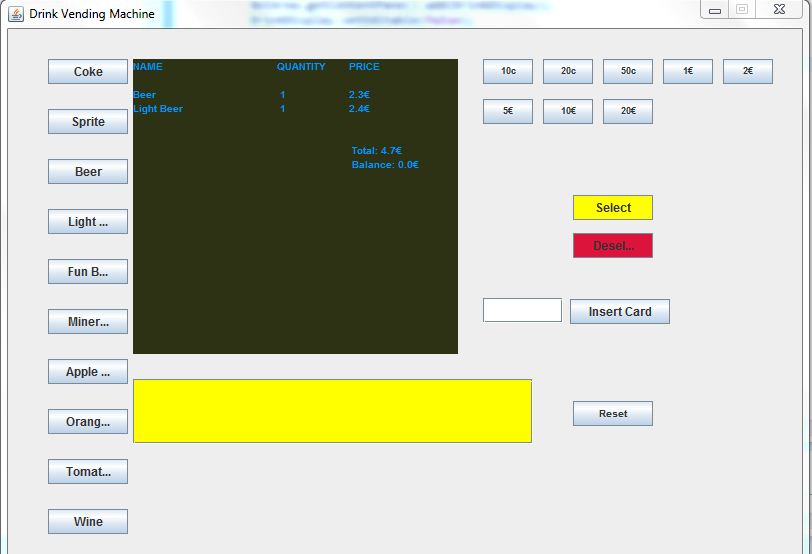
}

}

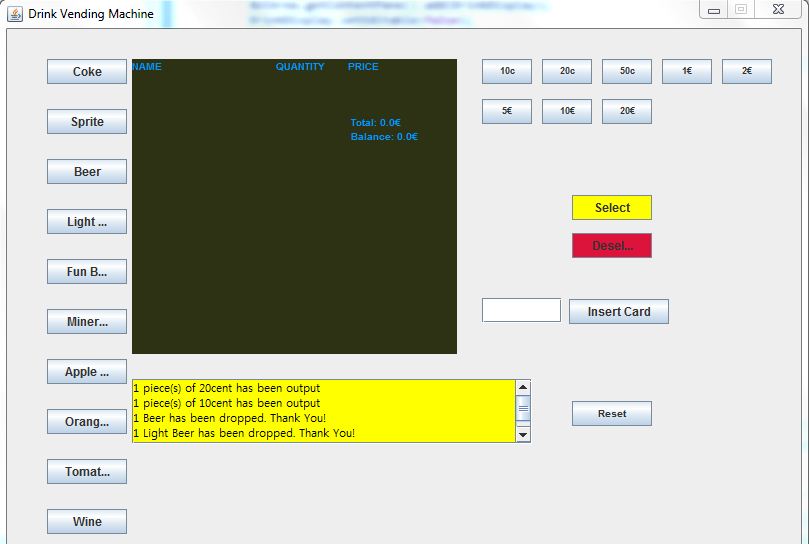
**Output via GUI:**



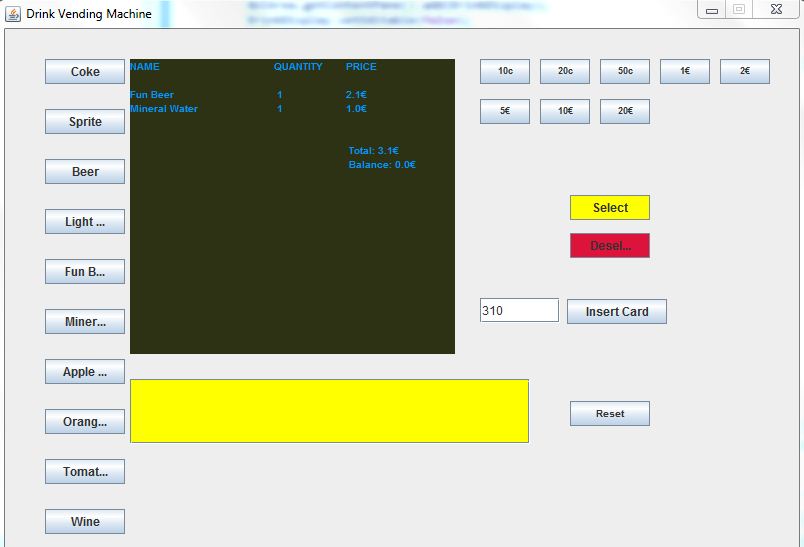
**Fig.1. Initial view**



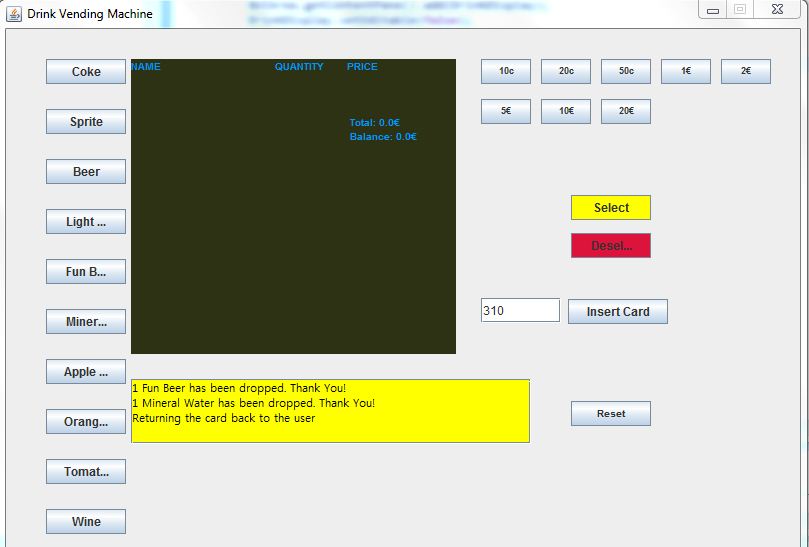
**Fig.2. Drink selection view**



**Fig.3. Payment by cash view**



**Fig.4. Drink selection and beginning of payment by card view**



**Fig.5. Payment by card view**