ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH
IT3103-744528-2024.1

BÀI THỰC HÀNH LAB 5

Họ và tên sv: Nguyễn Trịnh Hoàng Nguyên
Lớp: **K67-Việt Nhật 06**
GVHD: Lê Thị Hoa
TA: **Đặng Mạnh Cường**

Hà Nội 12/2024

Table of Contents

1. Swing components

1.1 AWTAccumulator

```java
package hust.soict.dsai.swing;

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

public class AWTAccumulator extends Frame {
    private TextField tfInput;  5 usages
    private TextField tfOutput;  4 usages
    private int sum = 0;  2 usages

    public AWTAccumulator() {  1 usage
        setLayout(new GridLayout( rows: 2, cols: 2));

        add(new Label( text: "Enter an Integer: "));

        tfInput = new TextField( columns: 10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        add(new Label( text: "The Accumulated Sum is: "));

        tfOutput = new TextField( columns: 10);
        tfOutput.setEditable(false);
        add(tfOutput);

        setTitle("AWT Accumulator");
        setSize( width: 350,  height: 120);
        setVisible(true);
    }

    public static void main(String[] args) { new AWTAccumulator(); }

    private class TFInputListener implements ActionListener {  1 usage
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(sum + "");
        }
    }
}
```

Figure 1. Source code of AWTAccumulator

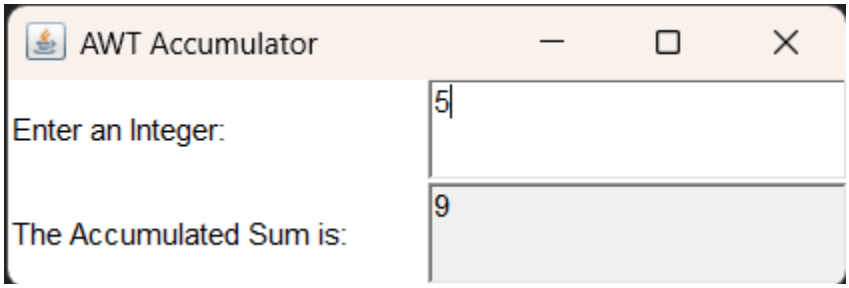

Figure 2.Demo of AWTAccumulator 1

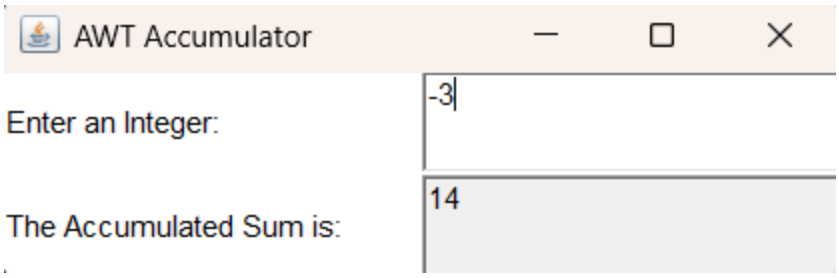

Figure 3.Demo of AWTAccumulator 2

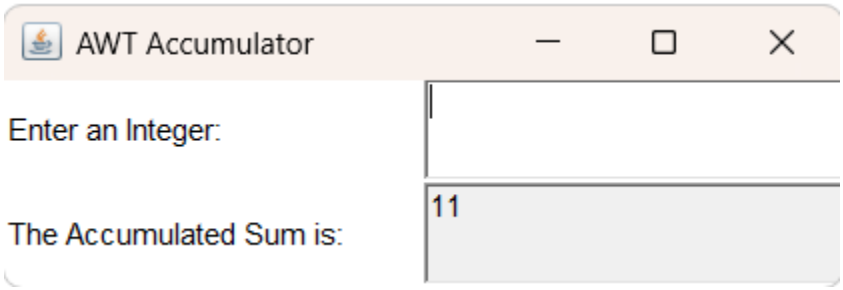

Figure 4.Demo of AWTAccumulator 3

1.2 SwingAccumulator

```java
package hust.soict.dsai.swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SwingAccumulator extends JFrame {
    private JTextField tfInput;  5 usages
    private JTextField tfOutput;  4 usages
    private int sum = 0;  2 usages

    public SwingAccumulator() {  1 usage
        Container cp = getContentPane();
        cp.setLayout(new GridLayout( rows: 2, cols: 2));

        cp.add(new JLabel( text: "Enter an Integer: "));

        tfInput = new JTextField( columns: 10);
        cp.add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        cp.add(new JLabel( text: "The Accumulated Sum is: "));

        tfOutput = new JTextField( columns: 10);
        tfOutput.setEditable(false);
        cp.add(tfOutput);

        setTitle("Swing Accumulator");
        setSize( width: 350, height: 120);
        setVisible(true);
    }
```

Figure 5.Source code of SwingAccumulator 1

```
33
34      public static void main(String[] args) {
35          new SwingAccumulator();
36      }
37
38      private class TFInputListener implements ActionListener {  1 usage
39          @Override
40          public void actionPerformed(ActionEvent evt) {
41              int numberIn = Integer.parseInt(tfInput.getText());
42              sum += numberIn;
43              tfInput.setText("");
44              tfOutput.setText(sum + "");
45          }
46      }
47  }
48
```

Figure 6.Source code of SwingAccumulator 2

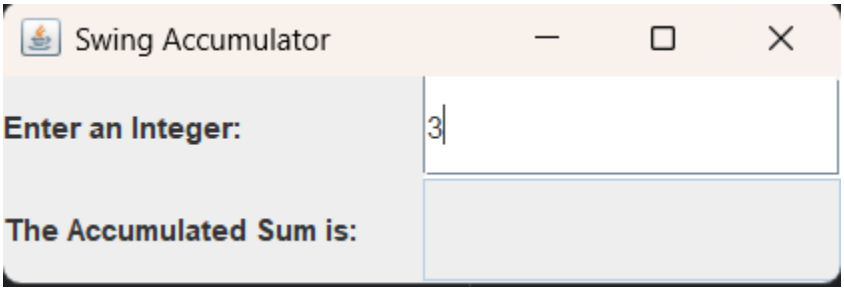

Figure 7.Demo of SwingAccumulator 1

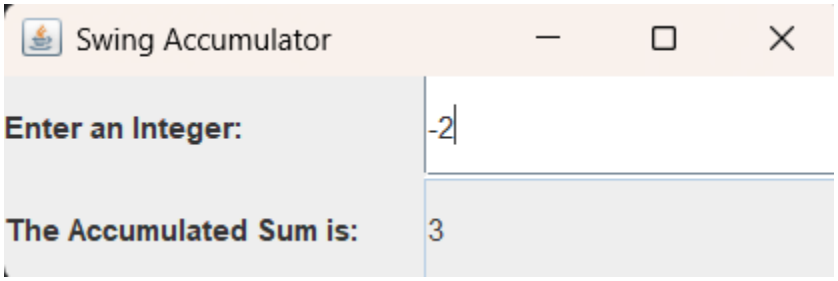

Figure 8.Demo of SwingAccumulator 2

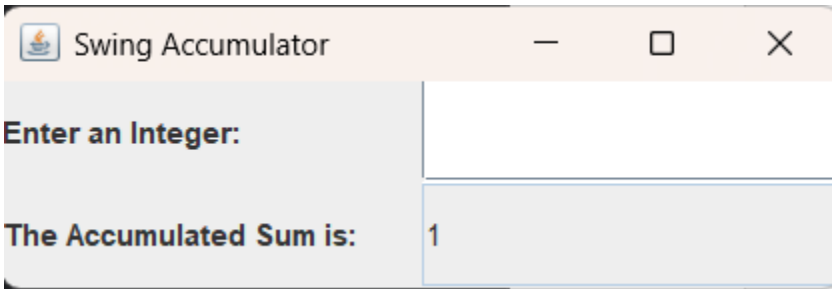

Figure 9.Demo of SwingAccumulator 3

2. Organizing Swing components with Layout Managers

2.1 Code

```java
package hust.soict.dsai.swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class NumberGrid extends JFrame {
    public static void main(String[] args) {
        new NumberGrid();
    }

    private JButton[] btnNumbers = new JButton[10]; 6 usages
    private JButton btnDelete, btnReset; 3 usages
    private JTextField tfDisplay; 8 usages

    public NumberGrid() { 1 usage

        tfDisplay = new JTextField();
        tfDisplay.setComponentOrientation(
                ComponentOrientation.RIGHT_TO_LEFT);

        JPanel panelButtons = new JPanel(new GridLayout( rows: 4, cols: 3));
        addButtons(panelButtons);

        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(tfDisplay, BorderLayout.NORTH);
        cp.add(panelButtons, BorderLayout.CENTER);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Number Grid");
        setSize( width: 200, height: 200);
        setVisible(true);
    }

    void addButtons(JPanel panelButtons) { 1 usage
        ButtonListener btnListener = new ButtonListener();
        for (int i = 1; i <= 9; i++) {
            btnNumbers[i] = new JButton( text: "" + i);
            panelButtons.add(btnNumbers[i]);
            btnNumbers[i].addActionListener(btnListener);
        }

```

Figure 10.Source code of NumberGrid 1

```
        btnDelete = new JButton( text: "DEL");
        panelButtons.add(btnDelete);
        btnDelete.addActionListener(btnListener);

        btnNumbers[0] = new JButton( text: "0");
        panelButtons.add(btnNumbers[0]);
        btnNumbers[0].addActionListener(btnListener);

        btnReset = new JButton( text: "C");
        panelButtons.add(btnReset);
        btnReset.addActionListener(btnListener);
    }

    private class ButtonListener implements ActionListener {  2 usages
        @Override
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
            if (button.charAt(0) >= '0' && button.charAt(0) <= '9') {
                tfDisplay.setText(tfDisplay.getText() + button);
            } else if (button.equals("DEL")) {
                String currentText = tfDisplay.getText();
                if (!currentText.isEmpty()) {
                    tfDisplay.setText(currentText.substring(0, currentText.length() - 1));
                }
            } else {
                tfDisplay.setText("");
            }
        }
    }
}
```

Figure 11.Source code of NumberGrid 2

2.2 Demo

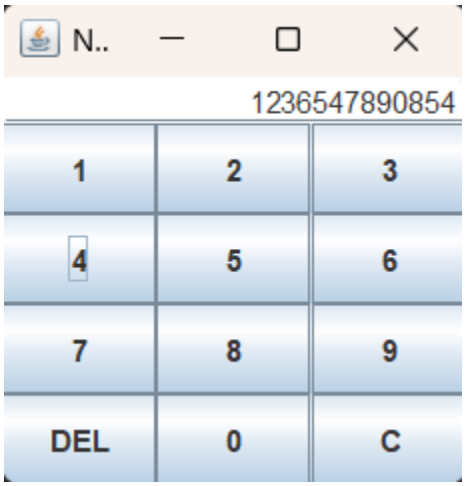

Figure 12.Demo buttons 0-9

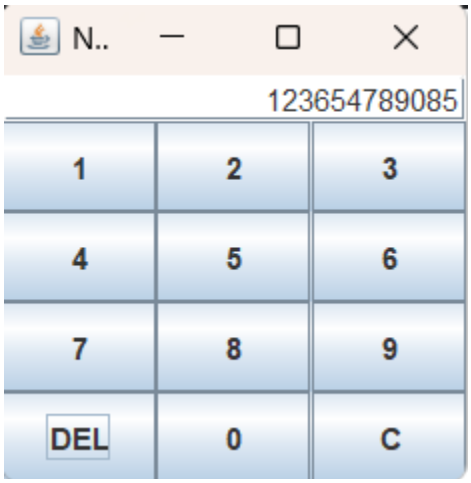

Figure 13.Demo DEL button

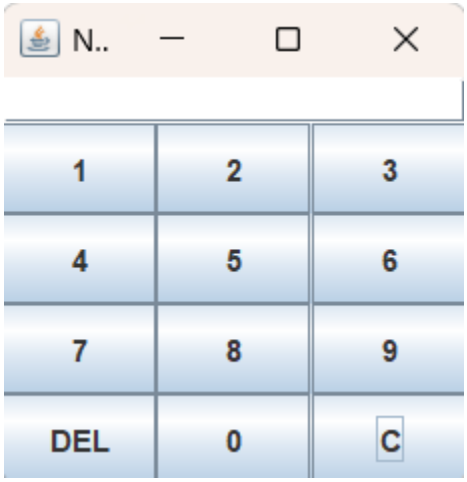

Figure 14.Demo C button

3. Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

```java
package hust.soict.dsai.aims.screen;

import hust.soict.dsai.aims.screen.*;
import hust.soict.dsai.aims.store.*;
import hust.soict.dsai.aims.cart.*;
import hust.soict.dsai.aims.exception.*;
import hust.soict.dsai.aims.media.*;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Arrays;

import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;

public class StoreScreen extends JFrame {  Hoàng Nguyên
    private Store store;  6 usages
    private Cart cart;  6 usages

    public StoreScreen(Store store, Cart cart) {  5 usages  Hoàng Nguyên
        this.store = store;
        this.cart = cart;
        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(createNorth(), BorderLayout.NORTH);
        cp.add(createCenter(), BorderLayout.CENTER);
        setVisible(true);
        setTitle("Store");
        setSize( width: 1024,  height: 768);
```

Figure 15.Class StoreScreen 1

```
42          cp.add(createCenter(), BorderLayout.CENTER);
43          setVisible(true);
44          setTitle("Store");
45          setSize( width: 1024,  height: 768);
46          Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
47          int w = getSize().width;
48          int h = getSize().height;
49          int x = (dim.width - w) / 2;
50          int y = (dim.height - h) / 2;
51          setLocation(x, y);
52      }
53
54      JPanel createNorth() {  1 usage  Hoàng Nguyên
55          JPanel north = new JPanel();
56          north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
57          north.add(createMenuBar());
58          north.add(createHeader());
59          return north;
60      }
61
62      JPanel createCenter() {  1 usage  Hoàng Nguyên
63          JPanel center = new JPanel();
64          center.setLayout(new GridLayout( rows: 5,  cols: 5,  hgap: 2,  vgap: 2));
65          ArrayList<Media> mediaInStore = store.getItemsInStore();
66          for (Media media : mediaInStore) {
67              MediaStore cell = new MediaStore(media, cart);
68              center.add(cell);
69          }
70          return center;
71      }
72
73      JMenuBar createMenuBar() {  1 usage  Hoàng Nguyên
74          JMenu menu = new JMenu( s: "Options");
75
76          JMenu smUpdateStore = new JMenu( s: "Update Store");
77          JMenuItem addBook = new JMenuItem( text: "Add Book");
78          addBook.addActionListener(new AddBookListener());
79          smUpdateStore.add(addBook);
80          JMenuItem addCD = new JMenuItem( text: "Add CD");
81          addCD.addActionListener(new AddCDListener());
82          smUpdateStore.add(addCD);
83          JMenuItem addDVD = new JMenuItem( text: "Add DVD");
84          addDVD.addActionListener(new AddDVDListener());
```

