

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

BÁO CÁO THỰC HÀNH  
**IT3103-744527-2024.1**

**BÀI THỰC HÀNH 05**

Họ và tên sv: **Trịnh Hữu An**  
Lớp: **Việt Nhật 03 – K67**

GVHD: **Lê Thị Hoa**

TA: **Đặng Mạnh Cường**

Hà Nội 12/2024

## Table of Contents

|     |  |    |
|-----|--|----|
| 1.  | Swing components .....                                     | 4  |
| 1.1 | AWTAccumulator.....  | 4  |
| 1.2 | SwingAccumulator.....                                      | 6  |
| 2   | Organizing Swing components with Layout Managers.....      | 8  |
| 2.1 | Code .....   | 8  |
| 2.2 | Demo.....  | 10 |
| 3   | Create a graphical user interface for AIMS with Swing..... | 12 |
| 3.1 | Create class StoreScreen.....                              | 12 |
| 3.2 | Create class MediaStore.....                               | 15 |
| 3.3 | Demo.....  | 17 |
| 4   | JavaFX API .....   | 20 |
| 4.1 | Create class Painter .....                                 | 20 |
| 4.2 | Create Painter.fxml .....                                  | 20 |
| 4.3 | Create class PainterController .....                       | 22 |
| 5   | View Cart Screen.....                                      | 26 |
| 5.1 | Create cart.fxml.....                                      | 26 |
| 5.2 | Create class CartScreen .....                              | 29 |
| 5.3 | Create class CartScreenController .....                    | 31 |
| 5.4 | Demo.....  | 34 |
| 6.1 | Edit class CartScreenController .....                      | 35 |
| 6.2 | Demo.....  | 36 |
| 7   | Deleting a media .....                                     | 37 |
| 7.1 | Code .....   | 37 |
| 7.2 | Demo.....  | 37 |
| 8   | Complete the Aims GUI application.....                     | 38 |
| 9   | Use case Diagram.....                                      | 45 |
| 10  | Class Diagram.....   | 46 |

## Table of Figures

|   |    |
|---|----|
| Figure 1.1: Source code of AWTAccumulator ..... | 4  |
| Figure 21.2: Demo of AWTAccumulator .....       | 6  |
| Figure 3: Source code of NumberGrid .....       | 9  |
| Figure 4: Demo NumberGrid.....                  | 12 |
| Figure 5: Class StoreScreen.....                | 15 |
| Figure 6: Class MediaStore.....                 | 16 |
| Figure 7: Demo.....                             | 19 |
| Figure 8: Class Painter.....                    | 20 |
| Figure 9: Painter.fxml.....                     | 21 |
| Figure 10: PainterController .....              | 22 |
| Figure 11: Use pen .....                        | 23 |
| Figure 12: Use eraser .....                     | 24 |
| Figure 13: Clear Button .....                   | 25 |
| Figure 14: cart.fxml .....                      | 28 |
| Figure 15: Create CartScreen .....              | 30 |
| Figure 16: Create CartScreenController.....     | 33 |
| Figure 17: Demo.....                            | 34 |
| Figure 18: Edit class CartScreenController..... | 35 |
| Figure 19: Demo.....                            | 36 |
| Figure 20: btnRemovePressed method .....        | 37 |
| Figure 21: Demo.....                            | 38 |
| Figure 22: Store before add book .....          | 39 |
| Figure 23: Add book .....                       | 39 |
| Figure 24: Store after add book .....           | 39 |
| Figure 25: Add CD .....                         | 40 |
| Figure 26: Store after add CD.....              | 40 |
| Figure 27: Add DVD.....                         | 41 |
| Figure 28: Store after add DVD .....            | 41 |
| Figure 29: Add to Cart.....                     | 43 |
| Figure 30: Cart.....                            | 43 |
| Figure 31: Exception.....                       | 44 |
| Figure 32: Use case diagram .....               | 45 |
| Figure 33: Class Diagram.....                   | 46 |

