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

BÁO CÁO THỰC HÀNH
IT1130-744528-2024.1

BÀI THỰC HÀNH 05

Họ và tên sv: Phạm Quốc Cường
Lớp: **K67-Việt Nhật 01**
GVHD: Lê Thị Hoa
TA: **Phạm Mạnh Cường**

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB
5 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. Swing components

1.1 AWTAccumulator

```java
//Pham Quoc Cuong - 20225604
package hust.soict.hedspi.swing;
import java.awt.Label;
import java.awt.GridLayout;
import java.awt.TextField;
import java.awt.Frame;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class AWTAccumulator extends Frame {
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;

    public AWTAccumulator(){
        setLayout(new GridLayout(2,2));
        add(new Label("CuongPQ 5604 - Enter an Integer: "));

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

        add(new Label("CuongPQ 5604 - The Accumulated Sum is: "));

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

        setTitle("CuongPQ 5604 - AWT Accumulator");
        setSize(350, 120);
        setVisible(true);
    }

    public static void main(String[] args){
        new AWTAccumulator();
    }
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent ewt){
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(sum + "");
        }
    }
}
```

Figure 1.1: Source code of AWTAccumulator

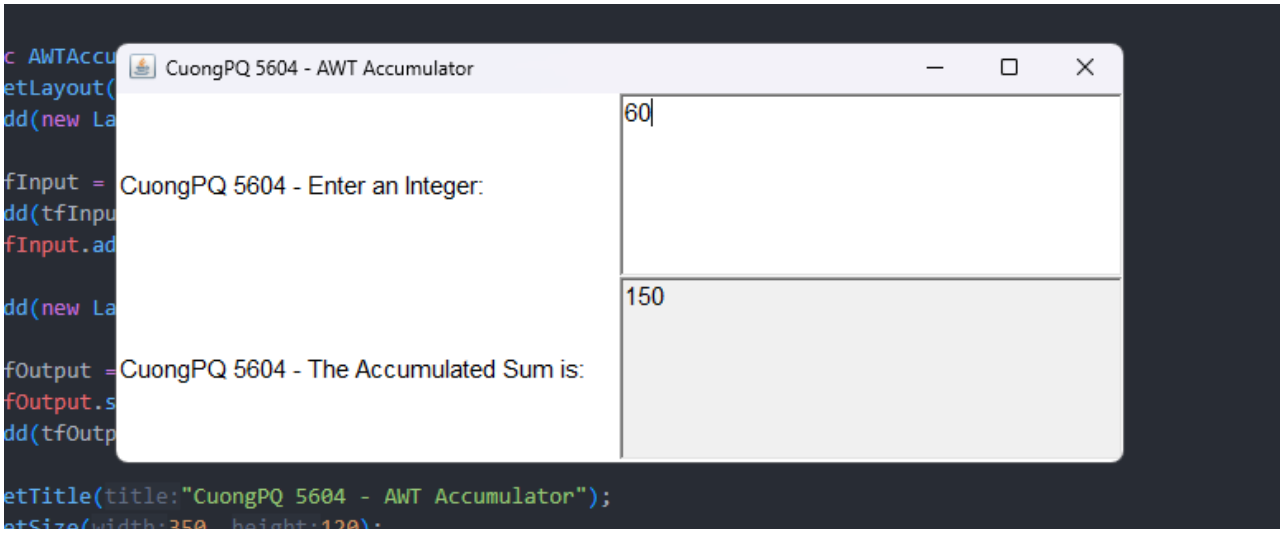


Figure 1.2: Demo of AWTAccumulator

1.2 SwingAccumulator

```java
// Pham Quoc Cuong - 20225604
package hust.soict.hedspi.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;
    private JTextField tfOutput;
    private int sum = 0;

    public SwingAccumulator() {
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(2,2));

        cp.add(new Label("CuongPQ 5604 - Enter an Integer: "));
        tfInput = new JTextField(10);
        cp.add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        cp.add(new Label("CuongPQ 5604 -The Accumulated Sum is: "));
        tfOutput = new JTextField(10);
        tfOutput.setEditable(false);
        cp.add(tfOutput);

        setTitle("CuongPQ 5604 - Swing Accumulator");
        setSize(350, 120);
        setVisible(true);
    }

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

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

Figure 1.3: Source code of SwingAccumulator

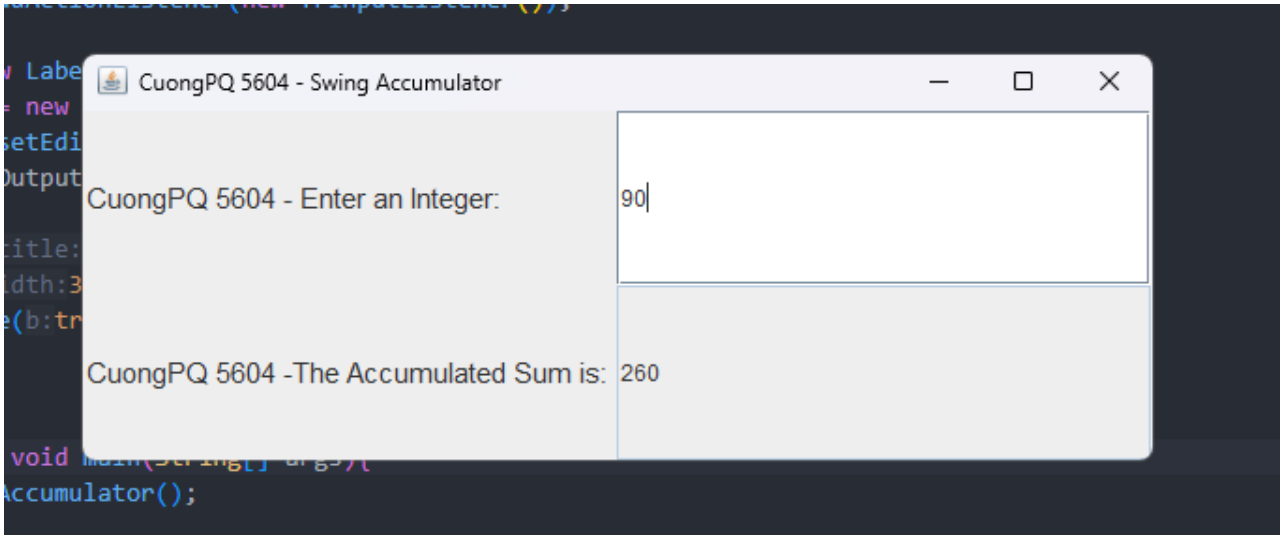


Figure 1.4: Demo of SwingAccumulator

2 Organizing Swing components with Layout Managers

2.1 Code

```java
// Pham Quoc Cuong - 20225604
package hust.soict.hedspi.swing;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class NumberGrid extends JFrame {
    private JButton[] btnNumbers = new JButton[10];
    private JButton btnDelete, btnReset;
    private JTextField tfDisplay;

    public NumberGrid() {
        tfDisplay = new JTextField();
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
        JPanel panelButtons = new JPanel(new GridLayout(4,3));
        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("CuongPQ 5604 - Number Grid");
        setSize(200,200);
        setVisible(true);
    }

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

Figure 2.1: Source code of NumberGrid 1

```
36
37          btnDelete = new JButton("DEL");
38          panelButtons.add(btnDelete);
39          btnDelete.addActionListener(btnListener);
40
41          btnNumbers[0] = new JButton("0");
42          panelButtons.add(btnNumbers[0]);
43          btnNumbers[0].addActionListener(btnListener);
44
45          btnReset = new JButton("C");
46          panelButtons.add(btnReset);
47          btnReset.addActionListener(btnListener);
48
49      }
50
51      private class ButtonListener implements ActionListener {
52          @Override
53          public void actionPerformed(ActionEvent e) {
54              String button = e.getActionCommand();
55              if(button.charAt(0) >= '0'  && button.charAt(0) <= '9') {
56                  tfDisplay.setText(tfDisplay.getText() + button);
57              }
58              else if (button.equals("DEL")) {
59
60                  tfDisplay.setText(tfDisplay.getText().substring(0,tfDisplay.getText().length() -1));
61              }
62              else {
63                  tfDisplay.setText("");
64              }
65          }
66      }
67
68      public static void main(String[] args) {
69          new NumberGrid();
70      }
71  }
```

Figure 2.2: Source code of NumberGrid 2

2.2 Demo

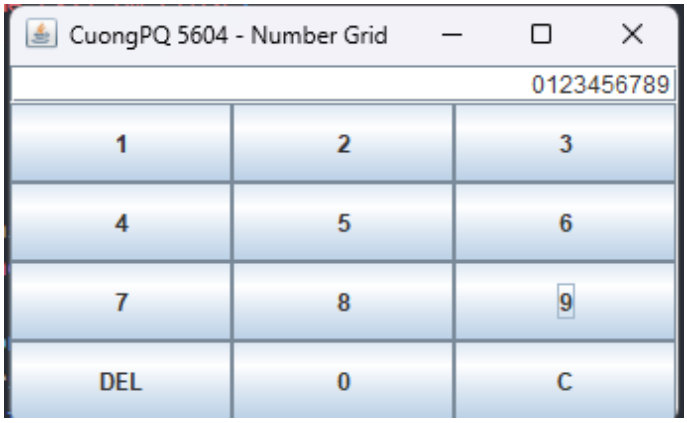


Figure 2.3: Demo buttons 0-9

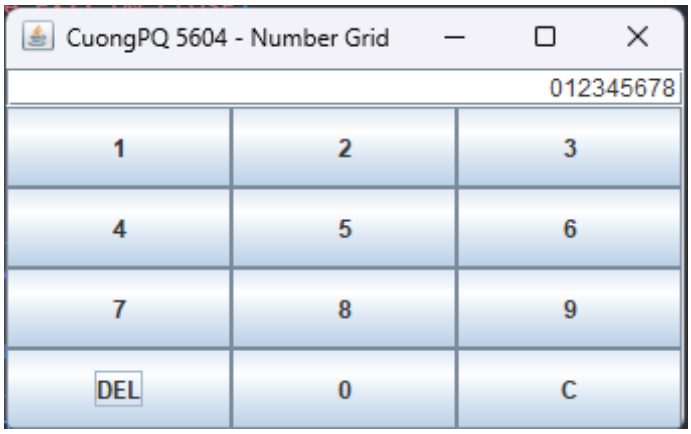


Figure 2.4: Demo DEL button

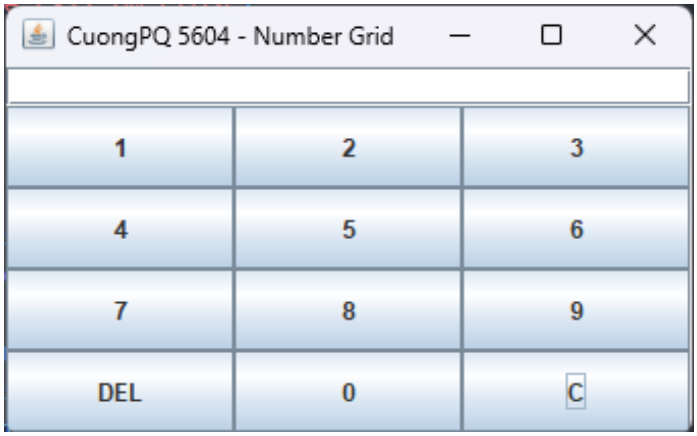


Figure 2.5: Demo C button

3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

```java
package hust.soict.dsai.aims.screen;
import javax.swing.*;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.*;
import hust.soict.dsai.aims.store.Store;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

public class StoreScreen extends JFrame {
    private Store store;
    private Container cp;
    private Cart cart;
    JPanel createNorth() {
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north,BoxLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }

    JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Options");
        JMenu smUpdateStore = new JMenu("CuongPQ 225604 - Update Store");
        JMenuItem addBook = new JMenuItem("CuongPQ 225604 - Add Book");
        smUpdateStore.add(addBook);
        addBook.addActionListener(e -> {
            new AddBookStoreScreen(store);
        });
        JMenuItem addCD = new JMenuItem("CuongPQ 225604 - Add CD");
        smUpdateStore.add(addCD);
        addCD.addActionListener(e -> {
            new AddCDStoreScreen(store);
        });
        JMenuItem addDVD = new JMenuItem("CuongPQ 225604 - Add DVD");
        smUpdateStore.add(addDVD);
        addDVD.addActionListener(e -> {
            new AddDVDStoreScreen(store);
        });
        menu.add(smUpdateStore);
        JMenuItem viewCart = new JMenuItem("View cart");
        viewCart.addActionListener(e -> {
            new CartScreen(cart);
        });
        menu.add(viewCart);
        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);
        return menuBar;
    }
```

Figure 3.1: Class StoreScreen 1

```java
JPanel createHeader() {
    JPanel header = new JPanel();
    header.setLayout(new BoxLayout(header,BoxLayout.X_AXIS));

    JLabel title = new JLabel("CuongPQ 225604 - AIMS");
    title.setFont(new Font(title.getFont().getName(),Font.PLAIN,50));
    title.setForeground(Color.BLACK);

    JButton cart1 = new JButton("View cart");
    cart1.setPreferredSize(new Dimension(100,50));
    cart1.setMaximumSize(new Dimension(100,50));
    cart1.addActionListener(e -> {
        new CartScreen(cart);
    });

    header.add(Box.createRigidArea(new Dimension(10,10)));
    header.add(title);
    header.add(Box.createHorizontalGlue());
    header.add(cart1);
    header.add(Box.createRigidArea(new Dimension(10,10)));
    return header;
}

JPanel createCenter() {
    JPanel center = new JPanel();
    center.setLayout(new GridLayout(3,3,2,2));
    ArrayList<Media> mediaStore = store.getItemsInStore();
    for(Media media : mediaStore) {
        MediaStore cell = new MediaStore(media,cart);
        center.add(cell);
    }
    return center;
}

public StoreScreen(Store store, Cart myCart) {
    this.store  = store;
    this.cart = myCart;
    cp = getContentPane();
    cp.setLayout(new BorderLayout());
    cp.add(createNorth(),BorderLayout.NORTH);
    cp.add(createCenter(),BorderLayout.CENTER);
    setVisible(true);
    setTitle("20225604 - Store");
    setSize(1024,768);
}
```