Figure 16.Class StoreScreen 2

```
        addDVD.addActionListener(new AddDVDListener());
        smUpdateStore.add(addDVD);

        menu.add(smUpdateStore);
        menu.add(new JMenuItem( text: "View store"));
        JMenuItem cart = new JMenuItem( text: "View cart");
        cart.addActionListener(new ViewCartListener());
        menu.add(cart);

        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }

    JPanel createHeader() {  1 usage  Hoàng Nguyên
        JPanel header = new JPanel();
        header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

        JLabel title = new JLabel( text: "AIMS:Nguyen Trinh Hoang Nguyen 20225656");
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN,  size: 50));
        title.setForeground(Color.CYAN);

        JButton cart = new JButton( text: "View cart");
        cart.setPreferredSize(new Dimension( width: 100,  height: 50));
        cart.setMaximumSize(new Dimension( width: 100,  height: 50));
        cart.addActionListener(new ViewCartListener());

        header.add(Box.createRigidArea(new Dimension( width: 10,  height: 10)));
        header.add(title);
        header.add(Box.createHorizontalGlue());
        header.add(cart);
        header.add(Box.createRigidArea(new Dimension( width: 10,  height: 10)));

        return header;
    }

    private class ViewCartListener implements ActionListener {  2 usages  Hoàng Nguyên
        @Override  Hoàng Nguyên
        public void actionPerformed(ActionEvent e) {
            new CartScreen(store, cart);
            dispose();
        }
```

Figure 17.Class StoreScreen 3

```
122     private class ViewCartListener implements ActionListener {  2 usages  Hoàng Nguyên
123         @Override  Hoàng Nguyên
124         public void actionPerformed(ActionEvent e) {
125             new CartScreen(store, cart);
126             dispose();
127         }
128 
129     }
130 
131     private class AddDVDListener implements ActionListener {  1 usage  Hoàng Nguyên
132         @Override  Hoàng Nguyên
133         public void actionPerformed(ActionEvent e) {
134             new AddDVDToStoreScreen(store, cart);
135             dispose();
136         }
137 
138     }
139 
140     private class AddBookListener implements ActionListener {  1 usage  Hoàng Nguyên
141         @Override  Hoàng Nguyên
142         public void actionPerformed(ActionEvent e) {
143             new AddBookToStoreScreen(store, cart);
144             dispose();
145         }
146 
147     }
148 
149     private class AddCDListener implements ActionListener {  1 usage  Hoàng Nguyên
150         @Override  Hoàng Nguyên
151         public void actionPerformed(ActionEvent e) {
152             new AddCDToStoreScreen(store, cart);
153             dispose();
154         }
155 
156     }
157 
158 }
```

Figure 18.Class StoreScreen 4

3.2 Create class MediaStore

```java
package hust.soict.dsai.aims.screen;

import hust.soict.dsai.aims.media.*;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.exception.*;

import java.awt.Color;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingConstants;

public class MediaStore extends JPanel {  Hoàng Nguyên
    private Media media;  6 usages
    private Cart cart;  2 usages

    public MediaStore(Media media, Cart cart) {  1 usage  Hoàng Nguyên
        this.media = media;
        this.cart = cart;
        this.setLayout(new BoxLayout( target: this, BoxLayout.Y_AXIS));

        JLabel title = new JLabel(media.getTitle());
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN,  size: 20));
        title.setAlignmentX(CENTER_ALIGNMENT);

        JLabel cost = new JLabel( text: "" + media.getCost() + " $");
        cost.setAlignmentX(CENTER_ALIGNMENT);

        JPanel container = new JPanel();
        container.setLayout(new FlowLayout(FlowLayout.CENTER));

        JButton addToCartButton = new JButton( text: "Add to cart");
```

Figure 19.Class MediaStore 1

```
        JButton addToCartButton = new JButton( text: "Add to cart");
        addToCartButton.addActionListener(new AddToCartListener());
        container.add(addToCartButton);

        if (media instanceof Playable) {
            JButton playButton = new JButton( text: "Play");
            playButton.addActionListener(new PlayButtonListener());
            container.add(playButton);
        }

        this.add(Box.createVerticalGlue());
        this.add(title);
        this.add(cost);
        this.add(Box.createVerticalGlue());
        this.add(container);

        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }

    private class PlayButtonListener implements ActionListener {  1 usage  Hoàng Nguyên

        @Override  Hoàng Nguyên
        public void actionPerformed(ActionEvent e) {
            try {
                ((Playable) media).play();
            } catch (PlayerException ex) {
                JPanel p = new JPanel();
                JDialog d = new JDialog();
                JLabel j1 = new JLabel( text: media.getTitle() + " cannot be played");
                JLabel j2 = new JLabel( text: "Media length is non-positive");
                p.setLayout(new BoxLayout(p, BoxLayout.Y_AXIS));
                j1.setAlignmentX(Component.CENTER_ALIGNMENT);
                j2.setAlignmentX(Component.CENTER_ALIGNMENT);
                p.add(Box.createVerticalGlue());
                p.add(j1);
                p.add(j2);
                p.add(Box.createVerticalGlue());
```

Figure 20.Class MediaStore 2

```
80                  p.add(Box.createVerticalGlue());
81                  d.add(p);
82                  d.setSize( width: 250,  height: 100);
83                  Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
84                  int w = d.getSize().width;
85                  int h = d.getSize().height;
86                  int x = (dim.width - w) / 2;
87                  int y = (dim.height - h) / 2;
88                  d.setLocation(x, y);
89                  d.setVisible(true);
90              }
91          }
92      }
93
94      private class AddToCartListener implements ActionListener {  1 usage  Hoàng Nguyên
95
96          @Override  Hoàng Nguyên
97          public void actionPerformed(ActionEvent e) {
98              JPanel p = new JPanel();
99              JDialog d = new JDialog();
100             JLabel l = new JLabel();
101             try {
102                 cart.addMedia(media);
103                 l.setText(media.getTitle() + " added to cart");
104             } catch (CartFullException ex) {
105                 l.setText("The cart is full");
106             } finally {
107                 p.setLayout(new BoxLayout(p, BoxLayout.Y_AXIS));
108                 l.setAlignmentX(Component.CENTER_ALIGNMENT);
109                 p.add(Box.createVerticalGlue());
110                 p.add(l);
111                 p.add(Box.createVerticalGlue());
112                 d.add(p);
```

Figure 21.Class MediaStore 3

```
                d.add(p);
113             d.setSize( width: 200,  height: 100);
114             Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
115             int w = d.getSize().width;
116             int h = d.getSize().height;
117             int x = (dim.width - w) / 2;
118             int y = (dim.height - h) / 2;
119             d.setLocation(x, y);
120             d.setVisible(true);
121         }
122     }
123 }
124 }
```

Figure 22.Class MediaStore 4

3.3 Demo

Figure 23.StoreScreen

Figure 24.Demo Add to cart button

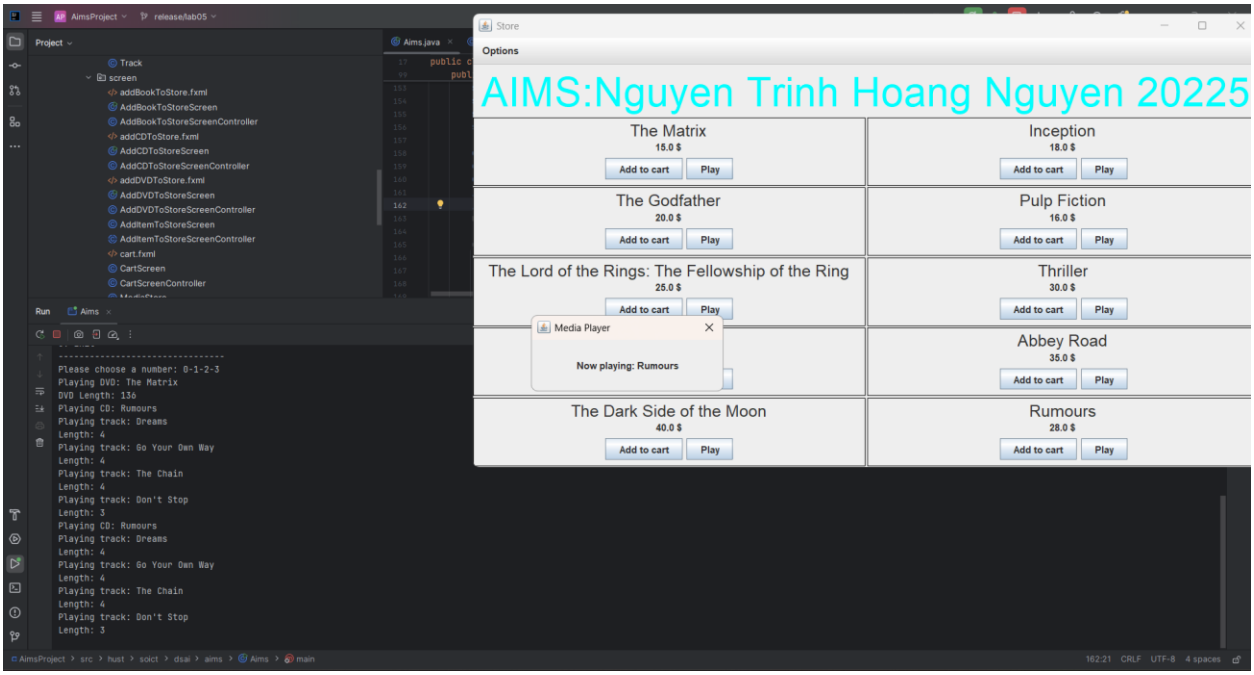

Figure 25.Demo Play button

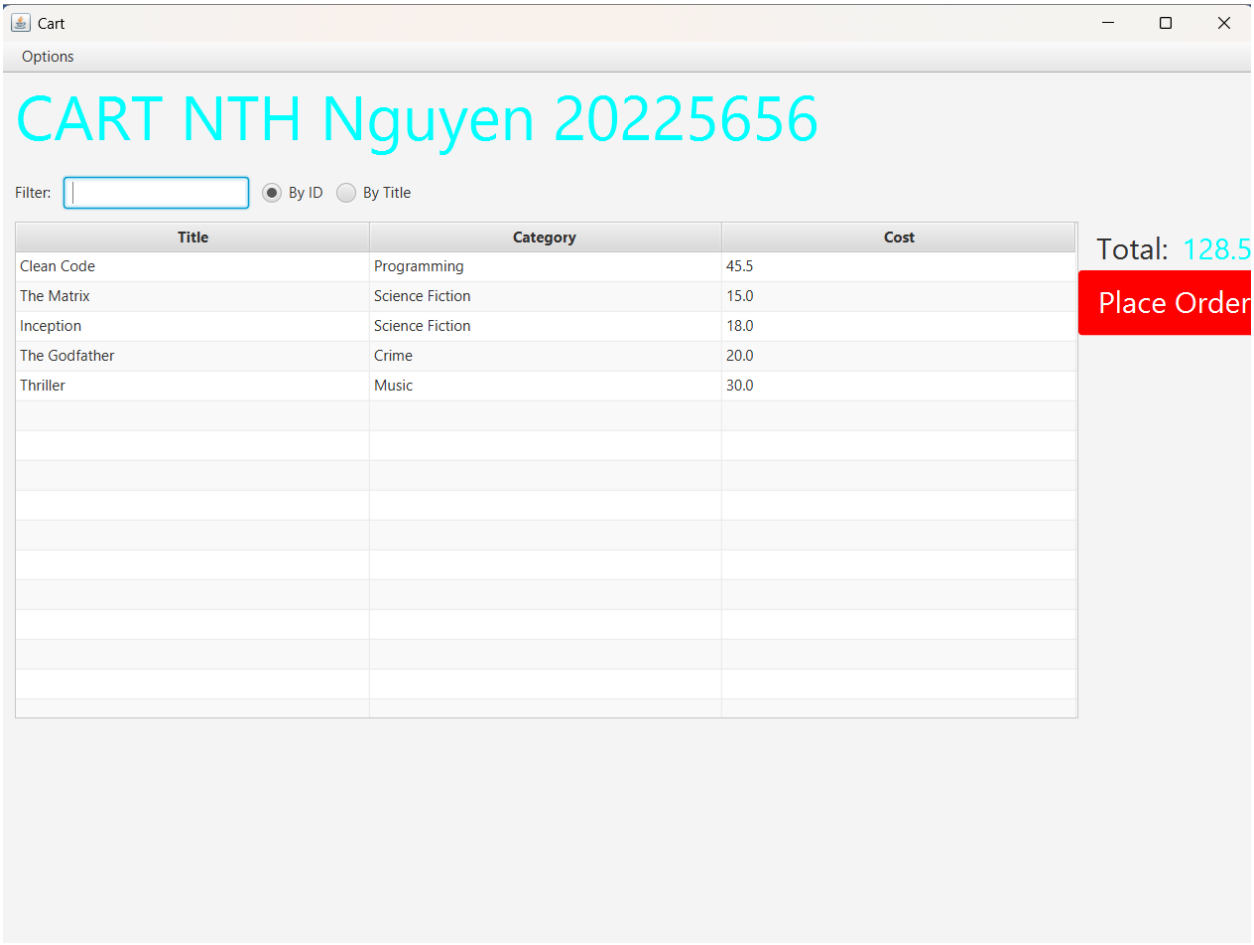

Figure 26.Demo View cart button

4. JavaFX API

4.1 Create class Painter

```java
package hust.soict.dsai.javafx;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Painter extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass()
                .getResource( name: "Painter.fxml"));

        Scene scene = new Scene(root);
        stage.setTitle("Painter");
        stage.setScene(scene);
        stage.show();

    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

Figure 27.Class Painter

4.2 Create Painter.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TitledPane?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0"
            xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="hust.soict.dsai.javafx.PainterController">
   <padding>
      <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
   </padding>
   <left>
      <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
         <BorderPane.margin>
            <Insets right="8.0" />
         </BorderPane.margin>
         <children>
            <TitledPane animated="false" maxWidth="1.7976931348623157E308" text="Tools">
               <font>
                  <Font size="13.0" />
               </font>
               <content>
                  <VBox>
                     <children>
                        <RadioButton mnemonicParsing="false" onAction="#penButtonPressed" text="Pen">
                           <font>
                              <Font size="13.0" />
                           </font>
                           <toggleGroup>
                              <ToggleGroup fx:id="chose" />
                           </toggleGroup>
                        </RadioButton>
                        <RadioButton mnemonicParsing="false" onAction="#eraserButtonPressed" text="Eraser" toggleGroup="$chose">
                           <font>
                              <Font size="13.0" />
                           </font>
                        </RadioButton>
                     </children>
```