# 1. Swing components

## 1.1 AWTAccumulator

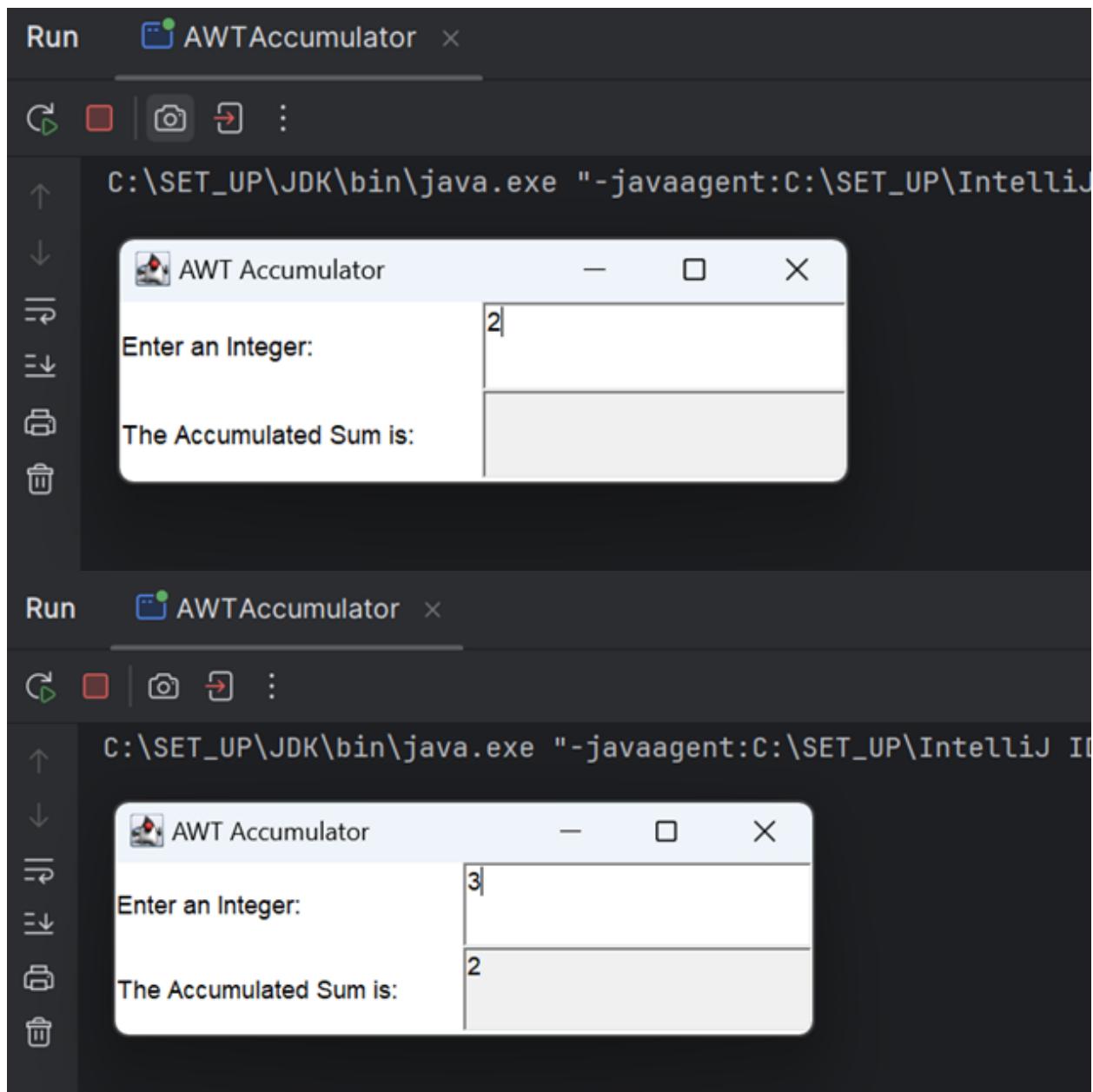
```

1 package hust.soict.hedspi.swing;
2
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 // Trịnh Hữu An 20225593
7 public class AWTAccumulator extends Frame {
8     private TextField tfInput; 5 usages
9     private TextField tfOutput; 4 usages
10    private int sum = 0; 2 usages
11
12    public AWTAccumulator(){ 1 usage
13        setLayout(new GridLayout( 2, cols: 2));
14        add(new Label( text: "Enter an Integer: "));
15        tfInput = new TextField( columns: 10);
16        add(tfInput);
17        tfInput.addActionListener(new TFIInputListener());
18        add(new Label( text: "The Accumulated Sum is: "));
19        tfOutput = new TextField((10));
20        tfOutput.setEditable(false);
21        add(tfOutput);
22        setTitle("AWT Accumulator");
23        setSize( width: 350, height: 120);
24        setVisible(true);
25    }
...
public static void main(String[] args) {
    new AWTAccumulator();
}

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