Figure 3.2: Class StoreScreen 2

```java
private class AddBookStoreScreen extends JFrame {
    public AddBookStoreScreen(Store store) {
        this.setLayout(new GridLayout(5, 2, 5, 5));
        this.add(new JLabel("Enter title: "));
        TextField title = new TextField(10);
        this.add(title);
        this.add(new JLabel("Enter category: "));
        TextField category = new TextField(10);
        this.add(category);
        this.add(new JLabel("Enter cost: "));
        TextField cost = new TextField(10);
        this.add(cost);
        this.add(new JLabel("Enter authors: "));
        TextField authors = new TextField(10);
        this.add(authors);

        this.setTitle("Add Book");
        this.setSize(300, 150);
        JButton turnInBtn = new JButton("Add");
        this.add(turnInBtn);
        turnInBtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                ArrayList<String> authorList = new ArrayList<>();
                for (String author : authors.getText().split(",")) {
                    authorList.add(author.trim());
                }
                Book book = new Book(title.getText(), category.getText(), Float.parseFloat(cost.getText()), authorList);
                store.addMedia(book);
                cp.add(createCenter(), BorderLayout.CENTER);
                cp.revalidate();
                title.setText("");
                category.setText("");
                cost.setText("");
                authors.setText("");
            }
        });
        setVisible(true);
    }
}
```

Figure 3.3: Class StoreScreen 3

```java
private class AddDVDStoreScreen extends JFrame {
    public AddDVDStoreScreen(Store store) {
        this.setLayout(new GridLayout(4, 2, 5, 5));
        this.add(new JLabel("Enter title: "));
        TextField title = new TextField(10);
        this.add(title);
        this.add(new JLabel("Enter category: "));
        TextField category = new TextField(10);
        this.add(category);
        this.add(new JLabel("Enter cost: "));
        TextField cost = new TextField(10);
        this.add(cost);

        this.setTitle("Add DVD");
        this.setSize(300, 100);
        JButton turnInBtn = new JButton("Add");
        this.add(turnInBtn);
        turnInBtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                DigitalVideoDisc dvd = new DigitalVideoDisc(title.getText(), category.getText(), Float.parseFloat(cost.getText()));
                store.addMedia(dvd);
                cp.add(createCenter(), BorderLayout.CENTER);
                cp.revalidate();
                title.setText("");
                category.setText("");
                cost.setText("");
            }
        });
        setVisible(true);
    }
}
```

Figure 3.4: Class StoreScreen 4

```java
private class AddCDStoreScreen extends JFrame {
    public AddCDStoreScreen(Store store) {
        this.setLayout(new GridLayout(7, 2, 5, 5));
        this.add(new JLabel("Enter title: "));
        TextField title = new TextField(10);
        this.add(title);
        this.add(new JLabel("Enter category: "));
        TextField category = new TextField(10);
        this.add(category);
        this.add(new JLabel("Enter cost: "));
        TextField cost = new TextField(10);
        this.add(cost);
        this.add(new JLabel("Enter artist: "));
        TextField artist = new TextField(10);
        this.add(artist);
        this.setTitle("Add CD");
        this.add(new JLabel("Number of tracks: "));
        TextField numberOfTracks = new TextField(10);
        this.add(numberOfTracks);
        this.pack();
        JButton turnInBtn = new JButton("Add");
        this.add(turnInBtn);
        turnInBtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                CompactDisc cd = new CompactDisc(title.getText(), category.getText(), artist.getText(), new ArrayList<Track>(), Float.parseFloat(cost.getText()));
                store.addMedia(cd);
                cp.add(createCenter(), BorderLayout.CENTER);
                cp.revalidate();
                title.setText("");
                category.setText("");
                cost.setText("");
            }
        });
        setVisible(true);
    }
}
```

Figure 3.5: Class StoreScreen 5

```java
    public static void main(String[] args) {
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Disney", 15, 89);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Inception", "Sci-Fi", "Christopher Nolan", 20, 148);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("The Dark Knight", "Action", "Christopher Nolan", 25, 152);

        ArrayList<Track> tracks1 = new ArrayList<Track>();
        tracks1.add(new Track("Bohemian Rhapsody", 6));
        tracks1.add(new Track("Stairway to Heaven", 8));
        CompactDisc cd1 = new CompactDisc("Classic Rock", "Rock", "Various Artists", tracks1, 25.0f);

        ArrayList<Track> tracks2 = new ArrayList<Track>();
        tracks2.add(new Track("Blinding Lights", 3));
        tracks2.add(new Track("Save Your Tears", 4));
        tracks2.add(new Track("In Your Eyes", 4));
        CompactDisc cd2 = new CompactDisc("The Weeknd Hits", "Pop", "The Weeknd", tracks2, 30.0f);

        ArrayList<String> authors1 = new ArrayList<String>();
        authors1.add("J.K. Rowling");
        Book book1 = new Book("Harry Potter and the Philosopher's Stone", "Fantasy", 20.0f, authors1);
        tracks2.add(new Track("Deamn - Sign", 4));
        tracks2.add(new Track("MBB - Arrival", 3));
        tracks2.add(new Track("EnV - Pneumatic", 5));
        CompactDisc cd3 = new CompactDisc("Addictive EDM Music", "International Music", "Various Artists", tracks2, 37.25f);

        ArrayList<String> authors3 = new ArrayList<String>();
        authors3.add("Agatha Christie");
        Book book3 = new Book("And Then There Were None", "Detective", 25.2f, authors1);

        ArrayList<String> authors2 = new ArrayList<String>();
        authors2.add("Phung Quan");
        Book book2 = new Book("The Savage Childhood", "Novel", 27.45f, authors2);

        Store store = new Store();

        store.addMedia(cd1);
        store.addMedia(cd2);
        store.addMedia(cd3);
        store.addMedia(dvd1);
        store.addMedia(dvd2);
        store.addMedia(dvd3);
        store.addMedia(book1);
        store.addMedia(book2);
        store.addMedia(book3);


        Cart myCart = new Cart();
        new StoreScreen(store, myCart);
    }
}
```

Figure 3.6: Class StoreScreen 6

3.2 Create class MediaStore

```java
// Pham Quoc Cuong - 20225604
package hust.soict.dsai.aims.screen;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.*;
import javax.swing.*;
import java.awt.*;

public class MediaStore extends JPanel {
    private Media media;

    public MediaStore(Media media, Cart myCart) {
        this.media = media;
        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

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

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

        JPanel container = new JPanel();
        container.setLayout(new FlowLayout(FlowLayout.CENTER));
        JButton btnAdd = new JButton("Add to cart");
        container.add(btnAdd);
        btnAdd.addActionListener(e -> myCart.addMedia(media));

        if(media instanceof Playable) {
            JButton playBtn= new JButton("Play");
            playBtn.addActionListener(e -> {
                JDialog playDialog = createPlayDialog(media);
                playDialog.setVisible(true);
                playDialog.setSize(300,200);
                playDialog.pack();
            });
            container.add(playBtn);
        }
        this.add(Box.createVerticalGlue());
        this.add(title);
        this.add(cost);
        this.add(Box.createVerticalGlue());
        this.add(container);
        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }
```

Figure 3.7: Class MediaStore 1

```
1
2       static JDialog createPlayDialog(Media media) {
3           JDialog playDialog = new JDialog();
4           Container container = playDialog.getContentPane();
5           playDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
6           container.setLayout(new BoxLayout(container,BoxLayout.Y_AXIS));
7           container.add(Box.createRigidArea(new Dimension(10,10)));
8           if(media instanceof DigitalVideoDisc dvd) {
9               container.add(new JLabel("Playing DVD:" + dvd.getTitle()));
10              container.add(new JLabel("DVD length:" + dvd.getLength() +" min"));
11          } else if (media instanceof CompactDisc cd) {
12              container.add(new JLabel("Title: " + cd.getTitle()));
13              container.add(new JLabel("Artist: " + cd.getArtist()));
14              for (Track track : cd.getTracks()) {
15                  container.add(new JLabel("Play: " + track.getTitle() + ". Length: " + track.getLength() + " min"));
16              }
17          }
18          playDialog.setTitle("Play " + media.getTitle());
19          return playDialog;
20      }
21  }
```