Figure 28.Painter.fxml 1

```
                     <RadioButton mnemonicParsing="false" onAction="#penButtonPressed" text="Pen">
                        <font>
                           <Font size="13.0" />
                        </font>
                        <toggleGroup>
                           <ToggleGroup fx:id="chose" />
                        </toggleGroup>
                     </RadioButton>
                     <RadioButton mnemonicParsing="false" onAction="#eraserButtonPressed" text="Eraser" toggleGroup="$chose">
                        <font>
                           <Font size="13.0" />
                        </font>
                     </RadioButton>
                  </children>
               </VBox>
            </content>
         </TitledPane>
         <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear">
            <font>
               <Font size="13.0" />
            </font>
         </Button>
      </children>
   </VBox>
   </left>
   <center>
      <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" prefHeight="200.0" prefWidth="200.0" style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
   </center>
</BorderPane>
```

Figure 29.Painter.fxml 2

4.3 Create class PainterController

```java
package hust.soict.dsai.javafx;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.ToggleGroup;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.paint.Paint;
import javafx.scene.shape.Circle;

public class PainterController {
    private Paint penColor;

    @FXML
    private ToggleGroup chose;

    @FXML
    private Pane drawingAreaPane;

    @FXML
    void clearButtonPressed(ActionEvent event) {
        drawingAreaPane.getChildren().clear();
    }

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Circle newCircle = new Circle(event.getX(), event.getY(), 4, penColor);
        drawingAreaPane.getChildren().add(newCircle);
    }

    @FXML
    void eraserButtonPressed(ActionEvent event) {
        penColor = Color.WHITE;
    }

    @FXML
    void penButtonPressed(ActionEvent event) {
        penColor = Color.BLACK;
    }

}
```

Figure 30.Class PainterController

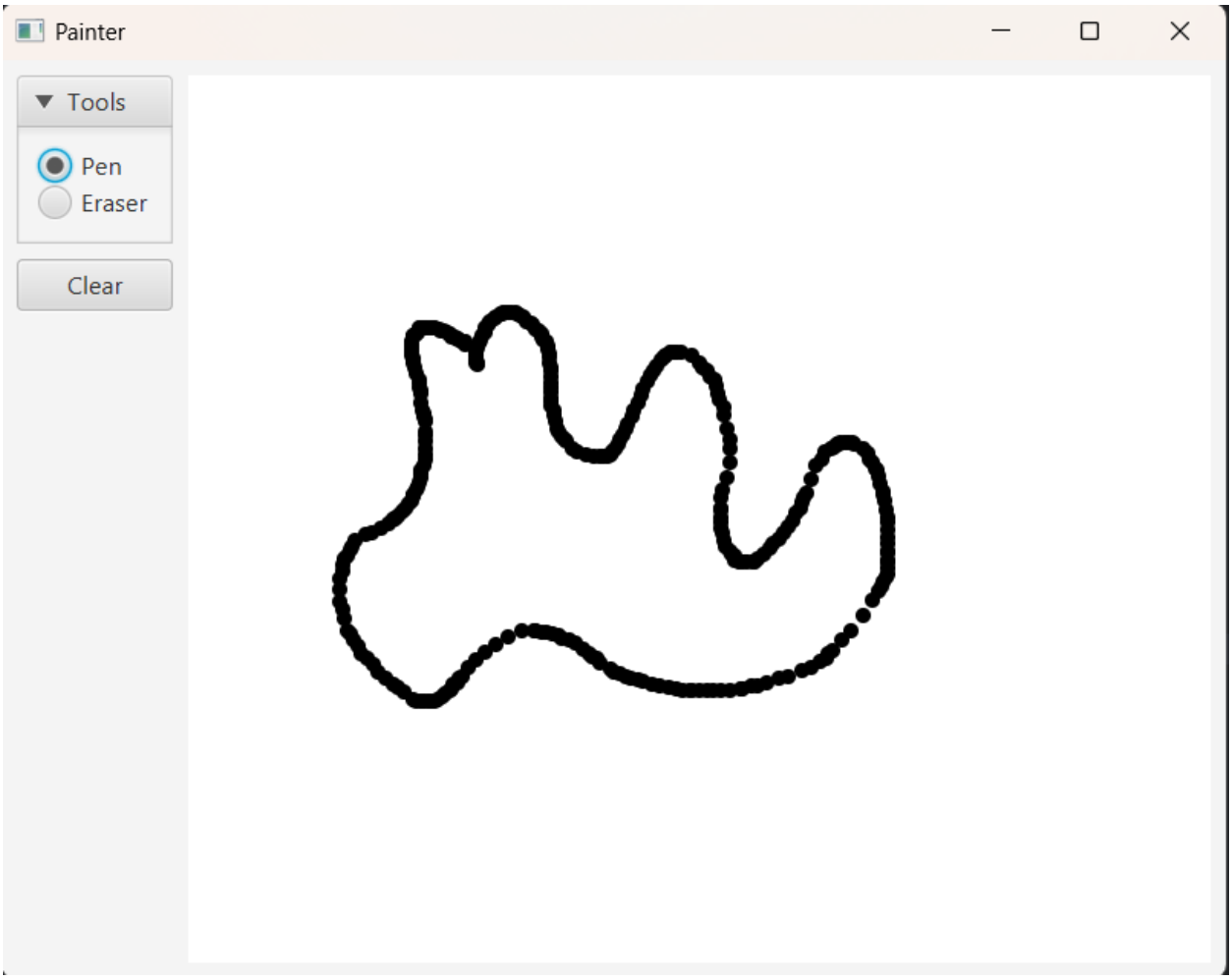

Figure 31.Use Pen

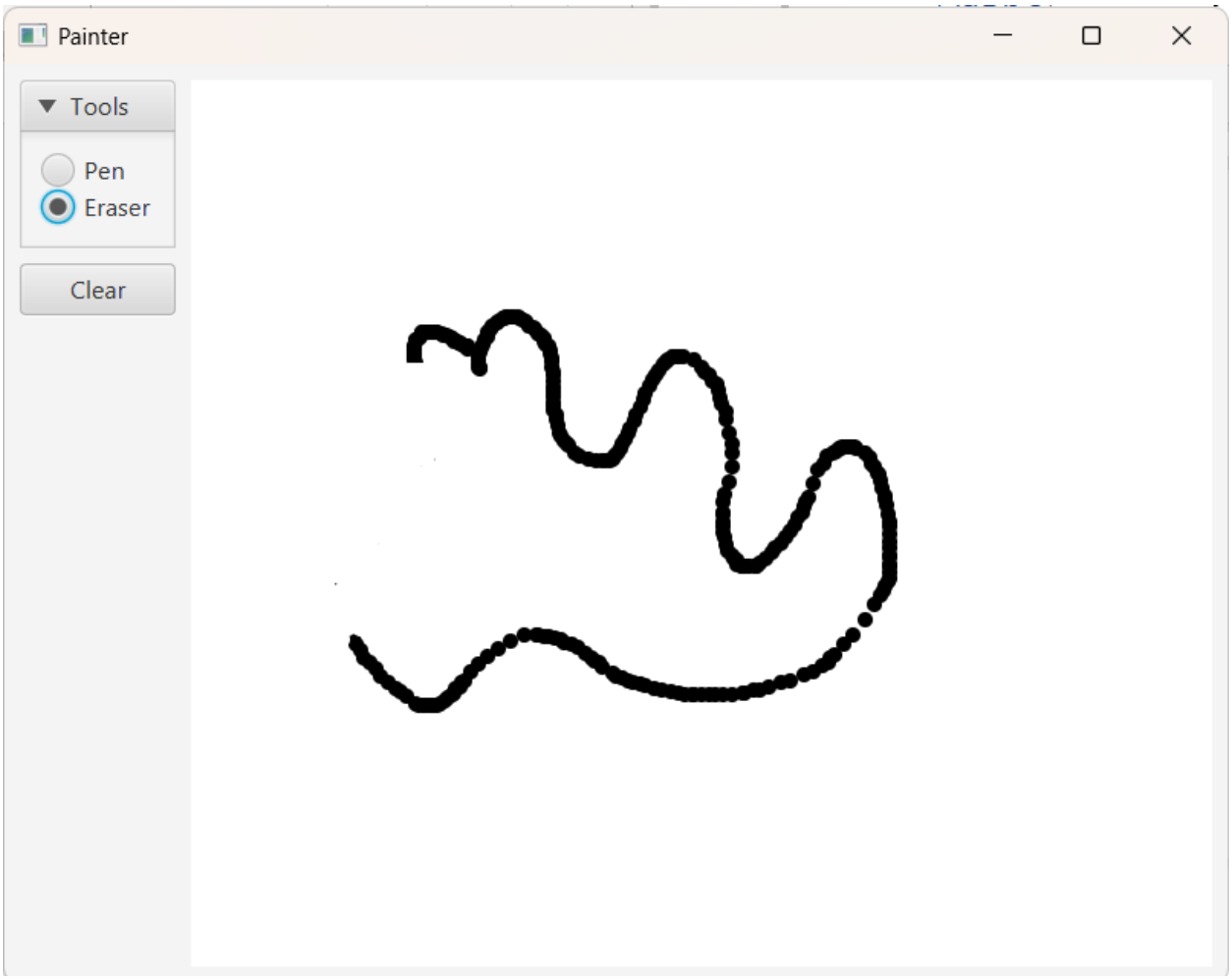

Figure 32.Use Eraser

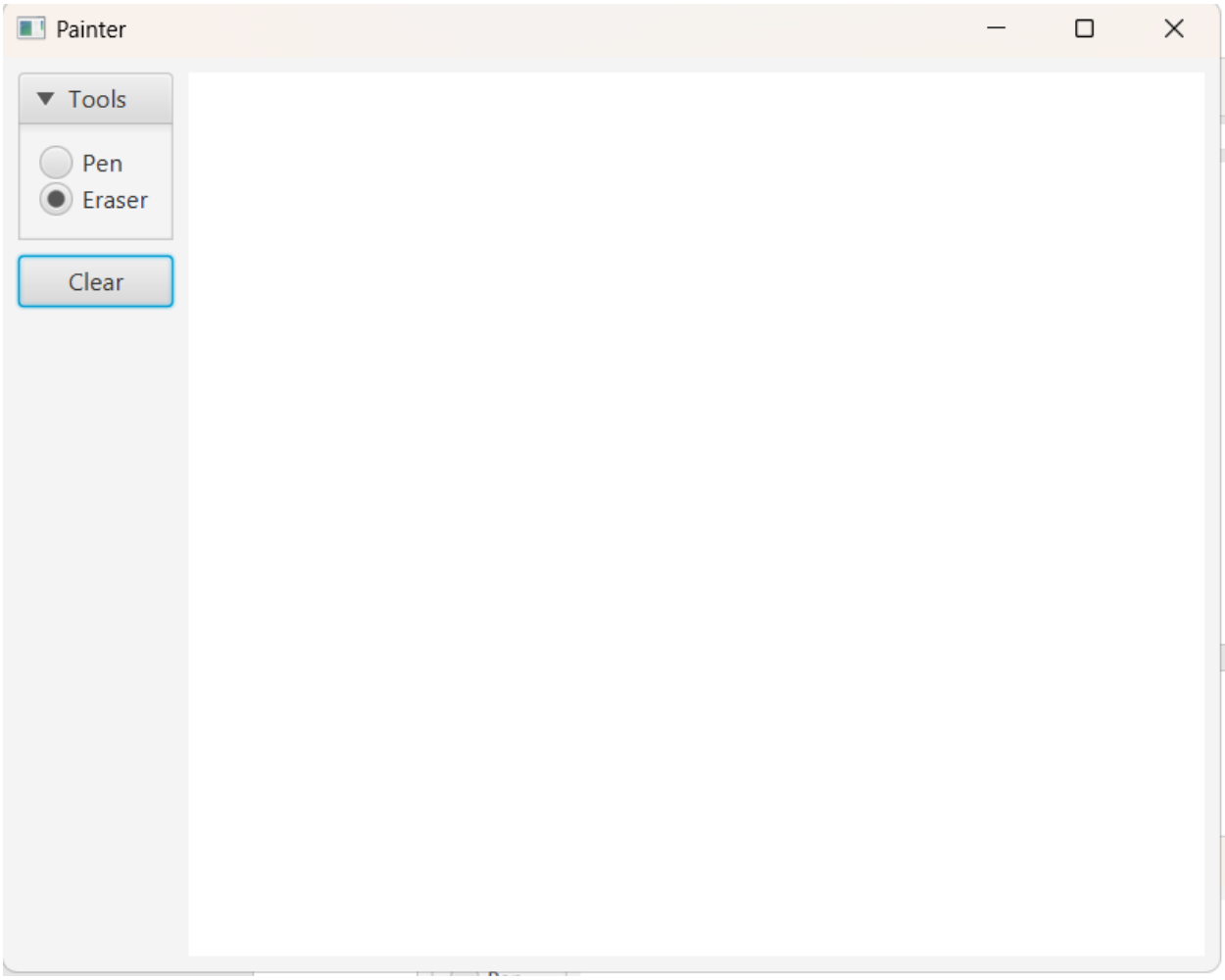

Figure 33.Clear button

5. View Cart Screen

5.1 Create cart.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ButtonBar?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="768.0" prefWidth="1024.0"
            xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1">
      <top>
         <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
            <children>
               <MenuBar>
                  <menus>
                    <Menu mnemonicParsing="false" text="Options">
                      <items>
                          <Menu mnemonicParsing="false" text="Update Store">
                            <items>
                              <MenuItem mnemonicParsing="false" onAction="#menuAddBook" text="Add Book" />
                                  <MenuItem mnemonicParsing="false" onAction="#menuAddCD" text="Add CD" />
                                  <MenuItem mnemonicParsing="false" onAction="#menuAddDVD" text="Add DVD" />
                            </items>
                          </Menu>
                        <MenuItem mnemonicParsing="false" onAction="#viewStore" text="View Store" />
                           <MenuItem mnemonicParsing="false" text="View Cart" />
                      </items>
                    </Menu>
                  </menus>
               </MenuBar>
               <Label text="CART NTH Nguyen 20225656" textFill="AQUA">
                  <font>
```

Figure 34.Cart.fxml 1