Figure 3.8: Class MediaStore 2

3.3 Demo

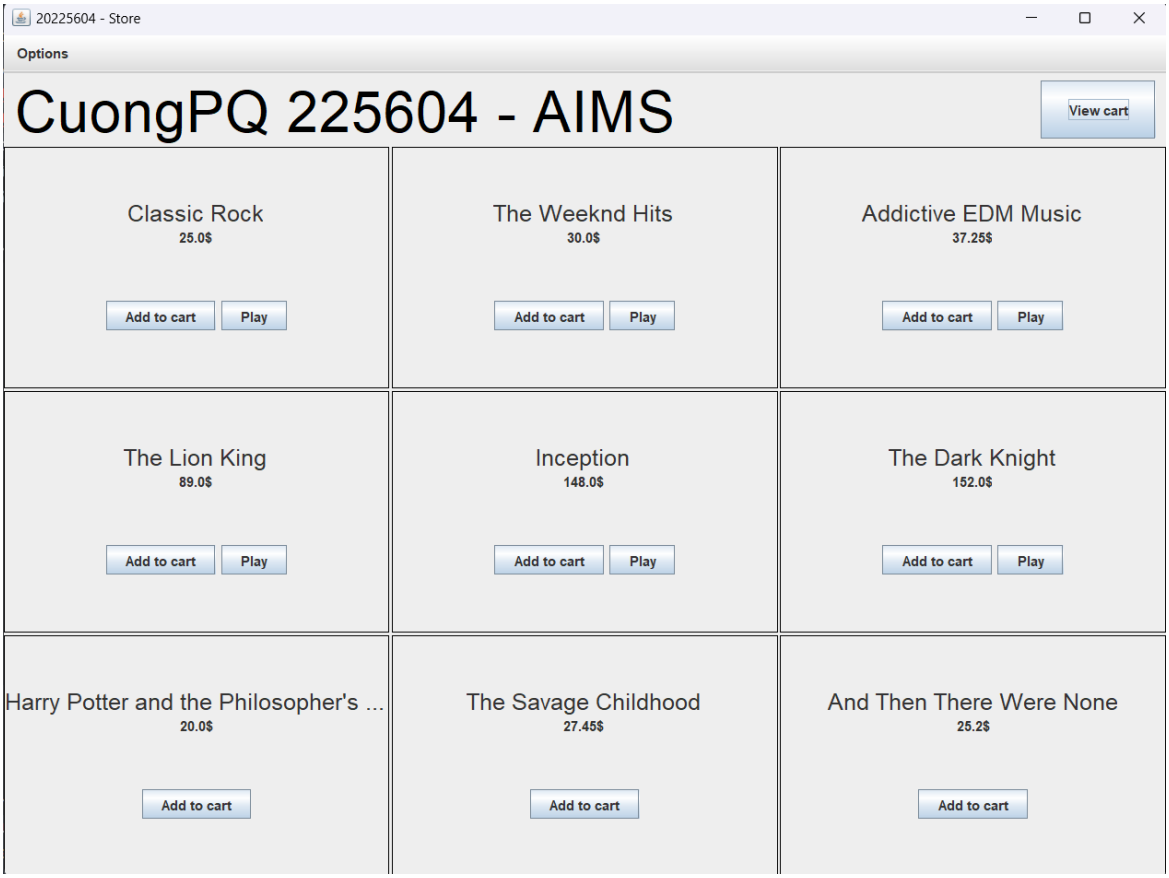


Figure 3.10: StoreScreen

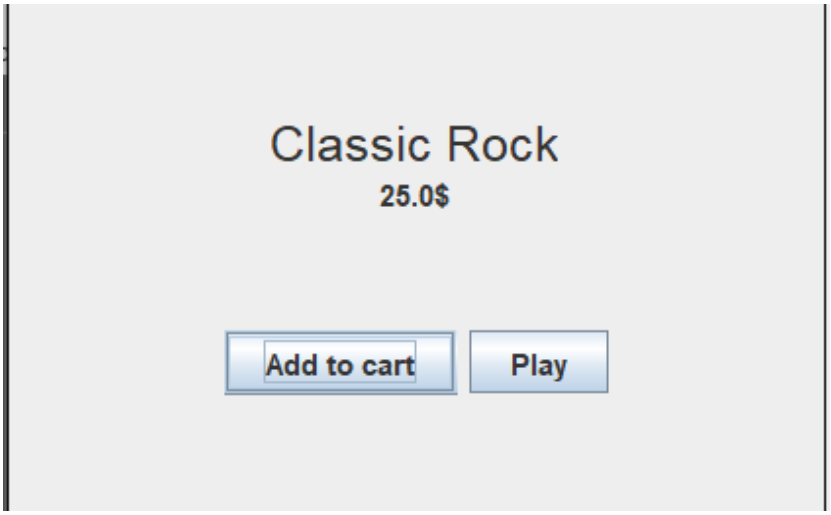


Figure 3.11 Demo Add to cart button

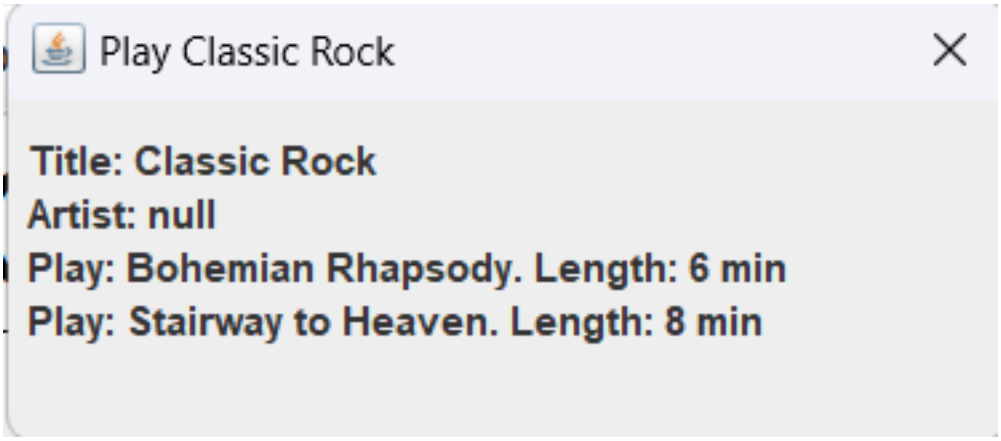


Figure 3.12 Demo Play button

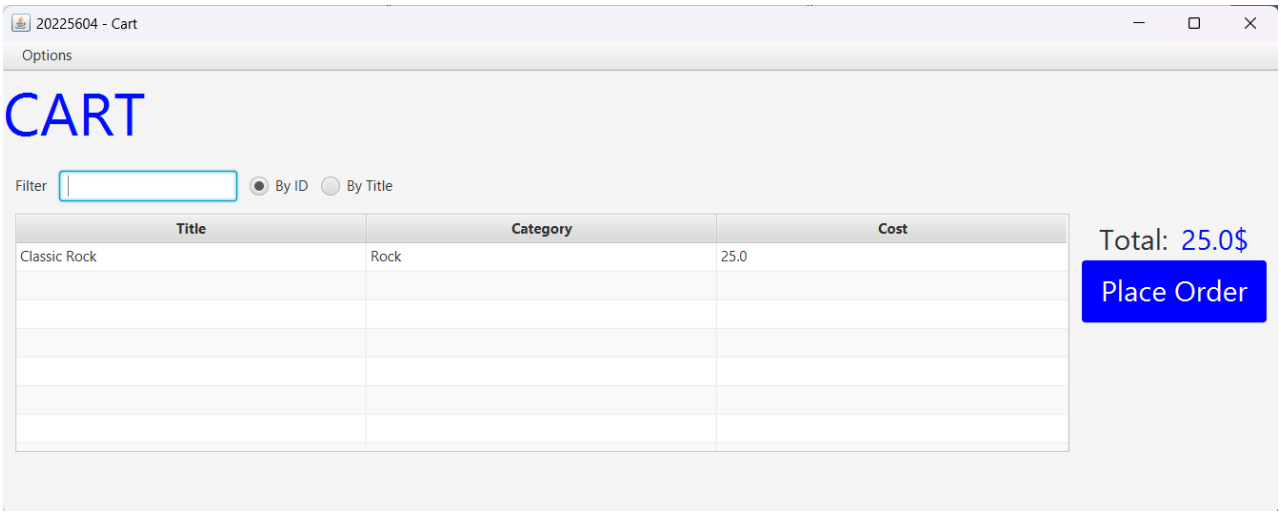


Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

```java
// PhamQuocCuong - 20225604
package hust.soict.hedspi.javafx;

import javafx.application.*;
import javafx.fxml.*;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.util.Objects;

public class Painter extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        try {
            System.out.println("Loading FXML...");
            Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("Painter.fxml")));
            System.out.println("FXML loaded successfully.");
            Scene scene = new Scene(root);
            stage.setTitle("Painter");
            stage.setScene(scene);
            stage.show();
        } catch (Exception e) {
            System.out.println("Exception: " + e.toString());
            e.printStackTrace();
        }
    }

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

Figure 4.1. Painter Class 1

4.2 Create Painter.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
            xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="PainterController">
    <top>
        <MenuBar BorderPane.alignment="CENTER">
            <menus>
                <Menu mnemonicParsing="false" text="File">
                    <items>
                        <MenuItem mnemonicParsing="false" text="Close" />
                    </items>
                </Menu>
                <Menu mnemonicParsing="false" text="Edit">
                    <items>
                        <MenuItem mnemonicParsing="false" text="Delete" />
                    </items>
                </Menu>
                <Menu mnemonicParsing="false" text="Help">
                    <items>
                        <MenuItem mnemonicParsing="false" text="About" />
                    </items>
                </Menu>
            </menus>
        </MenuBar>
    </top>
    <center>
        <AnchorPane id="drawingAreaPane" fx:id="drawingAreaPane" onMouseClicked="#drawAreaMouseDragged"
        onMouseDragged="#drawAreaMouseDragged" prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER" />
    </center>
    <left>
        <VBox prefHeight="150.0" prefWidth="100.0" style="-fx-background-color: grey;" BorderPane.alignment="CENTER">
            <children>
                <TitledPane animated="false" text="tool">
                    <content>
                        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="65.0" prefWidth="82.0">
                            <children>
                                <RadioButton fx:id="penRadioButton" layoutX="3.0" layoutY="10.0" mnemonicParsing="false" text="Pen" selected="true">
                                    <toggleGroup>
                                        <ToggleGroup fx:id="helloae" />
                                    </toggleGroup>
                                </RadioButton>
                                <RadioButton fx:id="eraserButton" layoutX="2.0" layoutY="40.0" mnemonicParsing="false" text="Eraser" toggleGroup="$helloae" />
                            </children>
                        </AnchorPane>
                    </content>
                </TitledPane>
                <Button fx:id="clearButton" maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear" />

            </children>
            <padding>
                <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
            </padding>
        </VBox>
    </left>
</BorderPane>
```

Figure 4.2: Painter.fxml 1

4.3 Create class PainterController

```java
// PhamQuocCuong - 20225604
package hust.soict.hedspi.javafx;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.Node;
import javafx.scene.control.RadioButton;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;

import java.util.Iterator;

public class PainterController {

    @FXML
    private RadioButton penRadioButton;

    @FXML
    private RadioButton eraserButton;

    @FXML
    private AnchorPane drawingAreaPane;

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

    @FXML
    void drawAreaMouseDragged(MouseEvent event) {
        if (penRadioButton.isSelected()) {
            Circle circle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
            drawingAreaPane.getChildren().add(circle);
        } else if (eraserButton.isSelected()) {
            Iterator<Node> iter = drawingAreaPane.getChildren().iterator();
            while (iter.hasNext()) {
                Node c = iter.next();
                if (c instanceof Circle circle) {
                    if (circle.getCenterX() <= event.getX() + 4 && circle.getCenterX() >= event.getX() - 4) {
                        if (circle.getCenterY() <= event.getY() + 4 && circle.getCenterY() >= event.getY() - 4) iter.remove();
                    }
                }
            }
        }
    }
}
```

Figure 4.4: PainterControl

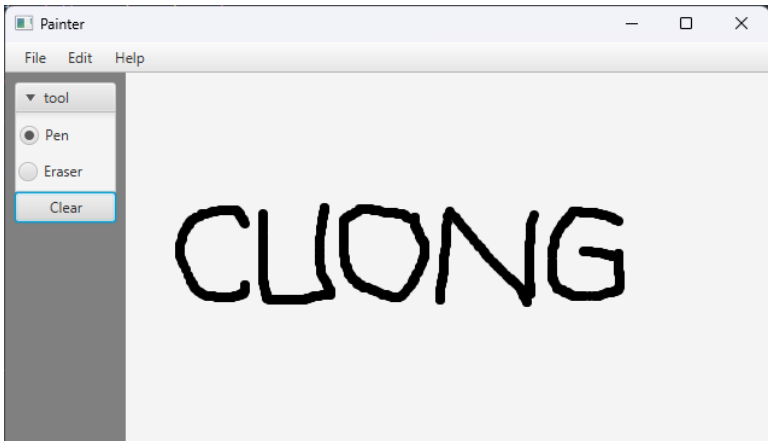


Figure 4.5: Use Pen

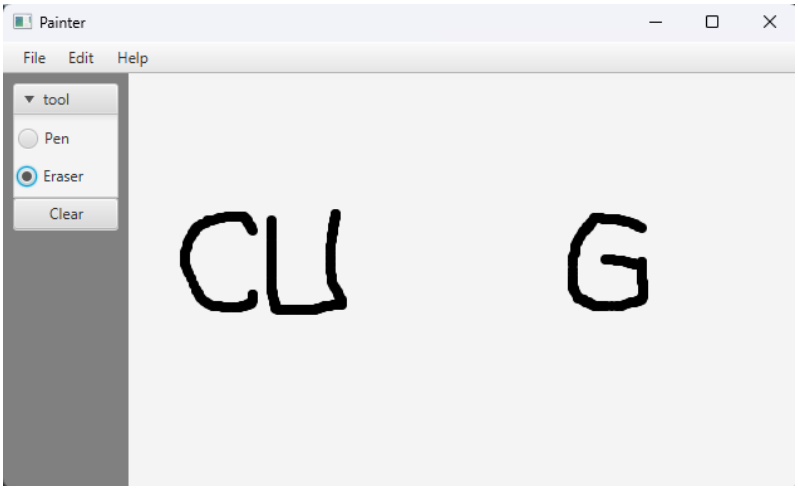


Figure 4.6: Use Eraser

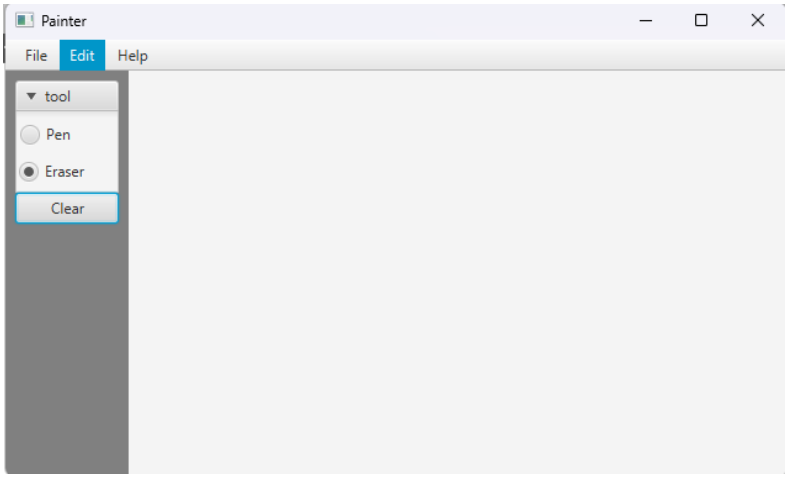


Figure 4.7: Clear button

5 View Cart Screen

5.1 Create cart.fxml

```xml
<?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/21" 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" text="Add book" />
                                        <MenuItem mnemonicParsing="false" text="Add DVD" />
                                        <MenuItem mnemonicParsing="false" text="Add CD" />
                                    </items>
                                </Menu>
                                <Menu mnemonicParsing="false" text="View store">
                                    <items>
                                        <MenuItem mnemonicParsing="false" text="Action 1" />
                                    </items>
                                </Menu>
                                <Menu mnemonicParsing="false" text="View cart">
                                    <items>
                                        <MenuItem mnemonicParsing="false" text="Action 1" />
                                    </items>
                                </Menu>
                            </items>
                        </Menu>
                    </menus>
                </MenuBar>
```

Figure 5.1: Cart.fxml 1

```
            <Label text="CART" textFill="#000fff">
                <font>
                    <Font size="50.0" />
                </font>
            </Label>
        </children>
    </VBox>
</top>
<center>
    <VBox BorderPane.alignment="CENTER">
        <padding>
            <Insets left="10.0" />
        </padding>
        <children>
            <HBox alignment="CENTER_LEFT" prefWidth="200.0" spacing="10.0">
                <padding>
                    <Insets bottom="10.0" top="10.0" />
                </padding>
                <children>
                    <Label text="Filter" />
                    <TextField fx:id="tfFilter" />
                    <RadioButton fx:id="radioBtnFilterId" mnemonicParsing="false" selected="true" text="By ID">
                        <toggleGroup>
                            <ToggleGroup fx:id="filterCategory" />
                        </toggleGroup>
                    </RadioButton>
                    <RadioButton fx:id="radioBtnFilterTitle" mnemonicParsing="false" text="By Title" toggleGroup="$filterCategory" />
                </children>
            </HBox>
            <TableView fx:id="tblMedia" prefHeight="200.0" prefWidth="200.0">
                <columns>
                    <TableColumn fx:id="colMediaTitle" prefWidth="75.0" text="Title" />
                    <TableColumn fx:id="colMediacategory" prefWidth="75.0" text="Category" />
                    <TableColumn fx:id="colMediaCost" prefWidth="75.0" text="Cost" />
                </columns>
                <columnResizePolicy>
                    <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
                </columnResizePolicy>
            </TableView>
            <ButtonBar prefHeight="40.0" prefWidth="200.0">
                <buttons>
                    <Button fx:id="btnPlay" mnemonicParsing="false" text="Play" />
                    <Button fx:id="btnRemove" mnemonicParsing="false" text="Remove" />
                </buttons>
            </ButtonBar>
        </children>
    </VBox>
</center>
```

Figure 5.2: Cart.fxml 2

```
    <right>
        <VBox alignment="TOP_CENTER" BorderPane.alignment="CENTER">
            <padding>
                <Insets left="10.0" right="10.0" top="50.0" />
            </padding>
            <children>
                <HBox alignment="CENTER">
                    <children>
                        <Label text="Total:">
                            <font>
                                <Font size="24.0" />
                            </font>
                            <padding>
                                <Insets right="10.0" />
                            </padding>
                        </Label>
                        <Label fx:id="totalPrice" text="0 $" textFill="#000fff">
                            <font>
                                <Font size="24.0" />
                            </font>
                        </Label>
                    </children>
                </HBox>
                <Button fx:id="btnPlaceOrder" mnemonicParsing="false" style="-fx-background-color: blue;" text="Place Order" textFill="WHITE">
                    <font>
                        <Font size="24.0" />
                    </font>
                </Button>
            </children>
        </VBox>
    </right>
</BorderPane>
```

Figure 5.3: Cart.fxml 3

5.2 Create class CartScreen

```java
// Pham Quoc Cuong
package hust.soict.dsai.aims.screen;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.*;
import javafx.application.Platform;
import javafx.embed.swing.JFXPanel;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javax.swing.*;
import java.io.IOException;

public class CartScreen extends JFrame {
    private Cart cart;

    public CartScreen(Cart cart) {
        super();
        this.cart = cart;
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle("20225604 - Cart");
        this.setVisible(true);
        setSize(1100,768);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource("cart.fxml"));
                    CartScreenController controller = new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

Figure 5.4: CartScreen class

5.3 Create class CartScreenController

```java
package hust.soict.dsai.aims.screen;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.*;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.transformation.FilteredList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.geometry.Pos;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.paint.*;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;

public class CartScreenController {
    private Cart cart;
    @FXML
    private Button btnPlaceOrder;
    @FXML
    private TextField tfFilter;
    @FXML
    private ToggleGroup filterCategory;
    @FXML
    private RadioButton radioBtnFilterId;
    @FXML
    private RadioButton radioBtnFilterTitle;
    @FXML
    private Button btnPlay;
    @FXML
    private Button btnRemove;
    @FXML
    private TableView<Media> tblMedia;
    @FXML
    private TableColumn<Media, String> colMediaTitle;
    @FXML
    private TableColumn<Media, String> colMediacategory;
    @FXML
    private TableColumn<Media,Float> colMediaCost;
    @FXML
    private Label totalPrice;

    public CartScreenController(Cart cart) {
        super();
        this.cart = cart;
    }
```

Figure 5.5: CartScreenController 1

```
@FXML
private void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media,String>("title"));
    colMediacategory.setCellValueFactory(new PropertyValueFactory<Media,String>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<Media,Float>("cost"));

    FilteredList<Media> filteredData = new FilteredList<>(FXCollections.observableList(this.cart.getItemsOrdered()), p -> true);
    tblMedia.setItems(filteredData);
    tblMedia.setPlaceholder(new Label("No item in cart"));
    btnPlay.setVisible(false);
    btnRemove.setVisible(false);

    btnRemove.setOnAction(new EventHandler<javafx.event.ActionEvent>() {
        @Override
        public void handle(javafx.event.ActionEvent event) {
            Media media = tblMedia.getSelectionModel().getSelectedItem();
            cart.removeMedia(media);
            totalPrice.setText(Float.toString(cart.totalCost()) +"$");
            tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
        }
    });

    tfFilter.textProperty().addListener((observable, oldValue, newValue) -> {
        showFilteredMedia(filteredData, newValue);
    });

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
            new ChangeListener<Media>() {
                @Override
                public void changed(ObservableValue<? extends Media> observableValue, Media media, Media newValue) {
                    updateButtonBar(newValue);
                }
            }
    );
    totalPrice.setText(Float.toString(cart.totalCost()) + "$");

    btnPlay.setOnAction(new EventHandler<ActionEvent>() {
        @Override
        public void handle(ActionEvent event) {
            JDialog playDialog = MediaStore.createPlayDialog(tblMedia.getSelectionModel().getSelectedItem());
            playDialog.setVisible(true);
            playDialog.setSize(300,200);
            playDialog.pack();
        }
    });

    btnPlaceOrder.setOnAction(new EventHandler<ActionEvent>() {
        @Override
        public void handle(ActionEvent event) {
            createPopUp();
            cart.getItemsOrdered().clear();
            tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
            totalPrice.setText(Float.toString(cart.totalCost()) + "$");
        }
    });
}
```

Figure 5.6: CartScreenController 2

```java
    @FXML
    void updateButtonBar(Media media) {
        btnRemove.setVisible(true);
        if(media instanceof Playable) {
            btnPlay.setVisible(true);
        } else {
            btnPlay.setVisible(false);
        }
    }

    @FXML
    void showFilteredMedia(FilteredList<Media> filteredData, String filter) {
        filteredData.setPredicate(media -> {
            if (filter == null || filter.isEmpty()) {
                return true;
            }
            String lowerCaseFilter = filter.toLowerCase();
            if (filterCategory.getSelectedToggle() == radioBtnFilterTitle) {
                return media.getTitle().toLowerCase().contains(lowerCaseFilter);
            } else if (filterCategory.getSelectedToggle() == radioBtnFilterId) {
                try {
                    return Integer.toString(media.getId()).contains(lowerCaseFilter);
                } catch (NumberFormatException e) {
                    return false;
                }
            }
            return false;
        });
    }

    @FXML
    void createPopUp() {
        Stage popupwindow =new Stage();
        popupwindow.initModality(Modality.APPLICATION_MODAL);
        popupwindow.setTitle("Place order");

        Label label1 = new Label("You have placed your order!");
        label1.setFont(Font.font("Arial", FontWeight.BOLD,14));
        Label label2 = new Label("Your bill total is " + Float.toString(cart.totalCost()) + "$");
        Button button1= new Button("OK!");
        label2.setTextFill(Color.RED);
        button1.setOnAction(e -> popupwindow.close());
        VBox layout= new VBox(10);
        layout.getChildren().addAll(label1, label2,button1);
        layout.setAlignment(Pos.CENTER);
        Scene scene1= new Scene(layout, 300, 200);
        popupwindow.setScene(scene1);
        popupwindow.show();
    }

    @FXML
    void btnRemovePressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        cart.removeMedia(media);
        totalPrice.setText(Float.toString(cart.totalCost()) + "$");
    }
}
```

5.4 Demo

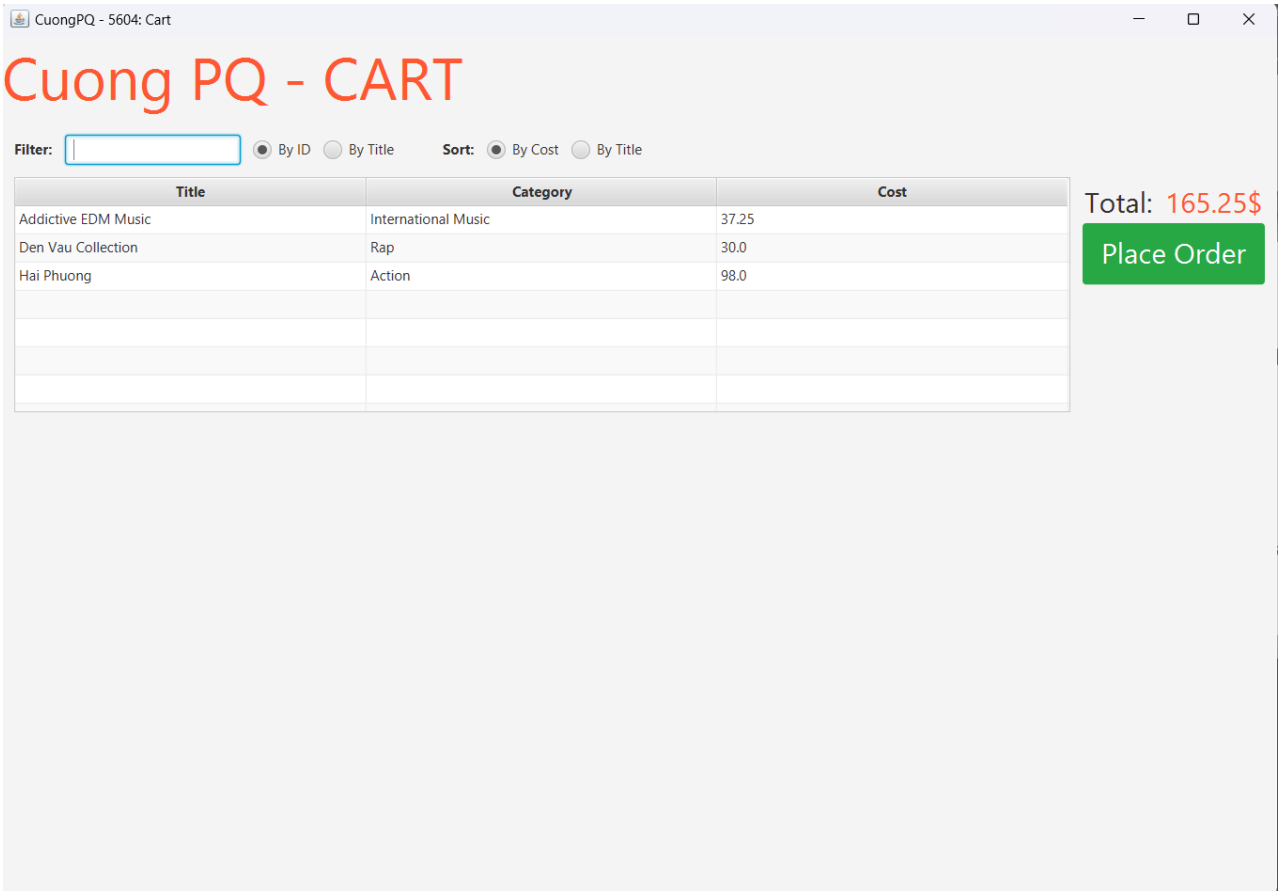


Figure 5.7: Demo CartScreen

6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController

```java
@FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<>("title"));
        colMediacategory.setCellValueFactory(new PropertyValueFactory<>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<>("cost"));
        tblMedia.setItems(FXCollections.observableList(this.cart.getItemsOrdered()));
        tblMedia.setPlaceholder(new Label("No item in cart"));
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);

        btnRemove.setOnAction(event -> {
            try {
                Media media = tblMedia.getSelectionModel().getSelectedItem();
                cart.removeMedia(media);
                totalPrice.setText(cart.totalCost() + "$");
                tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(null, "Cuong PQ 5604: Error: No media in cart to remove", "Error", JOptionPane.ERROR_MESSAGE);
            }
        });

        tfFilter.textProperty().addListener((observableValue, s, t1) -> showFilterMedia(t1));

        tblMedia.getSelectionModel().selectedItemProperty().addListener(
                (observableValue, media, t1) -> updateButtonBar(t1)
        );
        totalPrice.setText(cart.totalCost() + "$");

        btnPlay.setOnAction(event -> {
            try {
                JDialog playDialog = MediaStore.createPlayDialog(tblMedia.getSelectionModel().getSelectedItem());
                playDialog.setVisible(true);
                playDialog.setSize(300, 200);
                playDialog.pack();
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(null, "Cuong PQ 5604: Error: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
            }
        });

        btnPlaceOrder.setOnAction(event -> {
            createPopUp();
            cart.getItemsOrdered().clear();
            tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
            totalPrice.setText(cart.totalCost() + "$");
        });

        sortCategory.selectedToggleProperty().addListener((observable, oldValue, newValue) -> {
            if (newValue == radioBtnSortCost) {
                cart.getItemsOrdered().sort(Media.COMPARE_BY_COST_TITLE);
            } else if (newValue == radioBtnSortTitle) {
                cart.getItemsOrdered().sort(Media.COMPARE_BY_TITLE_COST);
            }
            tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
        });
    }
```

Figure 6.1: CartScreenController 1

6.2 Demo

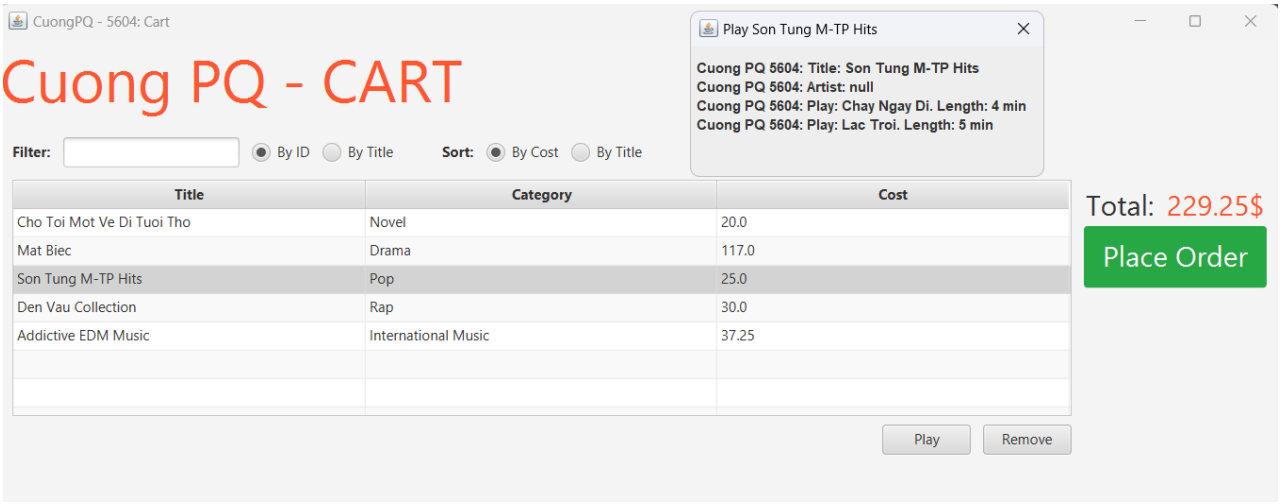


Figure 6.3: Demo media playable

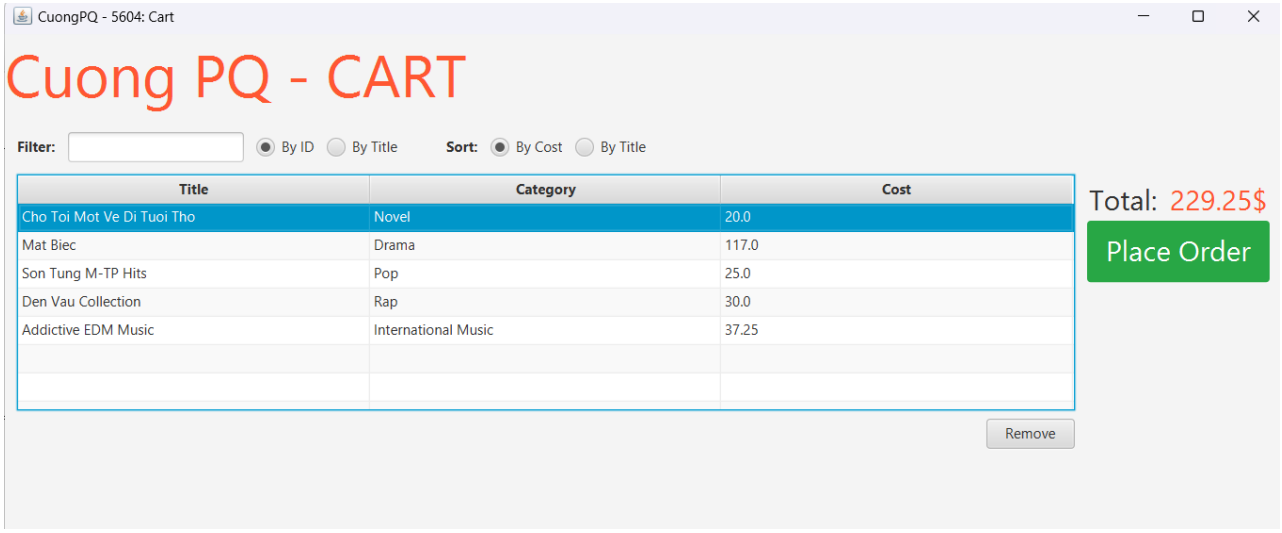


Figure 6.4: Demo media unplayable

7 Deleting a media

7.1 Code