```
                        <Font size="50.0" />
                     </font>
                     <padding>
                        <Insets left="10.0" />
                     </padding>
                  </Label>
               </children>
            </VBox>
         </top>
         <center>
            <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
               <children>
                  <HBox alignment="CENTER_LEFT" prefWidth="200.0" spacing="10.0">
                     <opaqueInsets>
                        <Insets />
                     </opaqueInsets>
                     <padding>
                        <Insets bottom="10.0" top="10.0" />
                     </padding>
                     <children>
                        <Label text="Filter:" />
                        <TextField fx:id="tfFilter" />
                        <RadioButton fx:id="radioBtnFilterId" mnemonicParsing="false" onAction="#setFilterByID" selected="true" text="By ID">
                           <toggleGroup>
                              <ToggleGroup fx:id="filterCategory" />
                           </toggleGroup>
                        </RadioButton>
                        <RadioButton fx:id="radioBtnFilterTitle" mnemonicParsing="false" onAction="#setFilterByTitle" text="By Title" toggleGroup="$filte
                     </children>
                  </HBox>
                  <TableView fx:id="tblMedia">
                    <columns>
                      <TableColumn fx:id="colMediaTitle" prefWidth="75.0" text="Title" />
                      <TableColumn fx:id="colMediacateggory" prefWidth="75.0" text="Category " />
                      <TableColumn fx:id="colMediaCost" prefWidth="75.0" text="Cost" />
                    </columns>
                     <columnResizePolicy>
                        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
                     </columnResizePolicy>
```

Figure 35.Cart.fxml 2

```
                  <ButtonBar prefHeight="40.0" prefWidth="200.0">
                    <buttons>
                      <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed" text="PLay" />
                      <Button fx:id="btnRemove" mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
                    </buttons>
                  </ButtonBar>
               </children>
               <padding>
                  <Insets left="10.0" />
               </padding>
            </VBox>
         </center>
         <right>
            <VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
               <padding>
                  <Insets top="50.0" />
               </padding>
               <children>
                  <HBox alignment="CENTER" spacing="10.0">
                     <children>
                        <Label text="Total:">
                           <font>
                              <Font size="24.0" />
                           </font>
                        </Label>
                        <Label fx:id="totalCost" text="0 $" textFill="AQUA">
                           <font>
                              <Font size="24.0" />
                           </font>
                        </Label>
                     </children>
                  </HBox>
                  <Button mnemonicParsing="false" onAction="#placeOrderPressed" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">
                     <font>
                        <Font size="24.0" />
                     </font>
                  </Button>
               </children>
            </VBox>
         </right>
```

Figure 36.Cart.fxml 3

5.2 Create class CartScreen

```java
package hust.soict.dsai.aims.screen;

import ...

public class CartScreen extends JFrame {  Hoàng Nguyên
    private Cart cart;  1 usage
    private Store store;  1 usage

    public CartScreen(Store store,Cart cart) {  2 usages  Hoàng Nguyên
        super();
        this.store = store;
        this.cart = cart;
        this.setSize( width: 1024, height: 768);
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);

        this.setTitle("Cart");
        this.setVisible(true);
        JFrame frame = this;

        this.addWindowListener((WindowAdapter) windowClosing(e) -> {
                new StoreScreen(store, cart);
                dispose();  // Giải phóng tài nguyên cửa sổ hiện tại
        });

        Platform.runLater(new Runnable() {  Hoàng Nguyên
            @Override  Hoàng Nguyên
            public void run() {
                try{
                    FXMLLoader loader = new FXMLLoader(getClass()
                            .getResource( name: "cart.fxml"));
                    CartScreenController controller =
                            new CartScreenController(store, cart,frame);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                }catch (IOException e){
                    e.printStackTrace();
                }
            }
        });
    }
}
```

Figure 37.CartScreen class

5.3 Create class CartScreenController

```java
package hust.soict.dsai.aims.screen;

import ...;


public class CartScreenController {  Hoàng Nguyên

    private Store store;  5 usages
    private Cart cart;  14 usages
    private JFrame stage;  5 usages
    private FilteredList<Media> filteredCart;  4 usages
    private boolean filterByID = true;  3 usages

    @FXML  3 usages
    private Label totalCost;

    @FXML  no usages
    private ToggleGroup filterCategory;

    @FXML  4 usages
    private Button btnPlay;

    @FXML  3 usages
    private Button btnRemove;

    @FXML  1 usage
    private TextField tfFilter;

    @FXML  no usages
    private RadioButton radioBtnFilterId;

    @FXML  no usages
    private RadioButton radioBtnFilterTitle;

    @FXML  4 usages
    private TableView<Media> tblMedia;

    @FXML  1 usage
    private TableColumn<Media, String> colMediaTitle;

    @FXML  1 usage
    private TableColumn<Media, String> colMediacategory;

    @FXML  1 usage
    private TableColumn<Media, Float> colMediaCost;
```

Figure 38.CartScreenController 1

```java
public CartScreenController(Store store, Cart cart, JFrame stage) {  1 usage  Hoàng Nguyên
    super();
    this.cart = cart;
    this.store = store;
    this.stage = stage;
}

@FXML  no usages  Hoàng Nguyên
private void initialize() {
    filteredCart = new FilteredList<Media>(this.cart.getItemsOrdered(), s -> true);

    colMediaTitle.setCellValueFactory(
            new PropertyValueFactory<Media, String>( s: "Title"));
    colMediacategory.setCellValueFactory(
            new PropertyValueFactory<Media, String>( s: "Category"));
    colMediaCost.setCellValueFactory(
            new PropertyValueFactory<Media, Float>( s: "Cost"));
    tblMedia.setItems(this.cart.getItemsOrdered());

    btnPlay.setVisible(false);
    btnRemove.setVisible(false);

    totalCost.setText(String.valueOf(this.cart.totalCost()));

    tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {  Hoàng Nguyên

            @Override  Hoàng Nguyên
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
            }
        }
    );

    tfFilter.textProperty().addListener(new ChangeListener<String>() {  Hoàng Nguyên
        @Override  Hoàng Nguyên
        public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
            showFilteredMedia(newValue);
        }
    });
}
```

Figure 39.CartScreenController 2

```
105
106     @FXML  no usages  Hoàng Nguyên
107     private void btnRemovePressed(ActionEvent event) throws NonExistingItemException {
108         // Lấy đối tượng media được chọn
109         Media media = tblMedia.getSelectionModel().getSelectedItem();
110
111         // Kiểm tra xem có item nào được chọn hay không
112         if (media == null) {
113             // Nếu không có item nào được chọn, hiển thị thông báo lỗi
114             Alert alert = new Alert(Alert.AlertType.WARNING);
115             alert.setTitle("Selection Error");
116             alert.setHeaderText("No item selected");
117             alert.setContentText("Please select an item to remove.");
118             alert.showAndWait();
119             return; // Dừng phương thức nếu không có item được chọn
120         }
121
122         // Thử xóa media khỏi giỏ hàng
123         try {
124             cart.removeMedia(media);
125             // Cập nhật lại tổng chi phí sau khi xóa
126             totalCost.setText(String.valueOf(this.cart.totalCost()));
127             // Hiển thị thông báo thành công
128             Alert alert = new Alert(Alert.AlertType.INFORMATION);
129             alert.setTitle("Success");
130             alert.setHeaderText("Item Removed");
131             alert.setContentText(media.getTitle() + " has been removed from the cart.");
132             alert.showAndWait();
133         } catch (NonExistingItemException e) {
134             // Nếu media không tồn tại trong giỏ hàng, ném ngoại lệ
135             throw new NonExistingItemException("Item does not exist in the cart.");
136         }
137     }
138
139
140     @FXML  no usages  Hoàng Nguyên
141     private void setFilterByID() { this.filterByID = true; }
144
145     @FXML  no usages  Hoàng Nguyên
146     private void setFilterByTitle() { this.filterByID = false; }
149
150     void updateButtonBar(Media media) {  1 usage  Hoàng Nguyên
151         if (media == null) {
152             btnPlay.setVisible(false);
```

Figure 40.CartScreenController 3

```
150     void updateButtonBar(Media media) {  1 usage  Hoàng Nguyên
151         if (media == null) {
152             btnPlay.setVisible(false);
153             btnRemove.setVisible(false);
154         } else {
155             btnRemove.setVisible(true);
156             if (media instanceof Playable) {
157                 btnPlay.setVisible(true);
158             } else {
159                 btnPlay.setVisible(false);
160             }
161         }
162     }
163
164     private void showFilteredMedia(String filter) {  1 usage  Hoàng Nguyên
165         if (filter == null || filter.length() == 0) {
166             filteredCart.setPredicate(s -> true);
167         } else {
168             // Nếu bộ lọc là ID, chỉ hiển thị sản phẩm có ID khớp.
169             if (filterByID) {
170                 try {
171                     filteredCart.setPredicate(s -> s.getId() == Integer.parseInt(filter));
172                 } catch (NumberFormatException e) {
173                 }
174             } else //Nếu bộ lọc là tên, hiển thị sản phẩm có tên chứa từ khóa (bỏ qua chữ hoa/chữ thường).
175             {
176                 filteredCart.setPredicate(s -> s.getTitle().toLowerCase().contains(filter));
177             }
178         }
179     }
180
181     @FXML  no usages  Hoàng Nguyên
182     private void btnPlayPressed(ActionEvent event) throws PlayerException {
183         // Lấy đối tượng media được chọn
184         Media media = this.tblMedia.getSelectionModel().getSelectedItem();
185
186         // Kiểm tra xem media có phải là đối tượng Playable không
187         if (media instanceof Playable) {
188             try {
189                 // Ép kiểu media thành Playable và gọi phương thức play
190                 ((Playable) media).play();
191             } catch (Exception e) {
192                 // Xử lý lỗi phát video hoặc âm thanh (nếu có)
193                 throw new PlayerException("Error while trying to play the media: " + e.getMessage(), e);
```

Figure 41.CartScreenController 4

```
                throw new PlayerException("Error while trying to play the media: " + e.getMessage(), e);
            }
        } else {
            // Nếu media không phải là Playable, hiển thị thông báo lỗi
            throw new PlayerException("Selected media is not playable.");
        }
    }


    @FXML  no usages  Hoàng Nguyên
    private void placeOrderPressed(ActionEvent event) {
        // Kiểm tra nếu giỏ hàng trống
        if (cart.getItemsOrdered().isEmpty()) {
            // Hiển thị thông báo lỗi
            Alert emptyCartAlert = new Alert(Alert.AlertType.ERROR);
            emptyCartAlert.setTitle("Empty Cart");
            emptyCartAlert.setHeaderText(null);
            emptyCartAlert.setContentText("Your cart is empty. Please add items before placing an order.");
            emptyCartAlert.showAndWait();
            return;
        }

        // Tính tổng chi phí
        float totalCost1 = cart.totalCost();

        // Hiển thị thông báo thành công
        Alert successAlert = new Alert(Alert.AlertType.INFORMATION);
        successAlert.setTitle("Order Placed");
        successAlert.setHeaderText(null);
        successAlert.setContentText("Your order has been placed successfully!\nTotal cost: " + totalCost1);
        successAlert.showAndWait();
        // Xóa giỏ hàng
        cart.clear();
        totalCost.setText(String.valueOf(this.cart.totalCost()));

    }

    @FXML  no usages  Hoàng Nguyên
    void menuAddBook(ActionEvent event) {
        new AddBookToStoreScreen(store, cart);   // Hiển thị giao diện thêm Book
        stage.hide();
    }

```

Figure 42.CartScreenController 5

```
230     @FXML  no usages  Hoàng Nguyên
231     void menuAddBook(ActionEvent event) {
232         new AddBookToStoreScreen(store, cart);   // Hiển thị giao diện thêm Book
233         stage.hide();
234     }
235
236     @FXML  no usages  Hoàng Nguyên
237     void menuAddCD(ActionEvent event) {
238         new AddCDToStoreScreen(store,cart);
239         stage.hide();
240     }
241
242     @FXML  no usages  Hoàng Nguyên
243     void menuAddDVD(ActionEvent event) {
244         new AddDVDToStoreScreen(store,cart);
245         stage.hide();
246     }
247
248     @FXML  no usages  Hoàng Nguyên
249     void viewStore(ActionEvent event) {
250         new StoreScreen(store,cart);
251         stage.hide();
252     }
253 }
254
255
```

Figure 43.CartScreenController 6

5.4 Demo

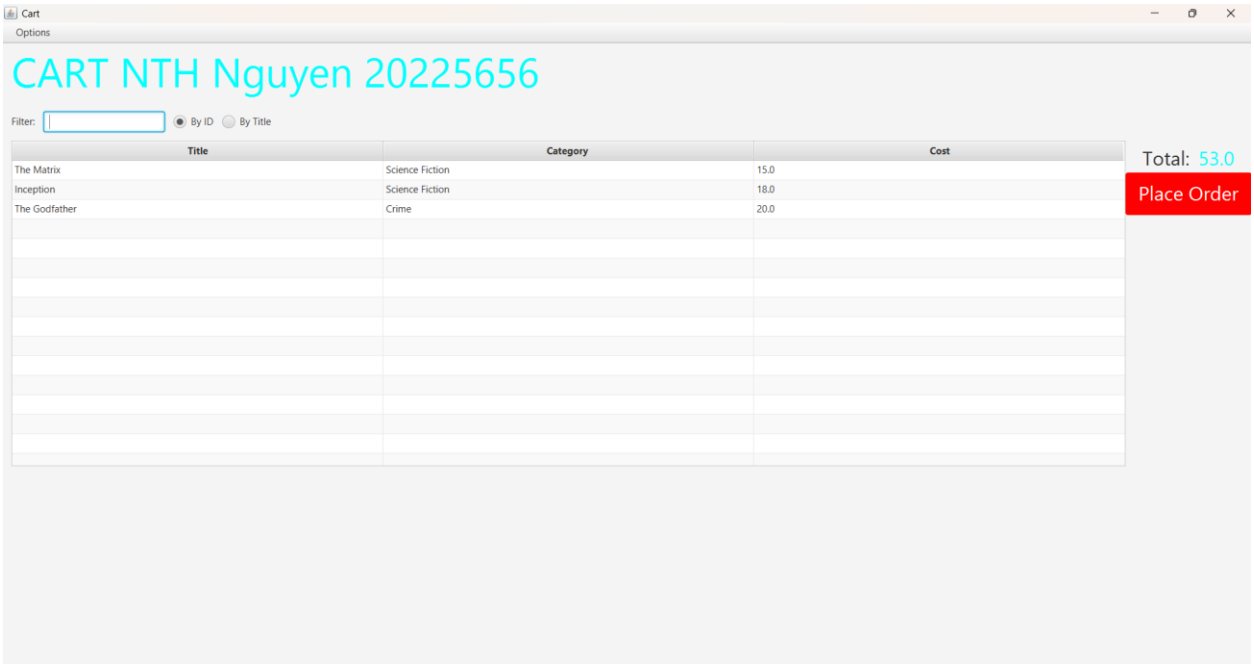

Figure 44.Demo CartScreen

6. Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController

```
@FXML  no usages  Hoàng Nguyên
private void initialize() {
    filteredCart = new FilteredList<Media>(this.cart.getItemsOrdered(), s -> true);

    colMediaTitle.setCellValueFactory(
            new PropertyValueFactory<Media, String>( s: "Title"));
    colMediacategory.setCellValueFactory(
            new PropertyValueFactory<Media, String>( s: "Category"));
    colMediaCost.setCellValueFactory(
            new PropertyValueFactory<Media, Float>( s: "Cost"));
    tblMedia.setItems(this.cart.getItemsOrdered());

    btnPlay.setVisible(false);
    btnRemove.setVisible(false);

    totalCost.setText(String.valueOf(this.cart.totalCost()));

    tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {  Hoàng Nguyên

            @Override  Hoàng Nguyên
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
            }
        }
    );

    tfFilter.textProperty().addListener(new ChangeListener<String>() {  Hoàng Nguyên
        @Override  Hoàng Nguyên
        public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
            showFilteredMedia(newValue);
        }
    });
}
```

Figure 45.CartScreenController 1

```
@FXML  no usages  Hoàng Nguyên
private void btnPlayPressed(ActionEvent event) throws PlayerException {
    // Lấy đối tượng media được chọn
    Media media = this.tblMedia.getSelectionModel().getSelectedItem();

    // Kiểm tra xem media có phải là đối tượng Playable không
    if (media instanceof Playable) {
        try {
            // Ép kiểu media thành Playable và gọi phương thức play
            ((Playable) media).play();
        } catch (Exception e) {
            // Xử lý lỗi phát video hoặc âm thanh (nếu có)
            throw new PlayerException("Error while trying to play the media: " + e.getMessage(), e);
        }
    } else {
        // Nếu media không phải là Playable, hiển thị thông báo lỗi
        throw new PlayerException("Selected media is not playable.");
    }
}
```

Figure 46.CartScreenController 2

6.2 Demo

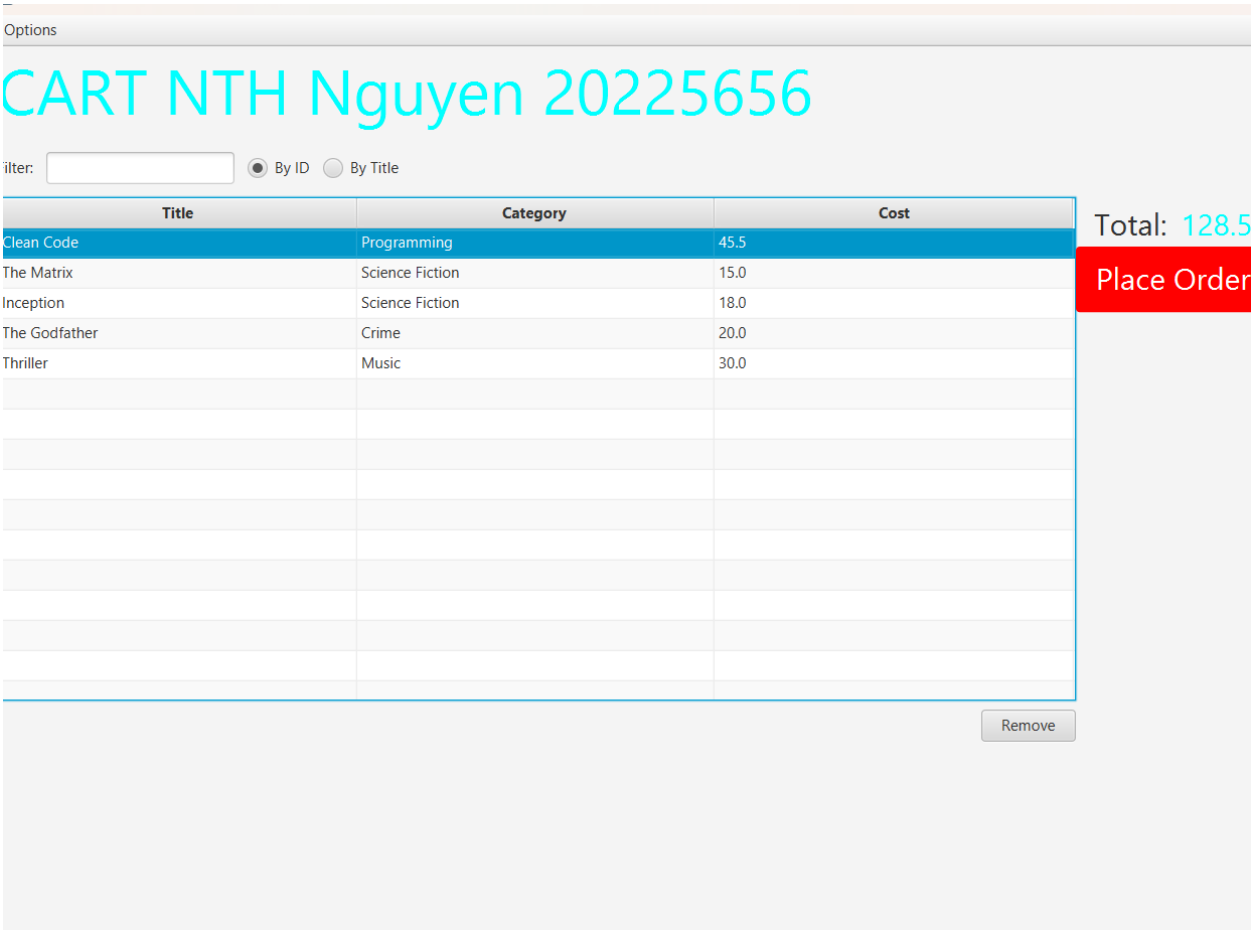

Figure 47.Demo media unplayable

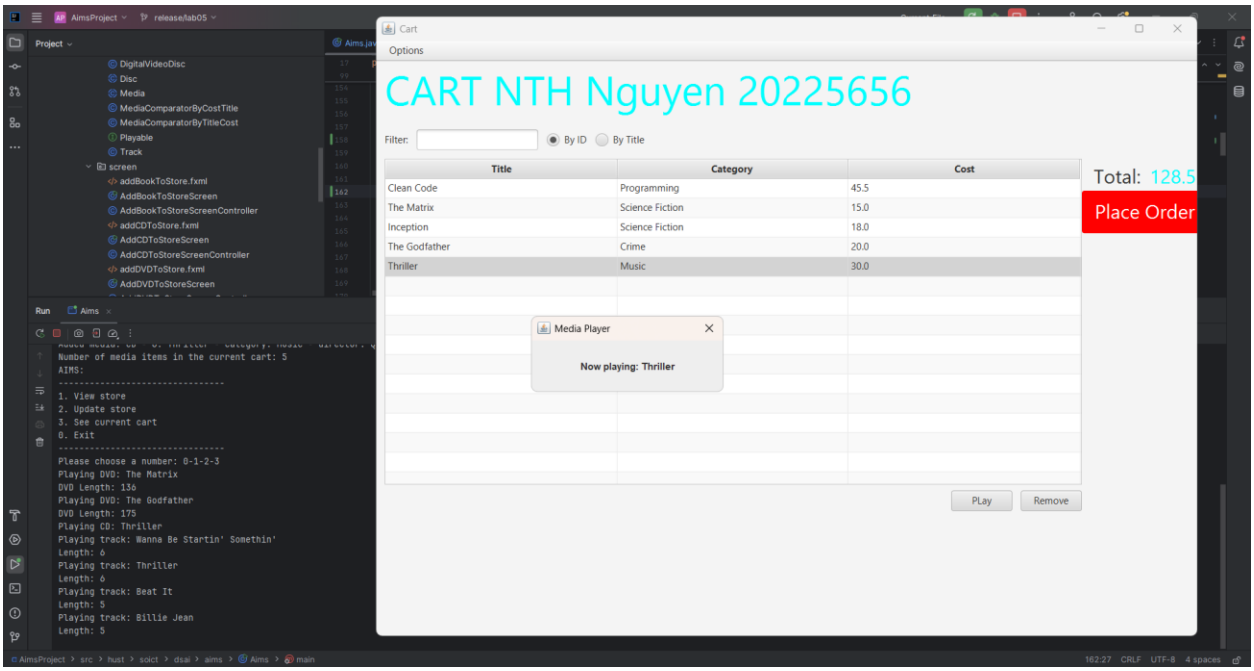

Figure 48.Demo media playable

7. Deleting a media

7.1 Code

```
@FXML no usages  Hoàng Nguyên
private void btnRemovePressed(ActionEvent event) throws NonExistingItemException {
    // Lấy đối tượng media được chọn
    Media media = tblMedia.getSelectionModel().getSelectedItem();

    // Kiểm tra xem có item nào được chọn hay không
    if (media == null) {
        // Nếu không có item nào được chọn, hiển thị thông báo lỗi
        Alert alert = new Alert(Alert.AlertType.WARNING);
        alert.setTitle("Selection Error");
        alert.setHeaderText("No item selected");
        alert.setContentText("Please select an item to remove.");
        alert.showAndWait();
        return; // Dừng phương thức nếu không có item được chọn
    }