```
btnRemove.setOnAction(new EventHandler<javafx.event.ActionEvent>() {
            @Override
            public void handle(javafx.event.ActionEvent event) {
                Media media = tblMedia.getSelectionModel().getSelectedItem();
                cart.removeMedia(media);
                totalPrice.setText(Float.toString(cart.totalCost()) +"$");
                tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
            }
        });
```

Figure 7.1: btnRemovePressed Method

7.2 Demo

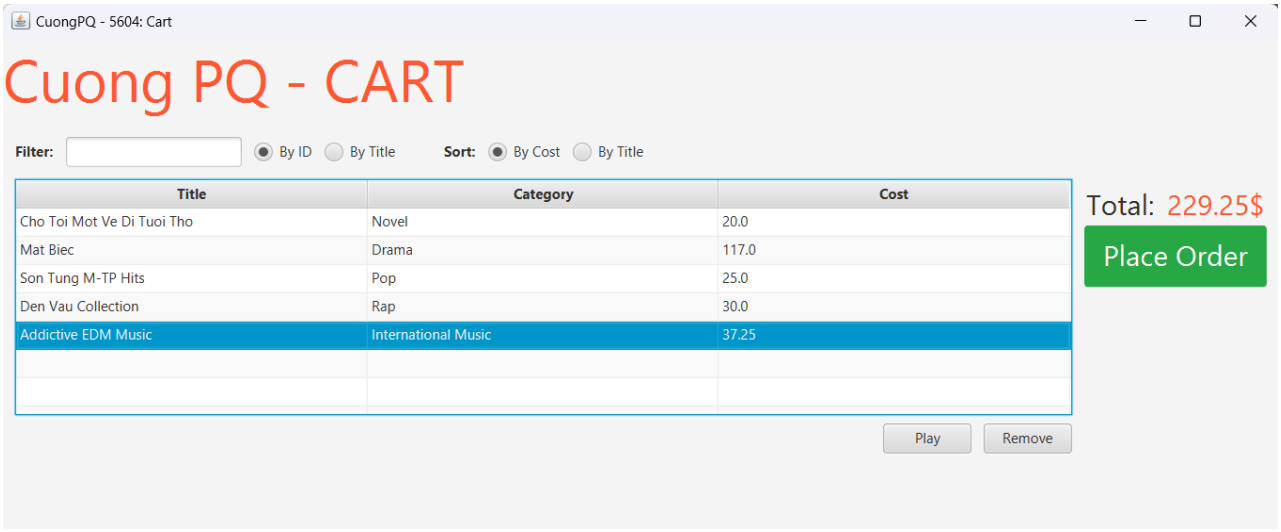


Figure 7.2: button Remove

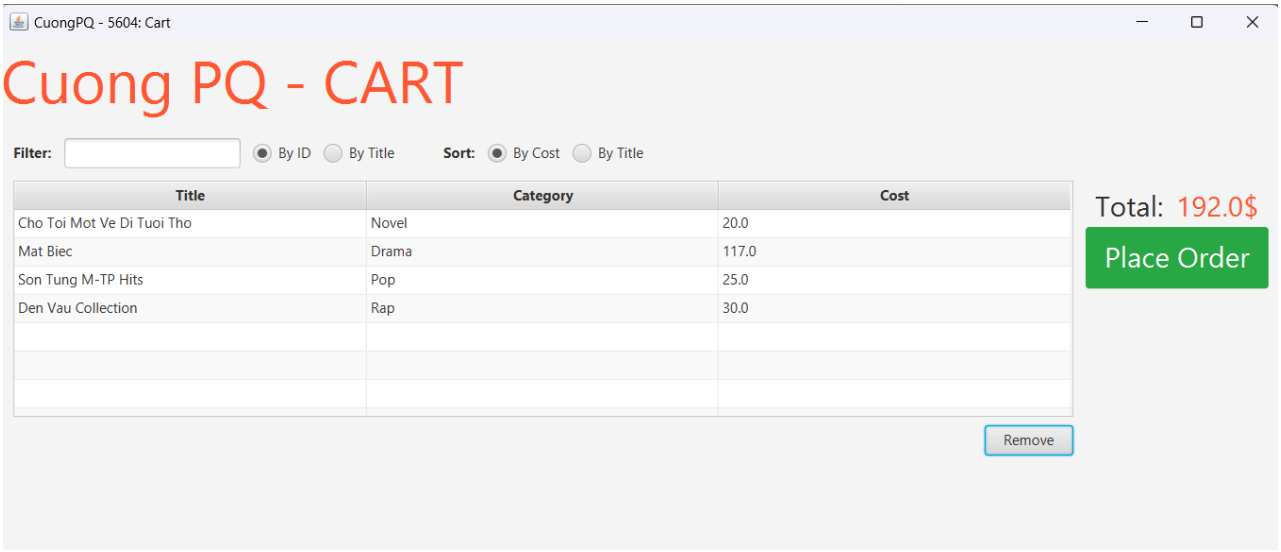


Figure 7.3: button Remove

8 Complete the Aims GUI application

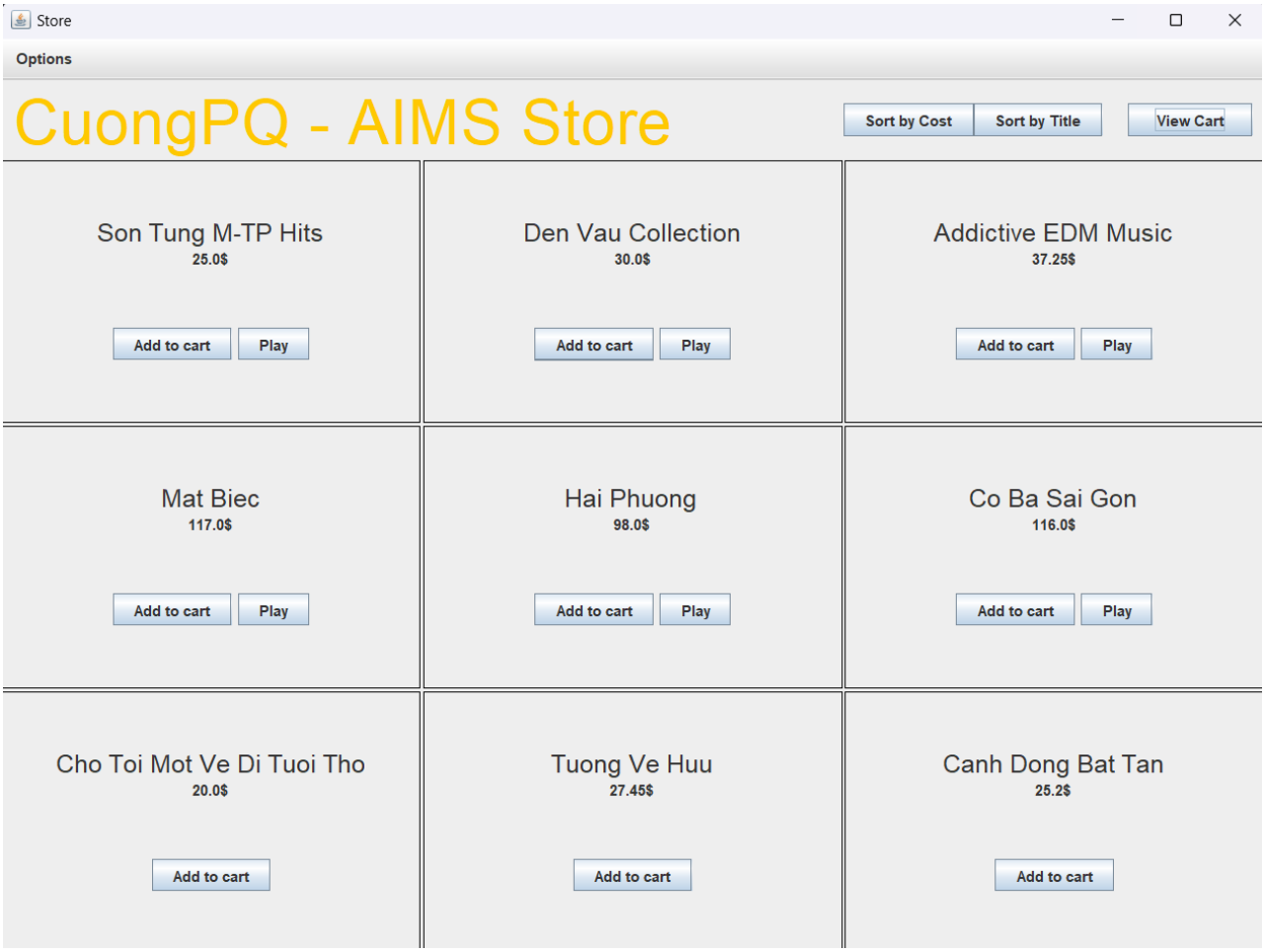


Figure 8.1: Store before add book

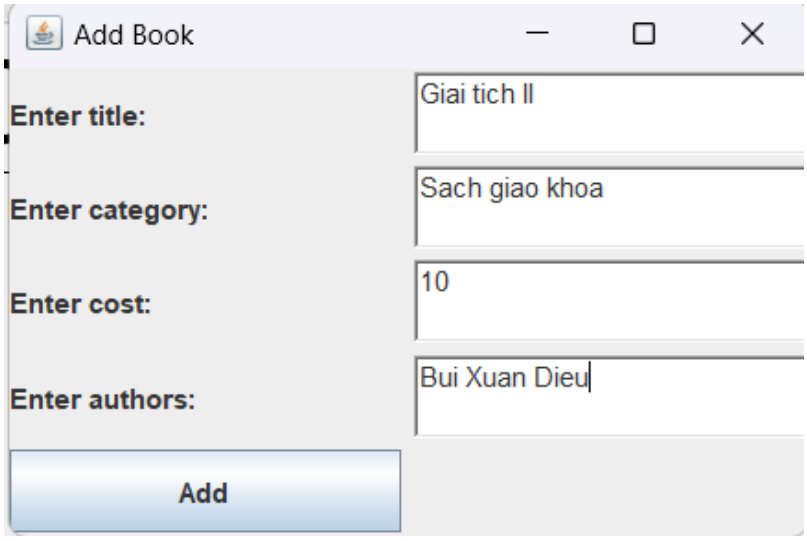


Figure 8.2: Add book

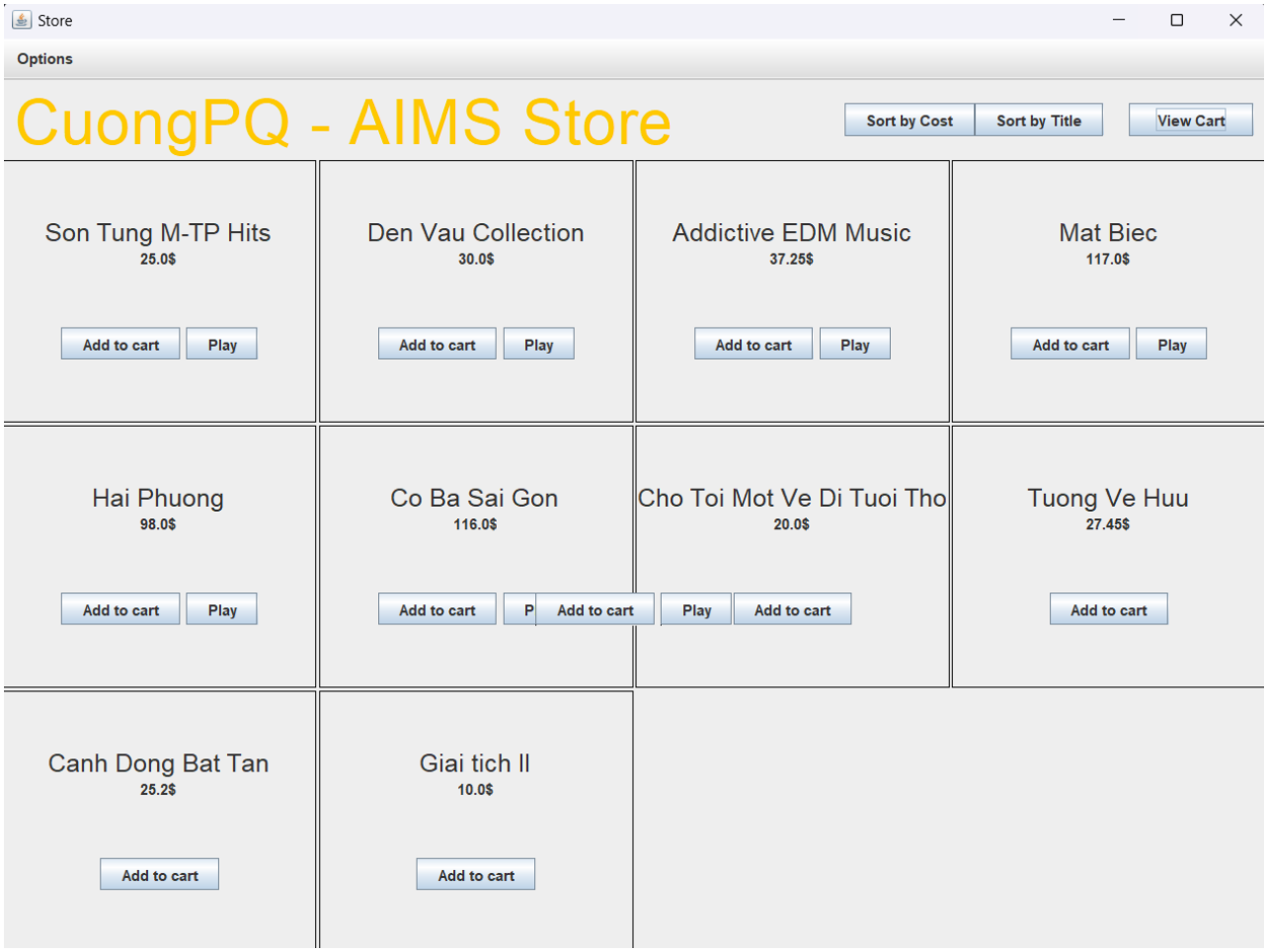


Figure 8.3: Store after add book

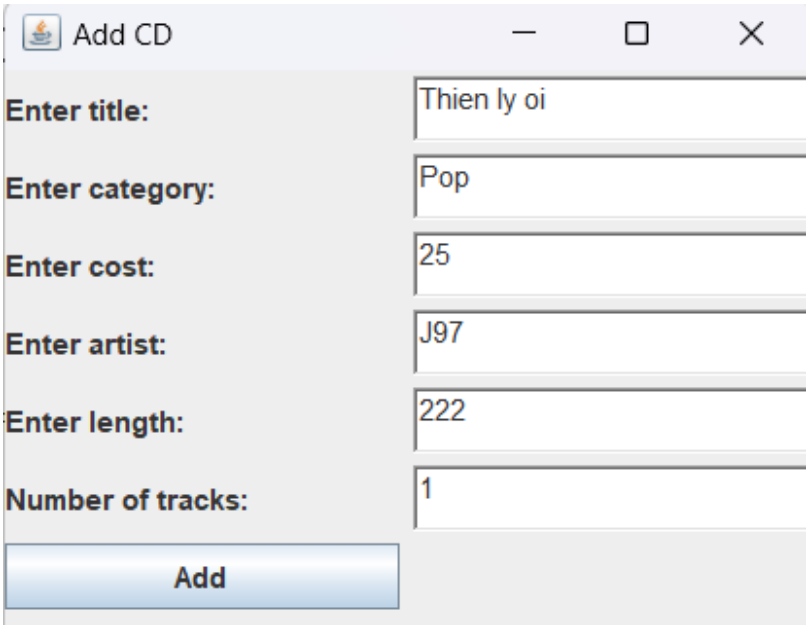


Figure 8.4: Add CD

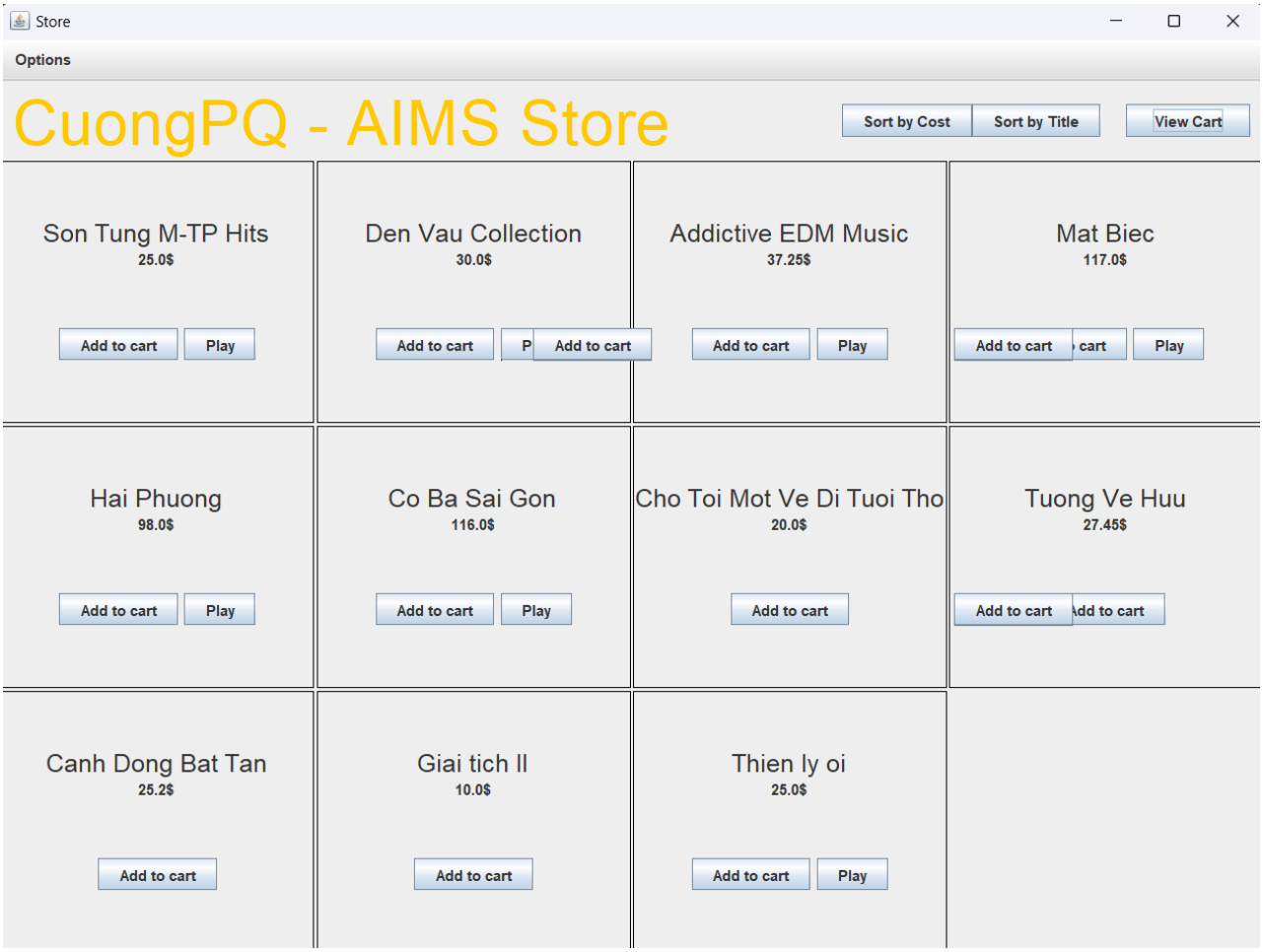


Figure 8.5: Store after add CD

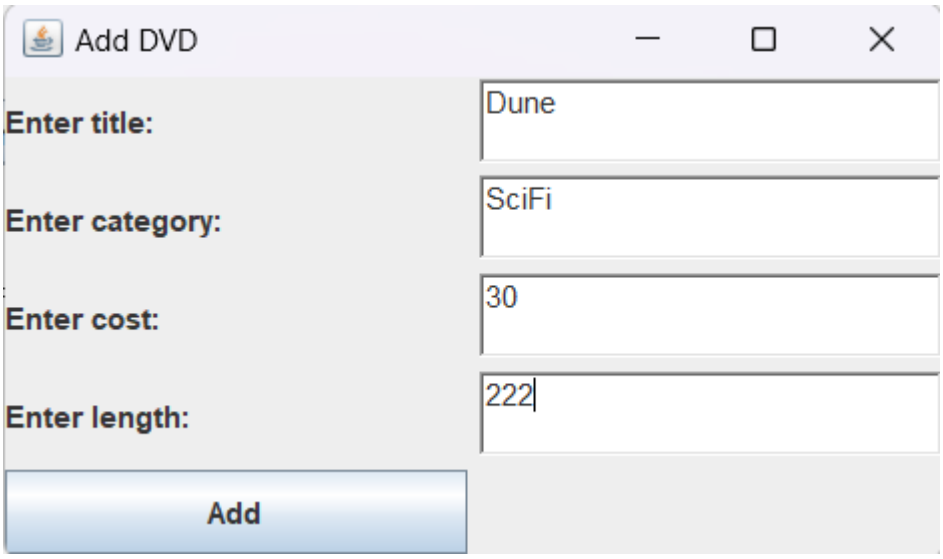


Figure 8.6 Add DVD

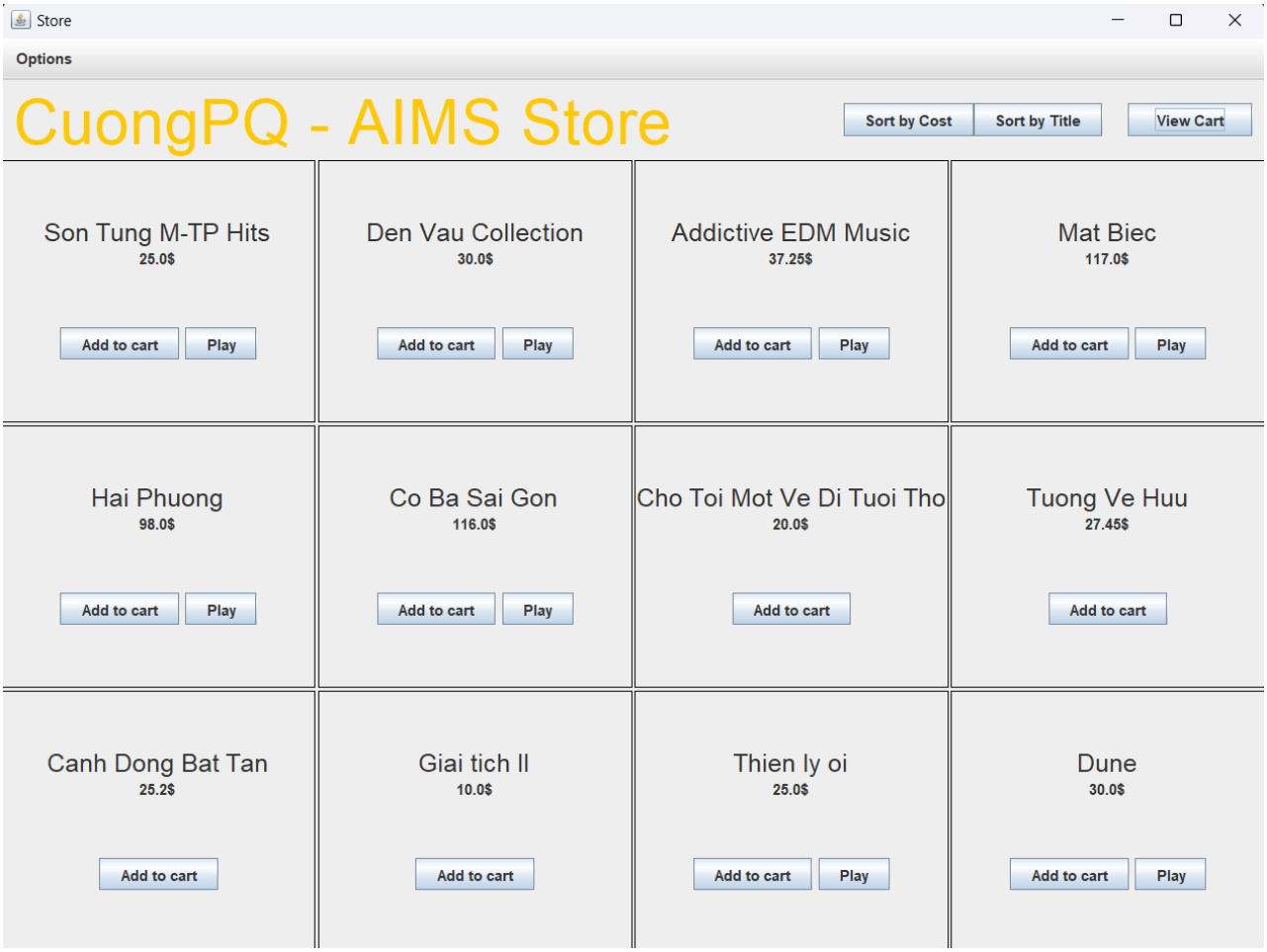


Figure 8.7: Store after add DVD

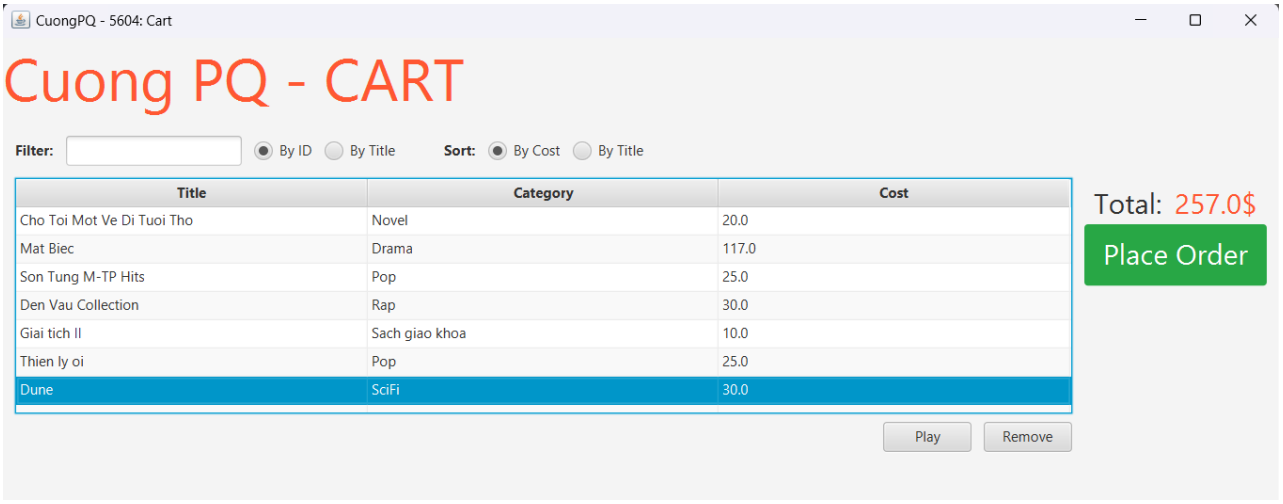


Figure 8.8: Cart