    // Thử xóa media khỏi giỏ hàng
    try {
        cart.removeMedia(media);
        // Cập nhật lại tổng chi phí sau khi xóa
        totalCost.setText(String.valueOf(this.cart.totalCost()));
        // Hiển thị thông báo thành công
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Success");
        alert.setHeaderText("Item Removed");
        alert.setContentText(media.getTitle() + " has been removed from the cart.");
        alert.showAndWait();
    } catch (NonExistingItemException e) {
        // Nếu media không tồn tại trong giỏ hàng, ném ngoại lệ
        throw new NonExistingItemException("Item does not exist in the cart.");
    }
}
```

Figure 49.btnRemovePressed Method

7.2 Demo

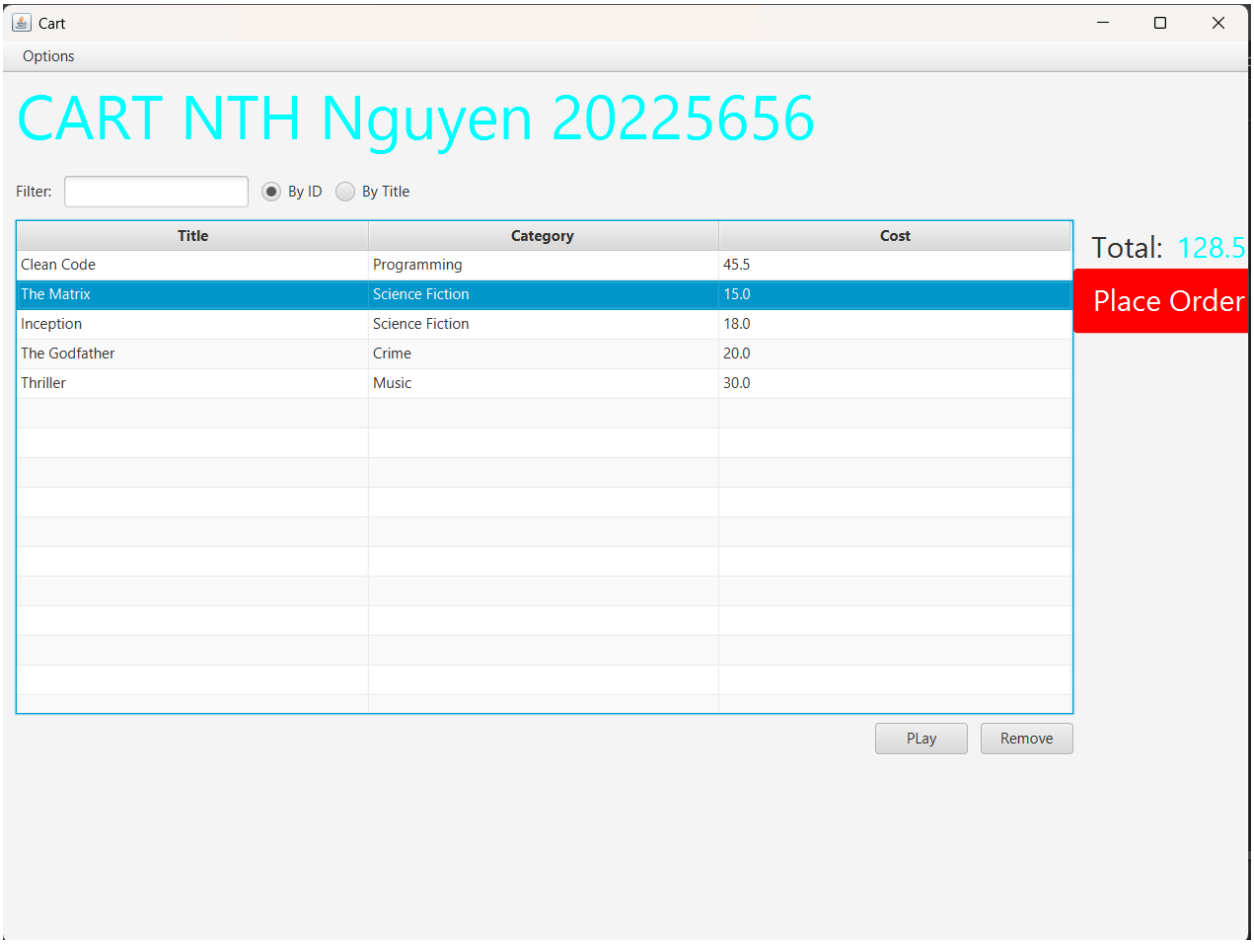

Figure 50.button Remove 1

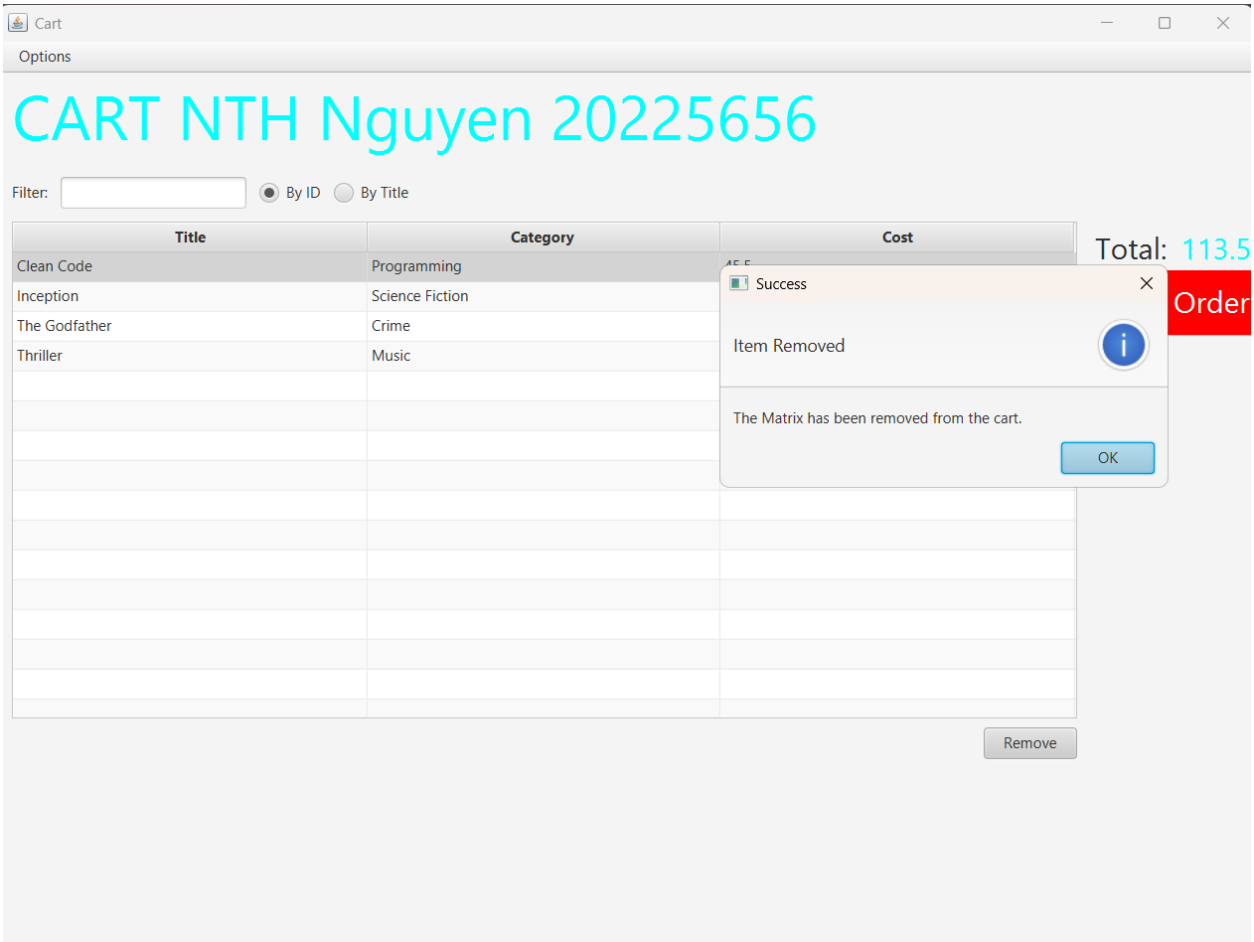

Figure 51.button Remove 2

8. Complete the Aims GUI application

Figure 52.Store before add book

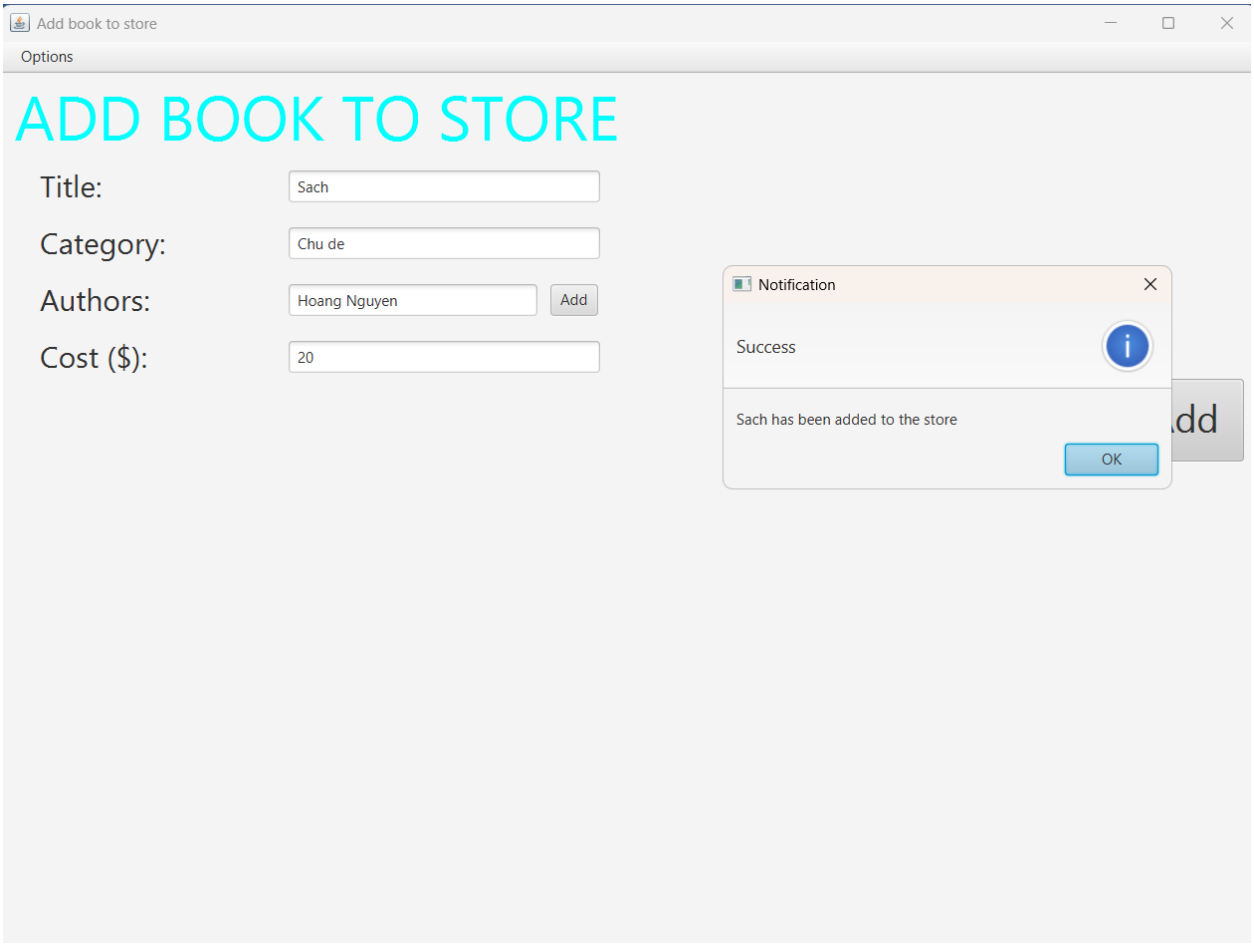

Figure 53.Add book

Figure 54.Store after add book

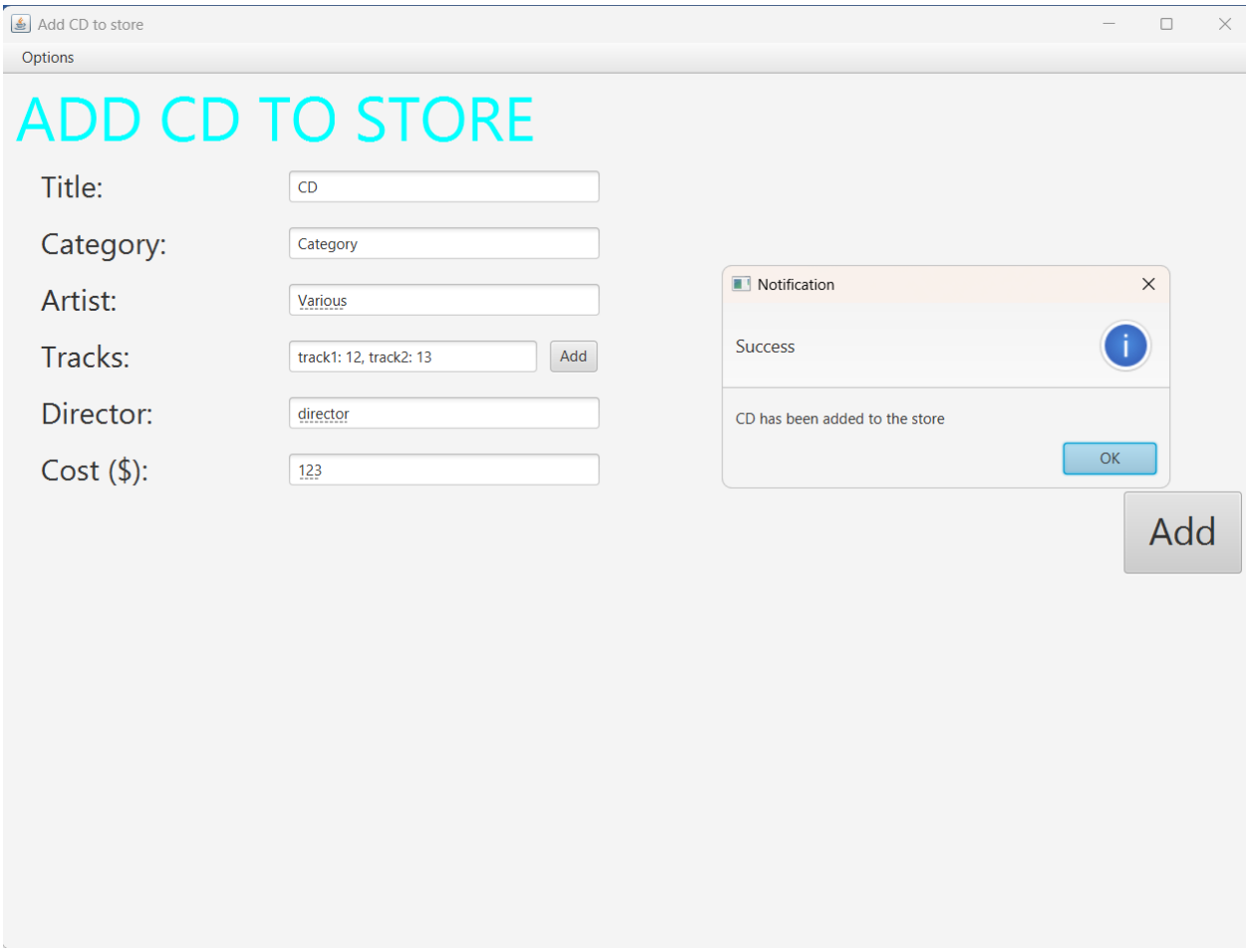

Figure 55.Add CD

Figure 56.Store after add CD

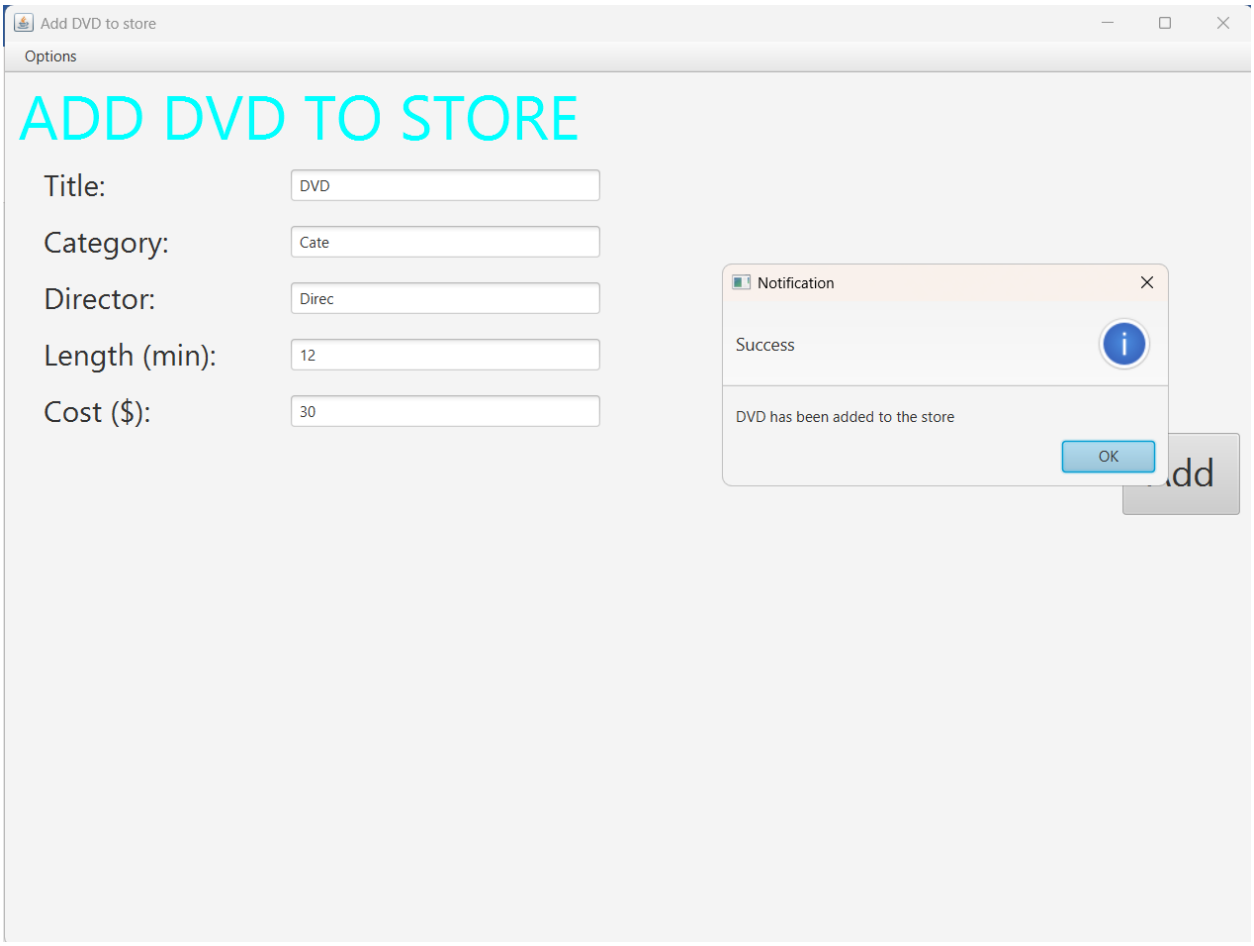

Figure 57.Add DVDFigure 58.Store after add DVD


Store
Options
AIMS:Nguyen Trinh Hoang Nguyen 2022565
The Matrix
15.0 $
Add to cart
Play
Inception
18.0 $
Add to cart
Play
The Godfather
20.0 $
Add to cart
Play
Pulp Fiction
16.0 $
Add to cart
Play
The Lord of the Rings: The Fellows...
25.0 $
Add to cart
Play
Thriller
30.0 $
Add to cart
Play
Back in Black
25.0 $
Add to cart
Play
Abbey Road
35.0 $
Add to cart
Play
The Dark Side of the Moon
40.0 $
Add to cart
Play
Rumours
28.0 $
Add to cart
Play
Sach
20.0 $
Add to cart
CD
123.0 $
Add to cart
Play
DVD
30.0 $
Add to cart
Play

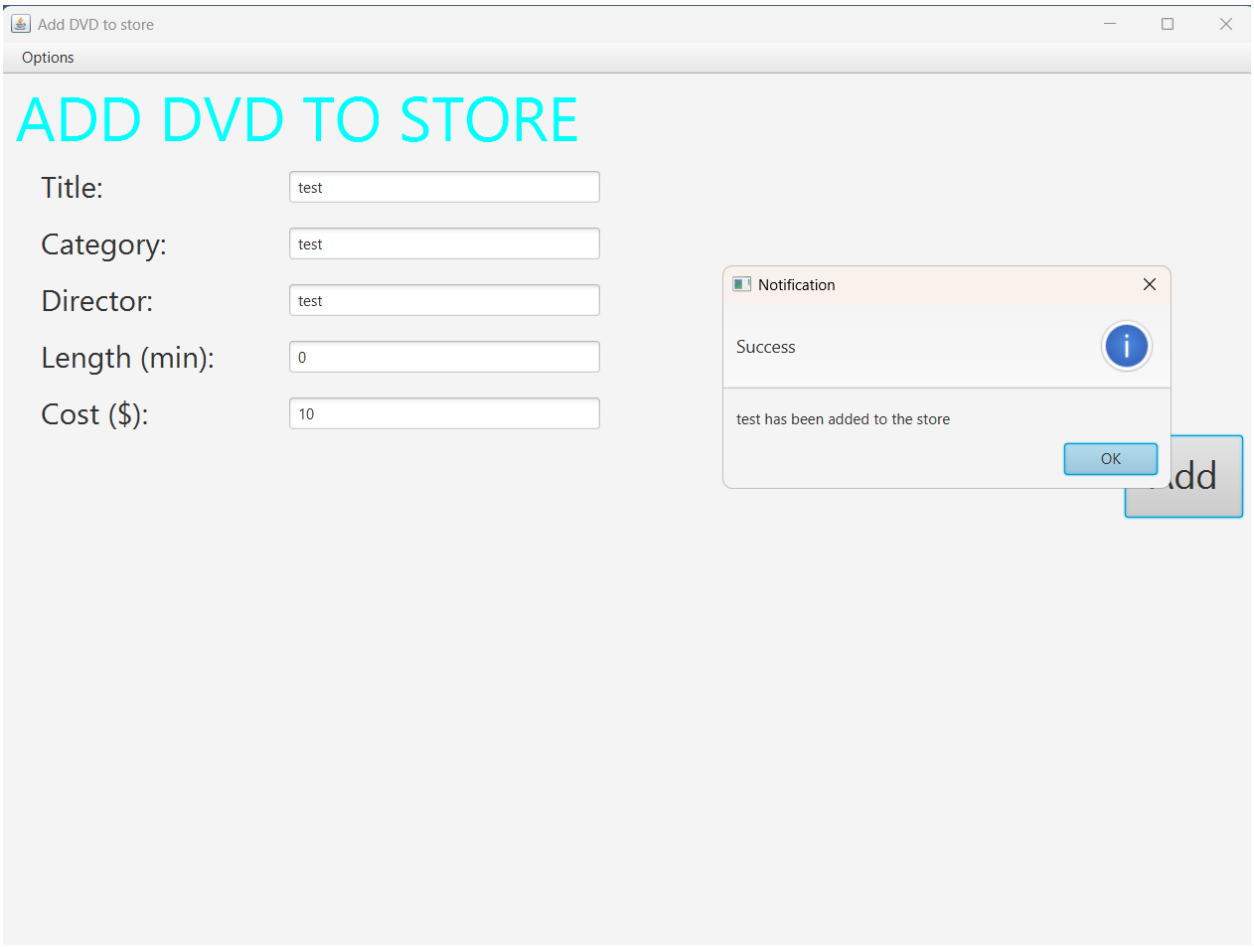

Figure 59.Test exception with DVD Length = 0

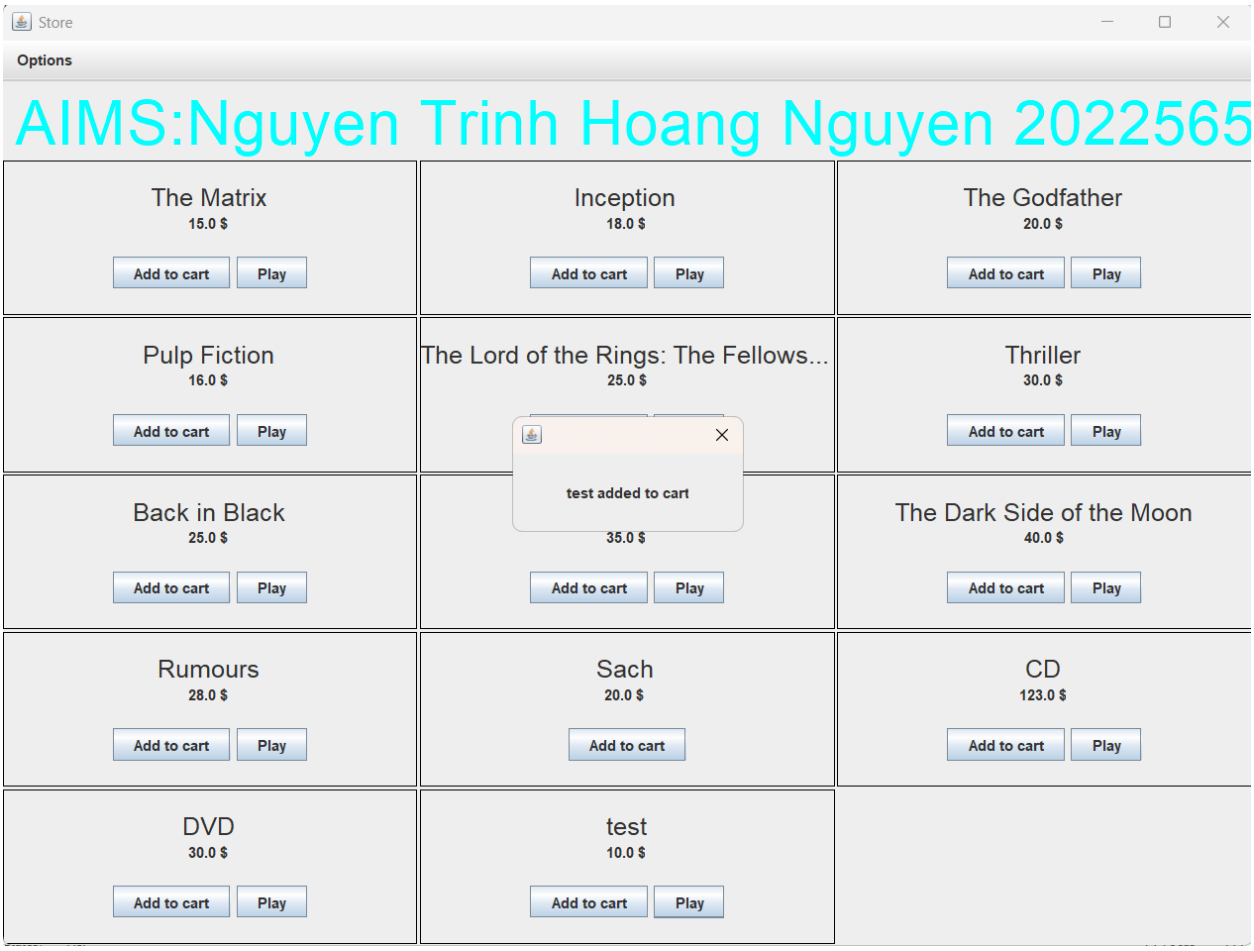

Figure 60.Add test to Cart

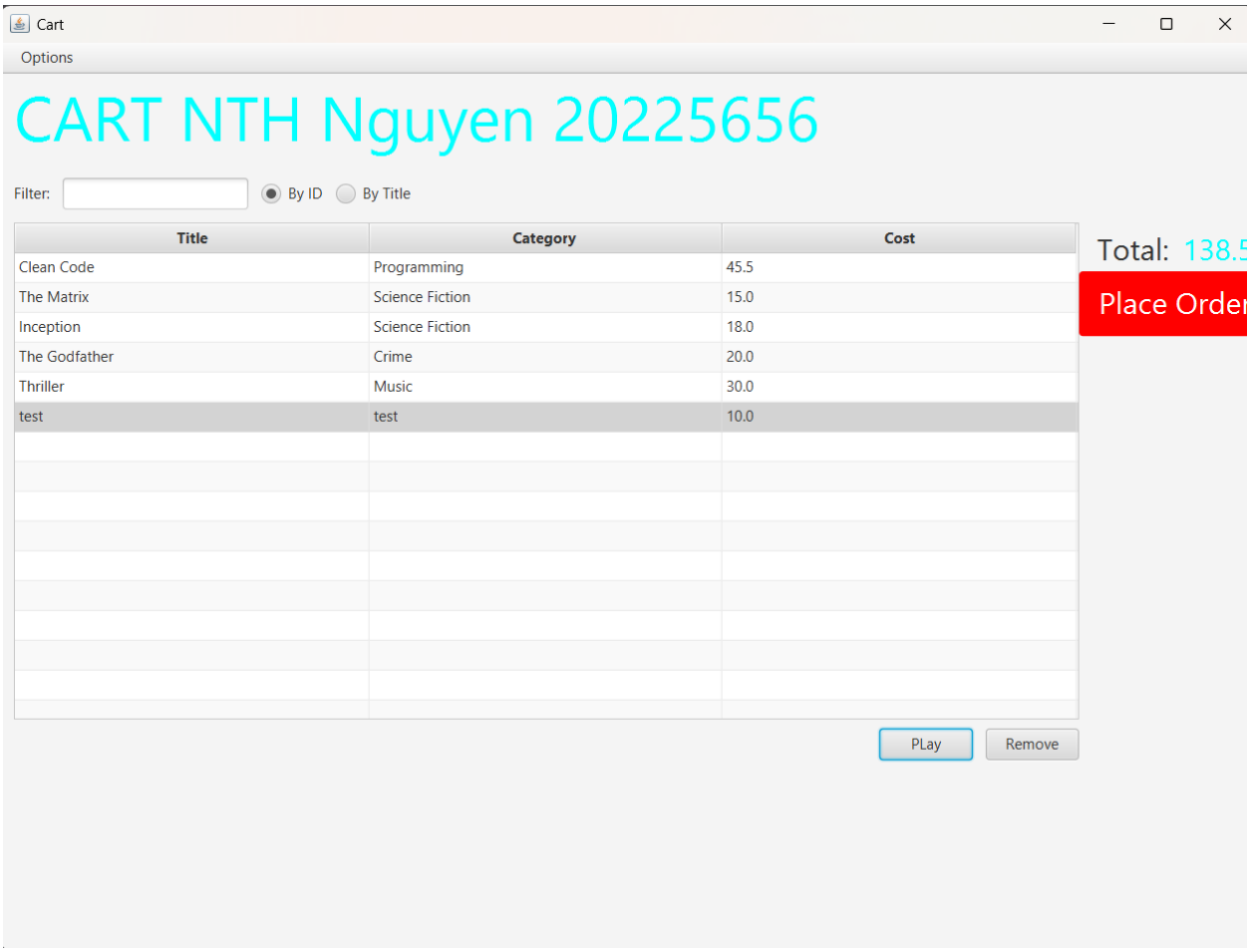

Figure 61.play test