```java
@Override
public void play() throws PlayerException {
    if(this.getLength() > 0) {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    } else {
        throw new PlayerException(message:"ERROR: DVD length is non-positive");
    }
}

// Pham Quoc Cuong - 20225604
@Override
public void play() throws PlayerException {
    if(this.getLength() > 0) {
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("CD length: " + this.getLength());
        System.out.println("CD artist: " + this.getArtist());
        System.out.println(x:"CD tracks: ");
        for (Track track : tracks) {
            track.play();
        }
    } else {
        throw new PlayerException(message:"ERROR: CD length is non-positive");
    }
}       You, 3 days ago • simple
// Pham Quoc Cuong - 20225604      You, 3 days ago • simple
@Override
public void play() throws PlayerException {
    if(this.getLength() > 0) {
        System.out.println("Playing Track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    } else {
        throw new PlayerException(message:"ERROR: Track length is non-positive");
    }
}
```

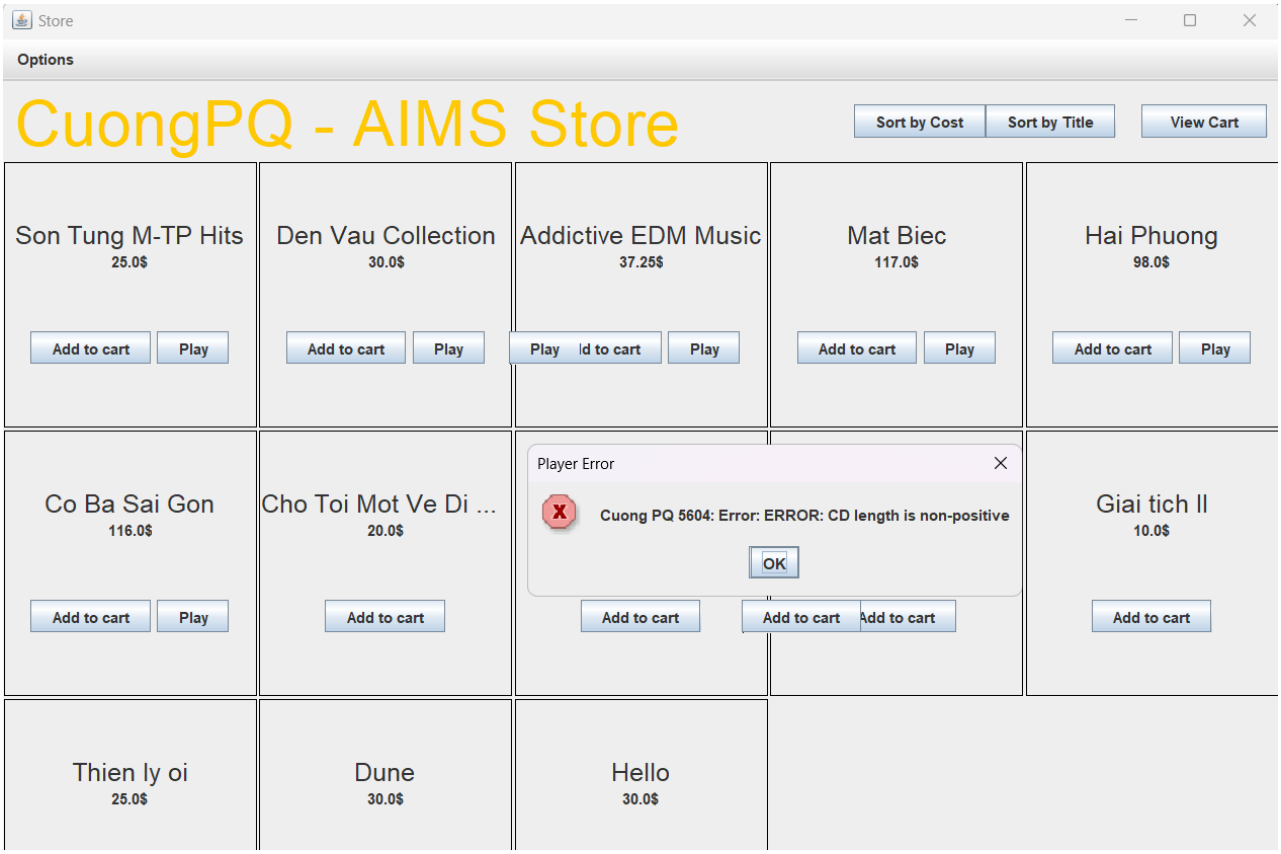
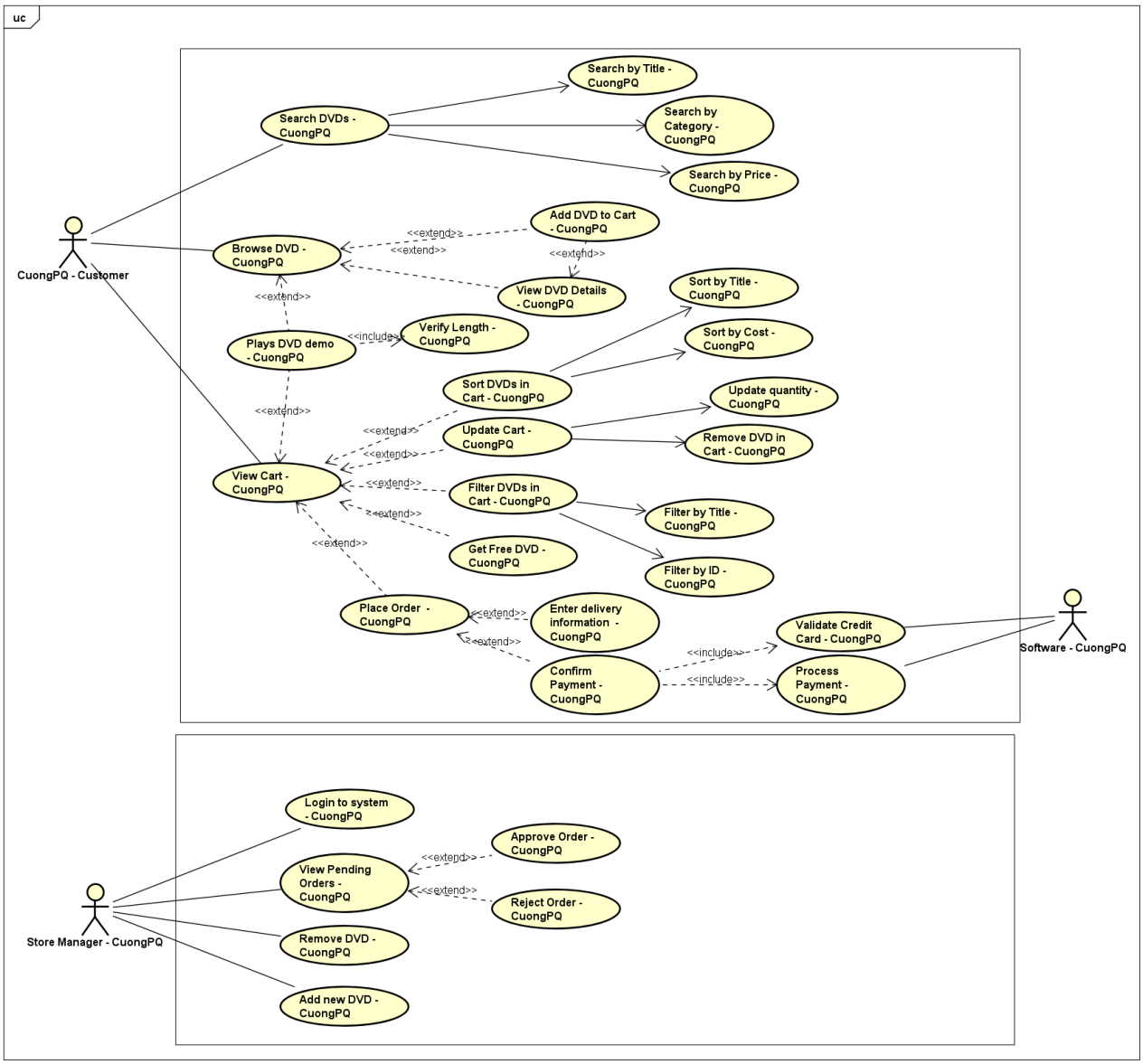
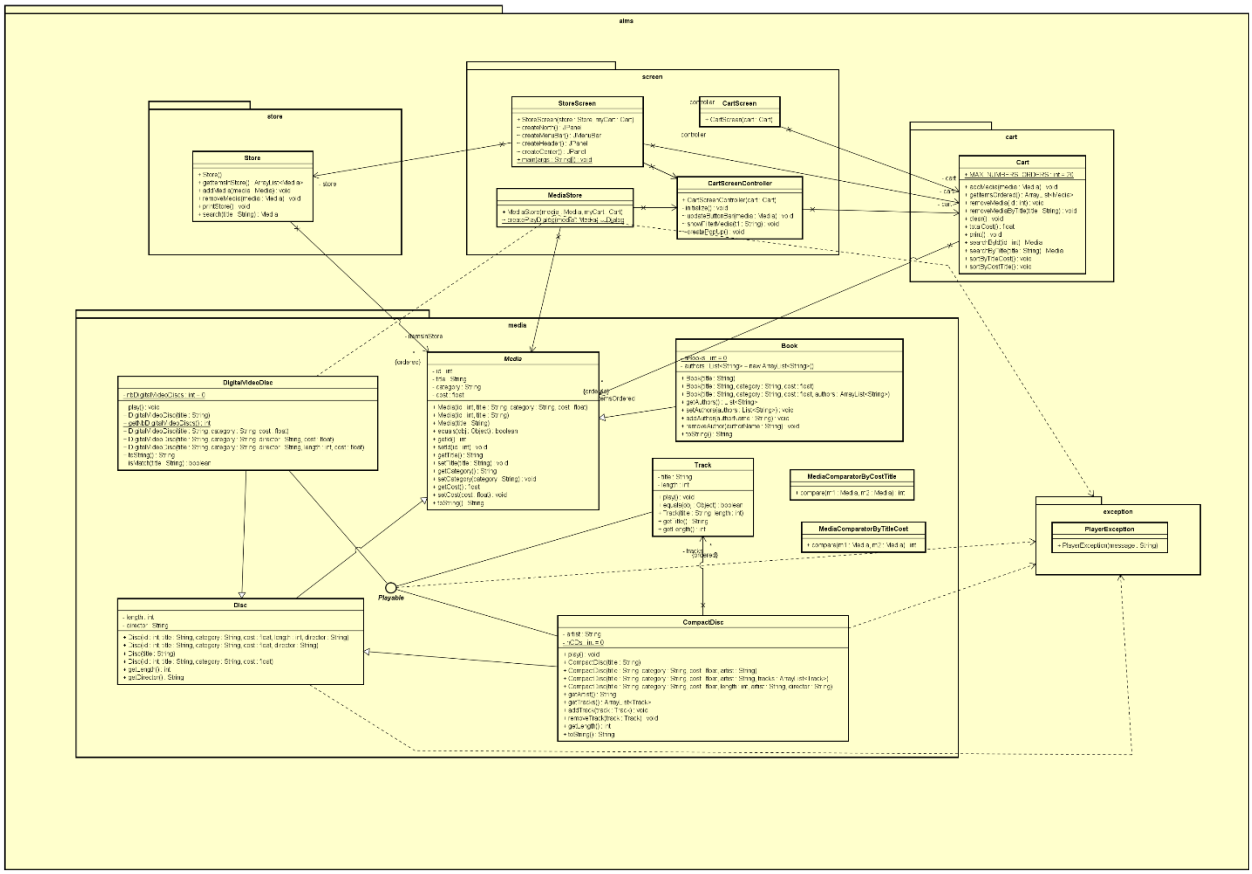


Figure 8.9: Exception

9 Use case Diagram

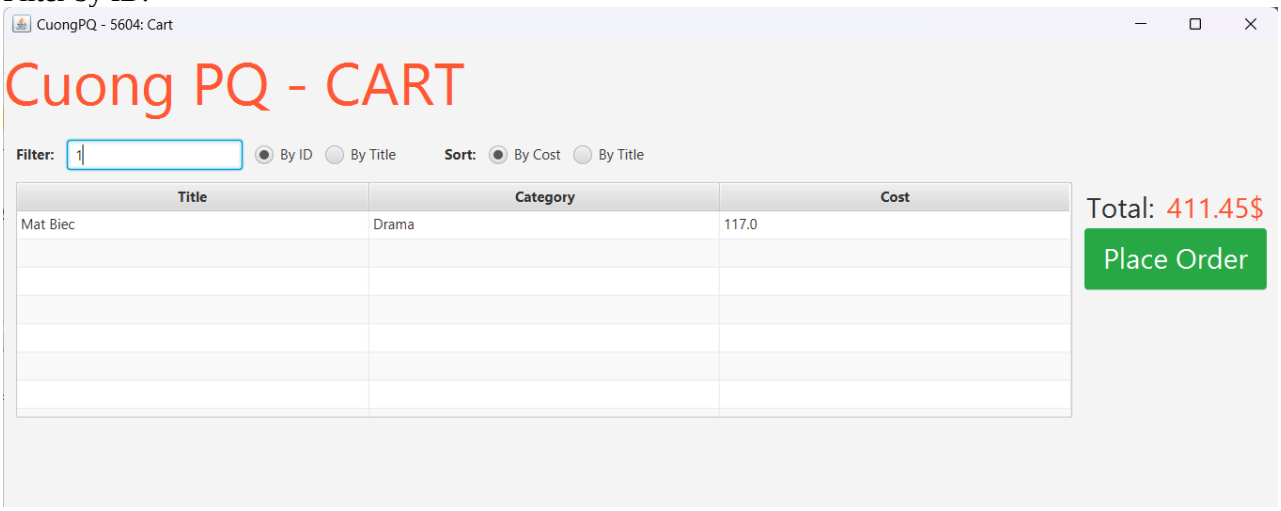

uc
Search by Title - CuongPQ
Search DVDs - CuongPQ
Search by Category - CuongPQ
Search by Price - CuongPQ
Add DVD to Cart - CuongPQ
CuongPQ - Customer
Browse DVD - CuongPQ
View DVD Details - CuongPQ
Sort by Title - CuongPQ
Sort by Cost - CuongPQ
Plays DVD demo - CuongPQ
Verify Length - CuongPQ
Sort DVDs in Cart - CuongPQ
Update quantity - CuongPQ
Update Cart - CuongPQ
Remove DVD in Cart - CuongPQ
View Cart - CuongPQ
Filter DVDs in Cart - CuongPQ
Filter by Title - CuongPQ
Get Free DVD - CuongPQ
Filter by ID - CuongPQ
Place Order - CuongPQ
Enter delivery information - CuongPQ
Validate Credit Card - CuongPQ
Confirm Payment - CuongPQ
Process Payment - CuongPQ
Software - CuongPQ
Login to system - CuongPQ
Approve Order - CuongPQ
View Pending Orders - CuongPQ
Reject Order - CuongPQ
Store Manager - CuongPQ
Remove DVD - CuongPQ
Add new DVD - CuongPQ
<<extend>>
<<include>>

10 Class Diagram

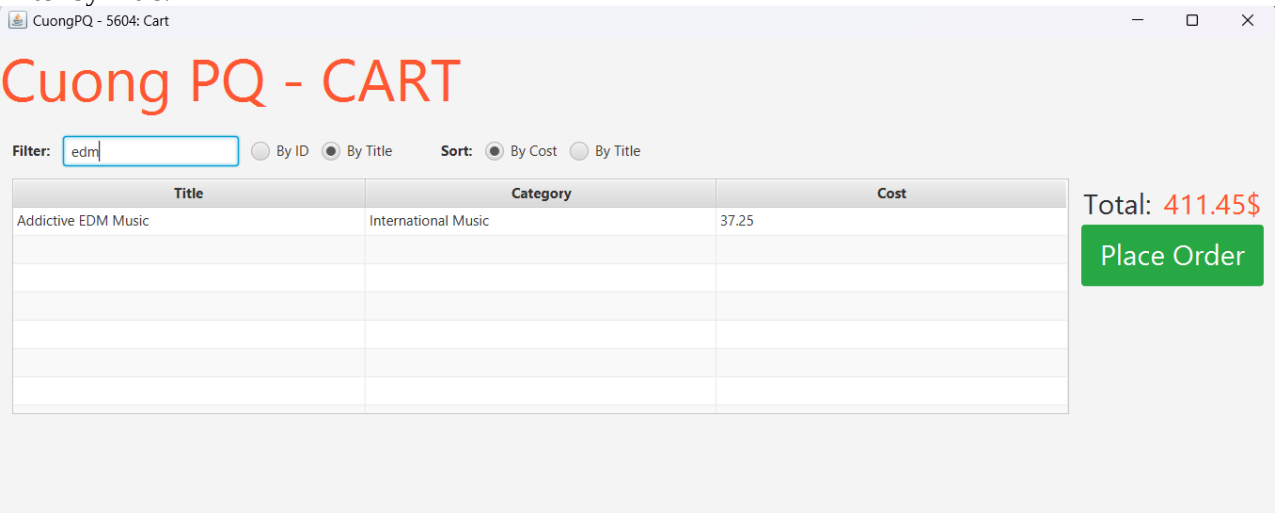


11 Filter items in cart

Filter by ID:

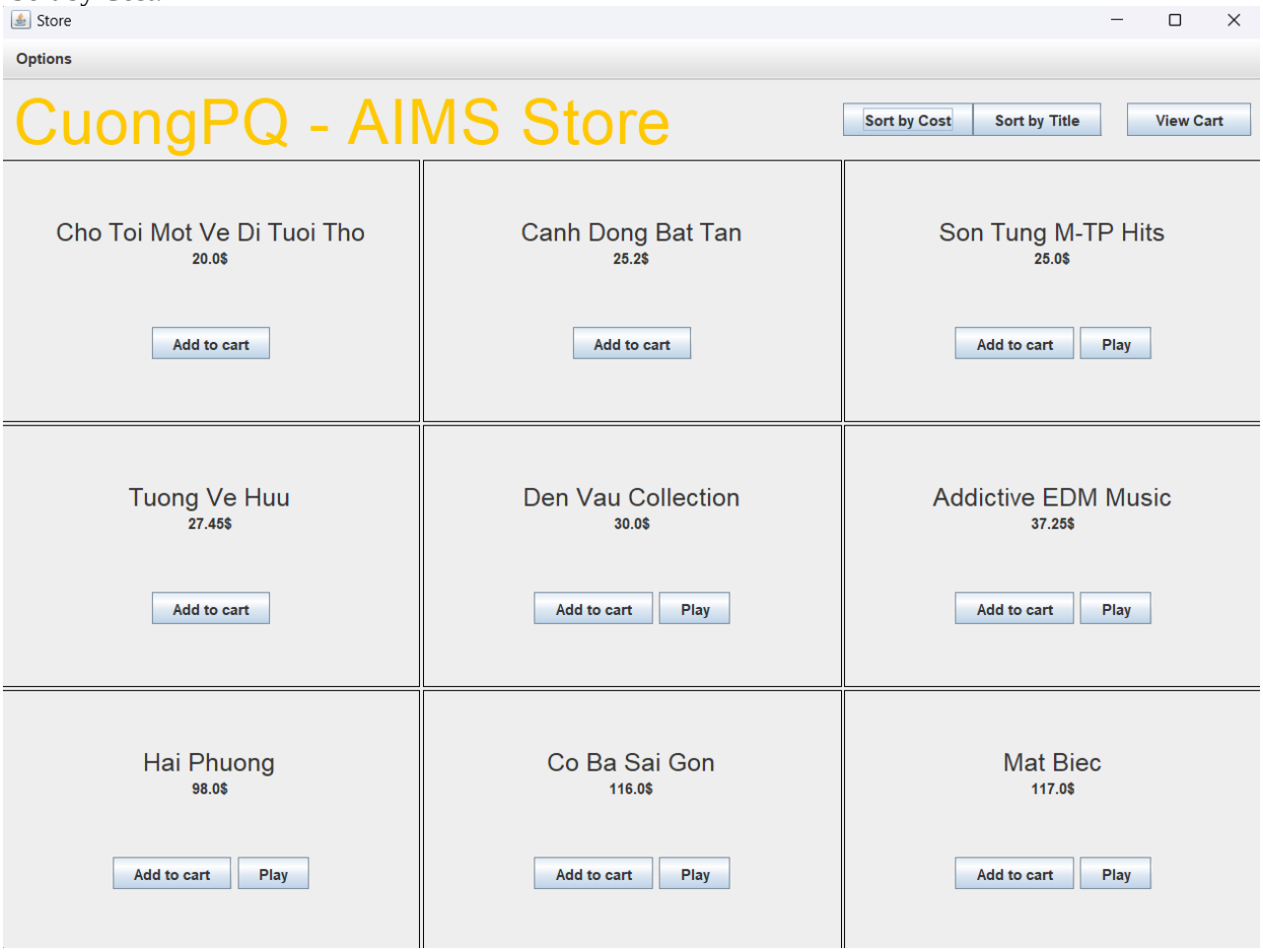


Filter by Title:

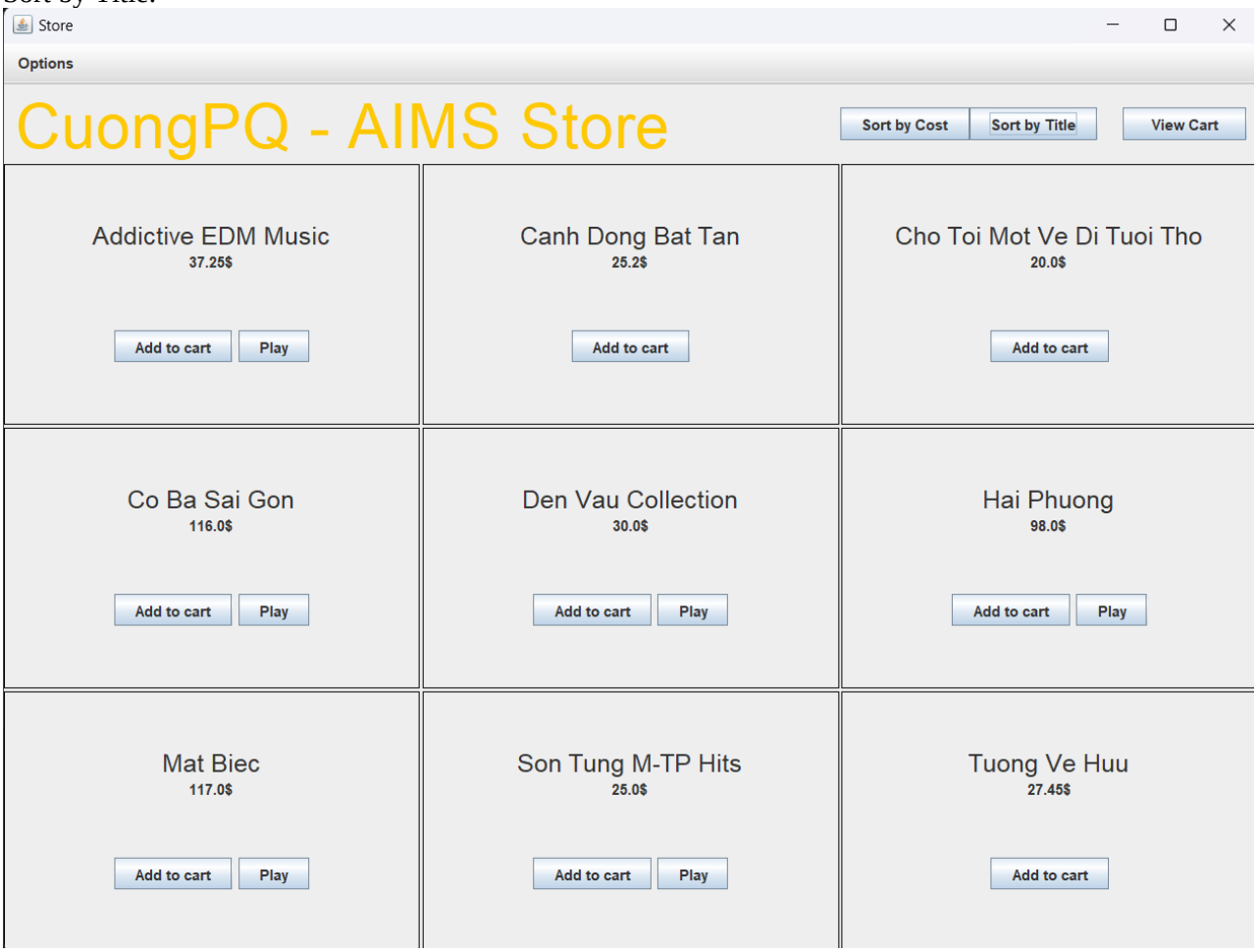


12 Sort items in store

Sort by Cost:

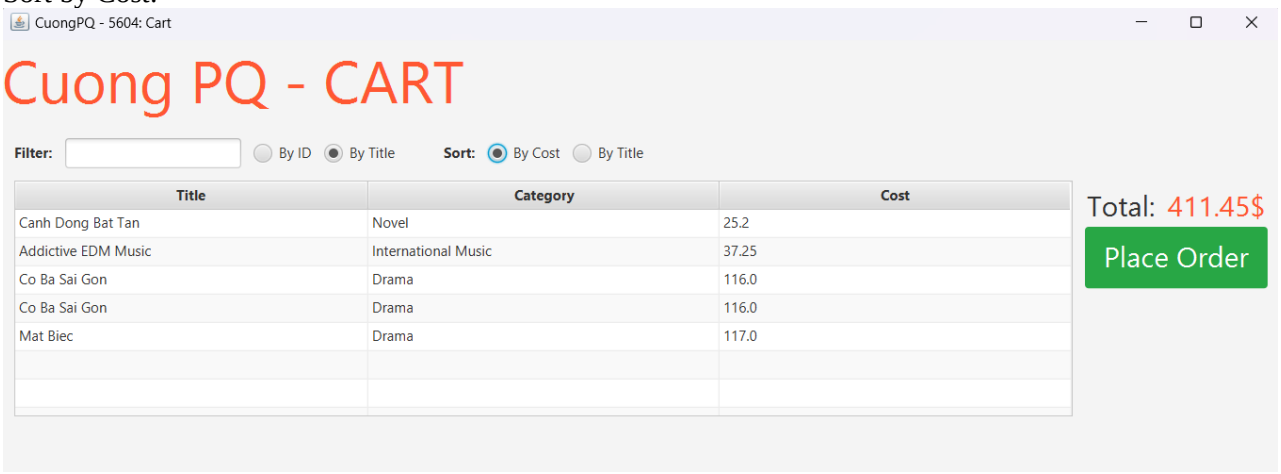


Sort by Title:

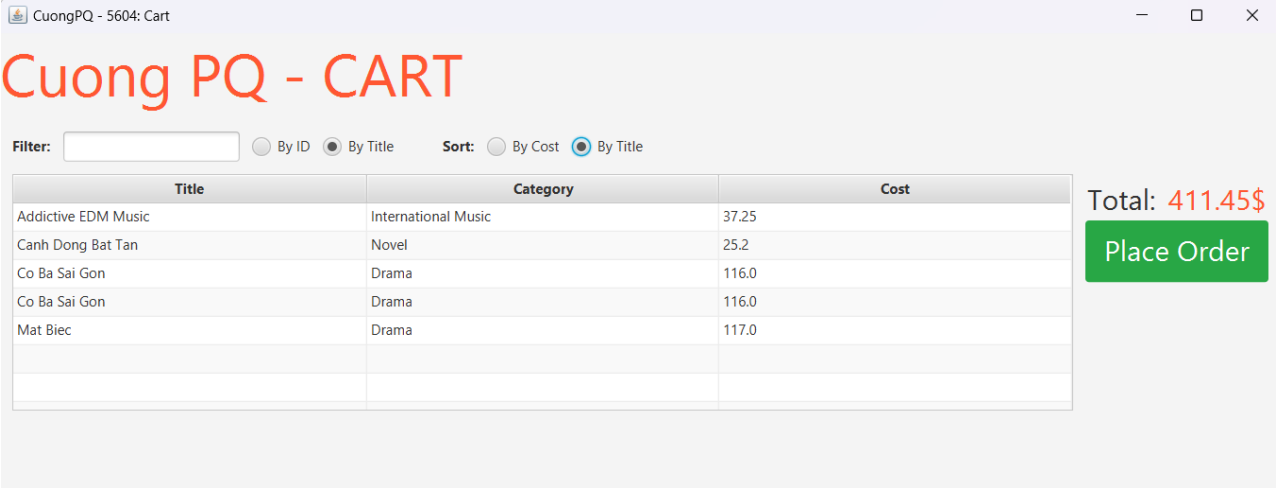


13 Sort items in cart

Sort by Cost:



Sort by Title:



14 Answer Questions

The Aims class must be updated to handle any exceptions generated when the play() methods are called. What happens when you don't update for them to catch?

Nếu lớp Aims không được điều chỉnh để xử lý ngoại lệ khi gọi phương thức play(), chương trình có thể gặp sự cố và dừng đột ngột mà không có cách khắc phục cụ thể. Điều này khiến người dùng khó xác định nguyên nhân gây ra lỗi, dẫn đến trải nghiệm sử dụng không thuận lợi.
Bằng cách sử dụng khối try-catch, ngoại lệ có thể được xử lý một cách chủ động. Khi lỗi xảy ra, khối catch sẽ nắm bắt và cung cấp thông tin chi tiết thông qua các phương thức như getMessage(), toString(), hoặc printStackTrace(). Điều này không chỉ giúp người dùng hiểu rõ vấn đề mà còn cho phép họ tiếp tục sử dụng ứng dụng mà không bị gián đoạn.