```
Caused by: hust.soict.dsai.aims.exception.PlayerException Create breakpoint : Error while trying to play the media: DVD Length is zero
    at AimsProject/hust.soict.dsai.aims.screen.CartScreenController.btnPlayPressed(CartScreenController.java:193)
    at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)
    ... 56 more
Caused by: hust.soict.dsai.aims.exception.PlayerException Create breakpoint : DVD Length is zero
    at AimsProject/hust.soict.dsai.aims.media.DigitalVideoDisc.play(DigitalVideoDisc.java:52)
    at AimsProject/hust.soict.dsai.aims.screen.CartScreenController.btnPlayPressed(CartScreenController.java:190)
    ... 57 more
```

Figure 62.Exception

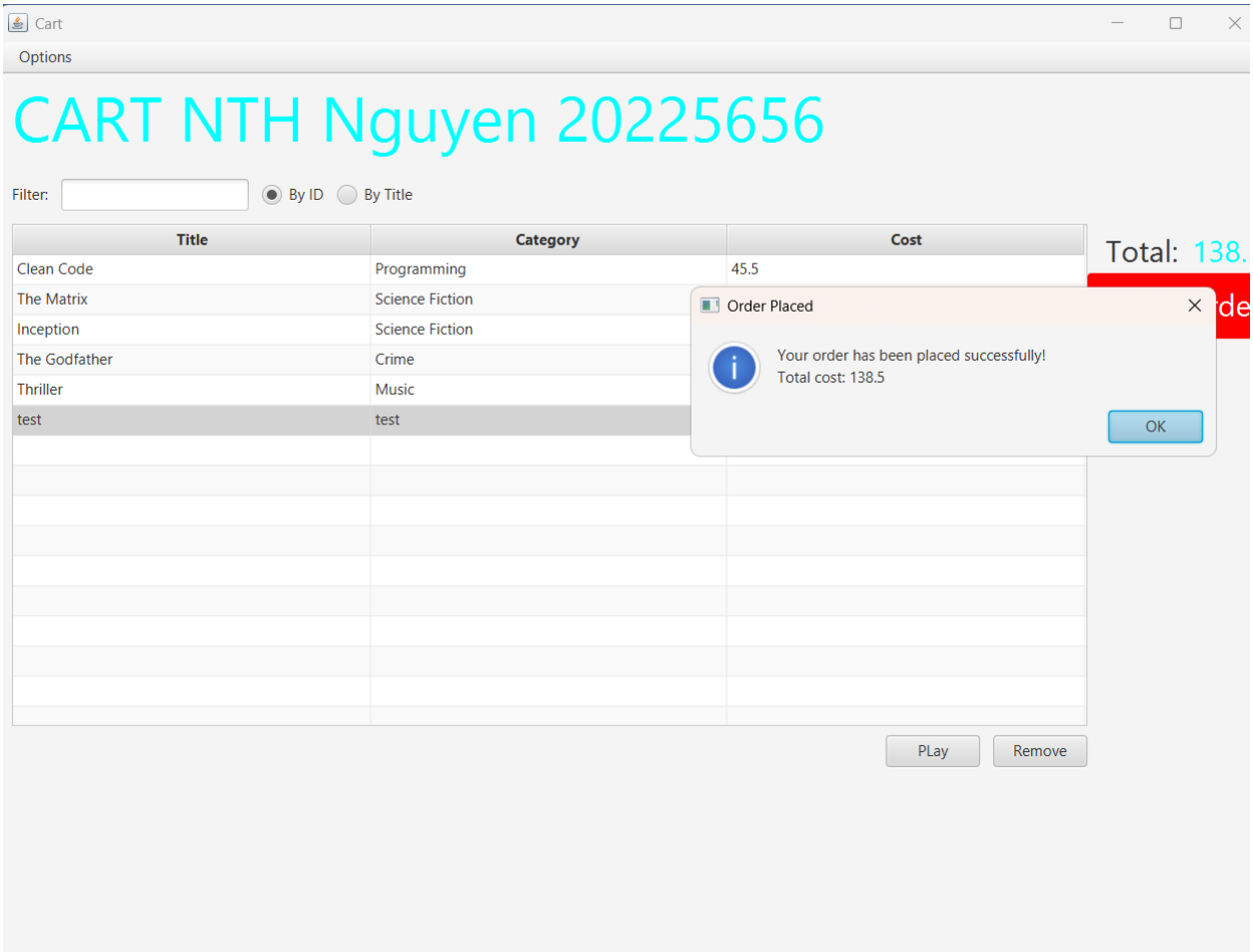

Figure 63.Place Order

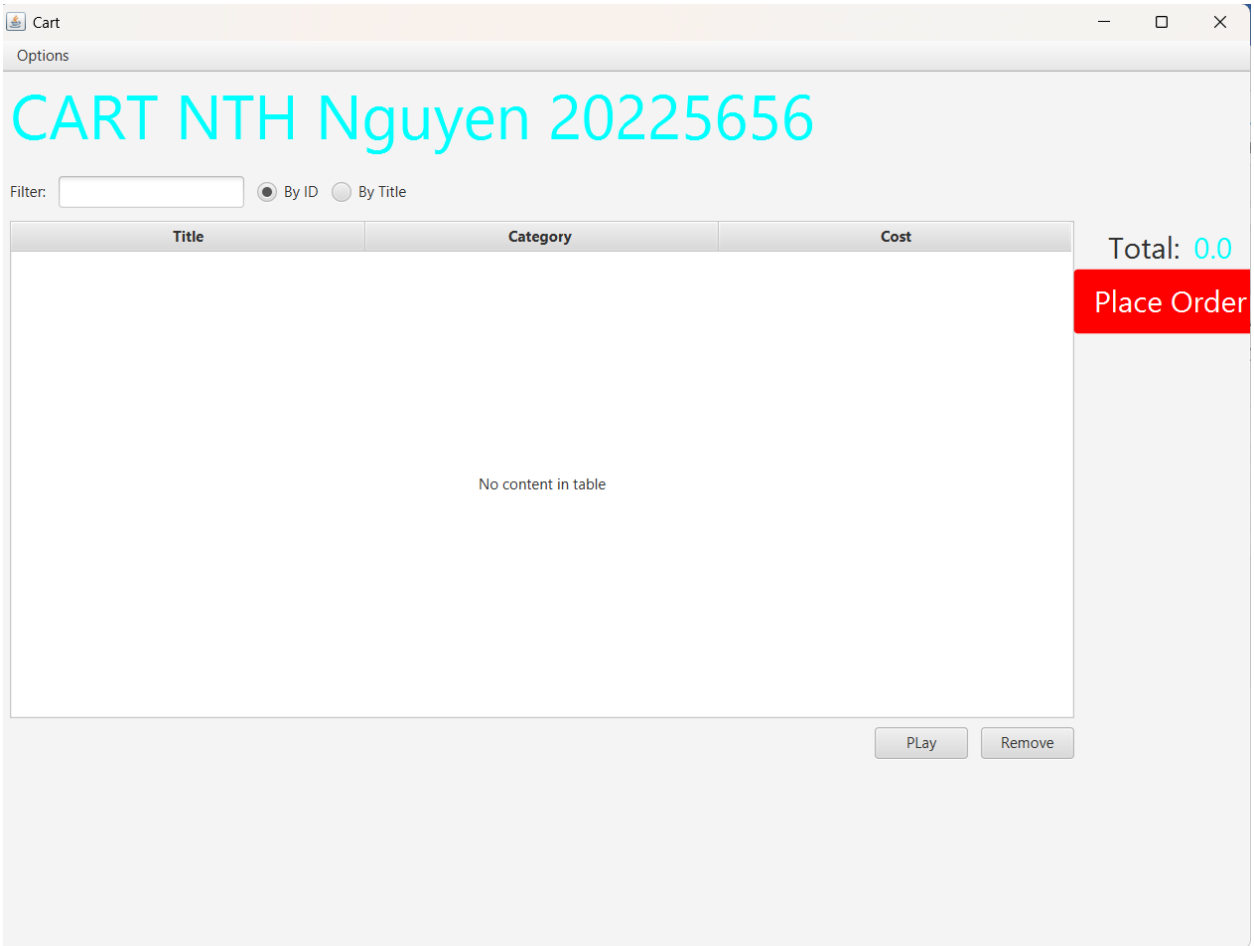

Figure 64.After Place Order

9. equals() method in Media class

```
@Override  Hoàng Nguyên
public boolean equals(Object object){
    if (object instanceof Media) {
        try {
            Media that = (Media) object;
            return this.title.equalsIgnoreCase(that.getTitle());
        } catch (NullPointerException | ClassCastException e1) {
            return false;
        }
    } else {
        return false;
    }
}
```

Figure 65.equals() method

10. Use case Diagram

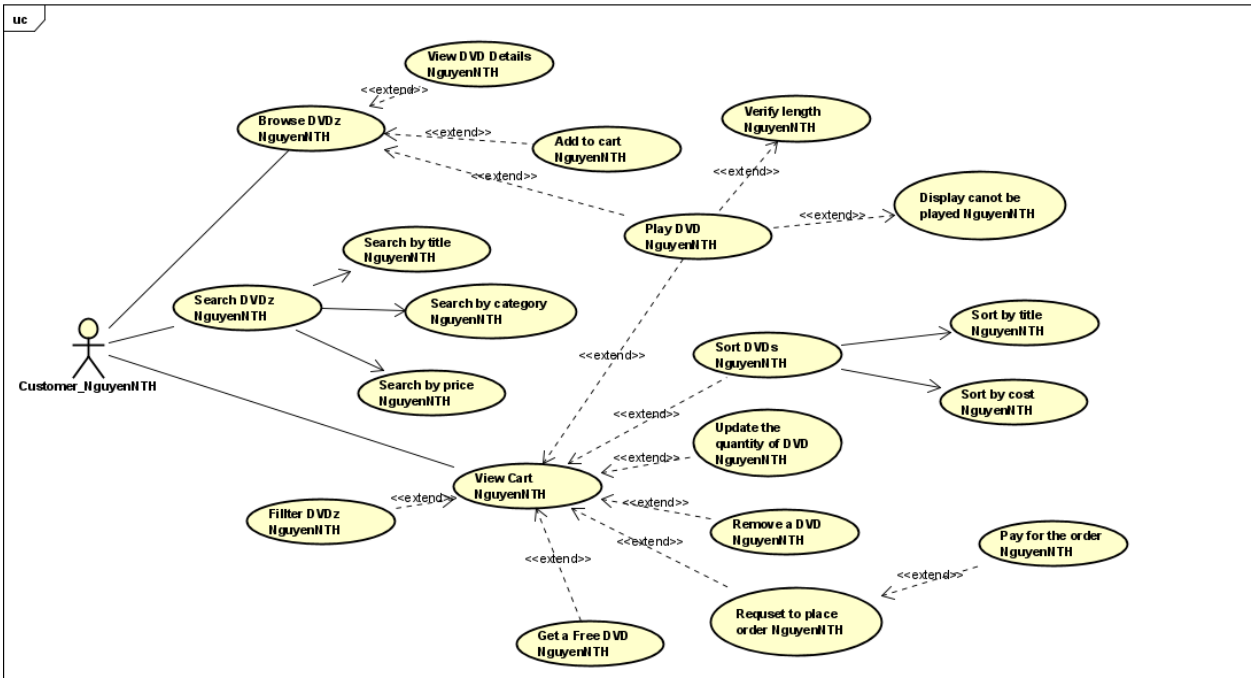

Figure 66.UC Diagram 1

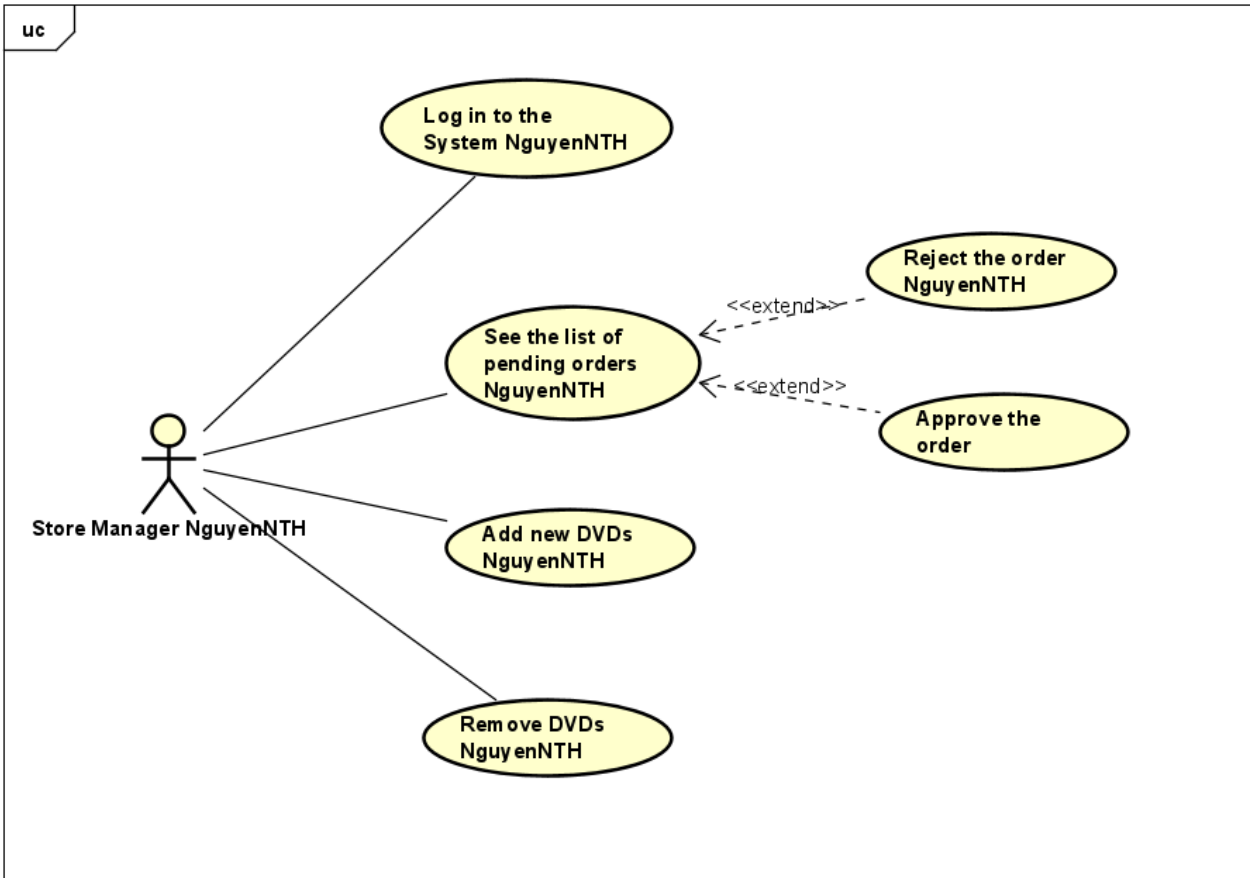

Figure 67..UC Diagram 2

11. Class Diagram

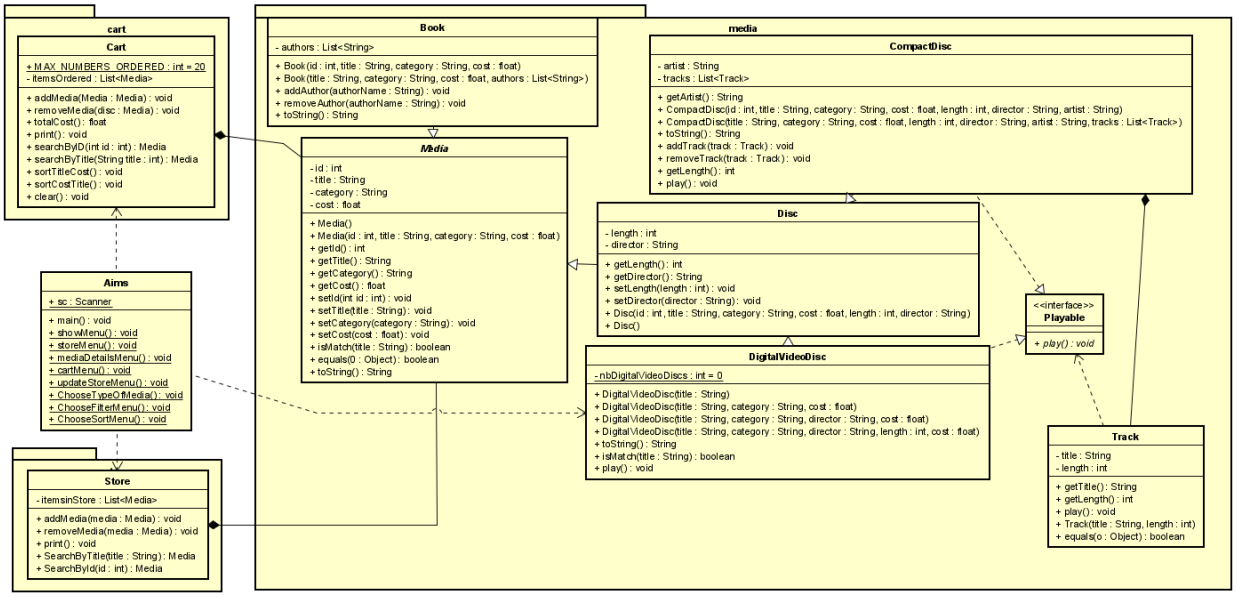

Figure 68.Class Diagram