```

Figure 1.1: Source code of AWTAccumulator



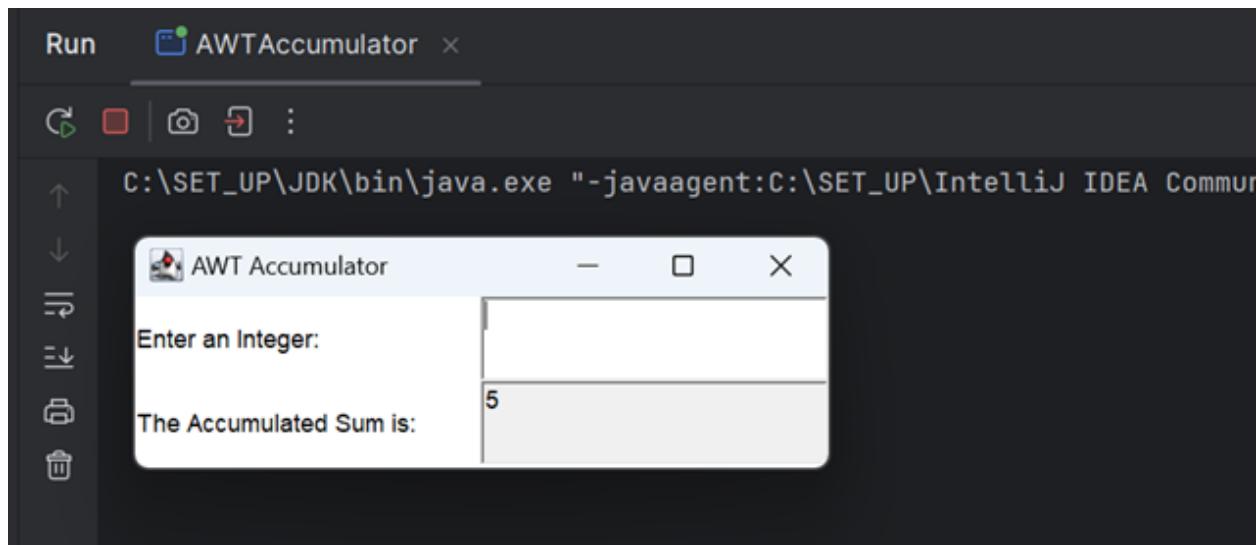


Figure 21.2: Demo of AWTAccumulator

## 1.2 SwingAccumulator

```

1 package hust.soict.hedspi.swing;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 //Trịnh Hữu An 20225593
8 public class SwingAccumulator extends JFrame {
9     private JTextField tfInput; 5 usages
10    private JTextField tfOutput; 4 usages
11    private int sum = 0; 2 usages
12
13    public SwingAccumulator(){ 1 usage
14        Container cp = getContentPane();
15        cp.setLayout(new GridLayout( rows: 2, cols: 2));
16        cp.add(new JLabel( text: "Enter an Integer: "));
17        tfInput=new JTextField( columns: 10);
18        cp.add(tfInput);
19        tfInput.addActionListener(new TFIputListener());
20        cp.add(new JLabel( text: "The Accumulated Sum is: "));
21        tfOutput = new JTextField( columns: 10);
22        tfOutput.setEditable(false);
23        cp.add(tfOutput);
24        setTitle("Swing Accumulator");

```

```

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

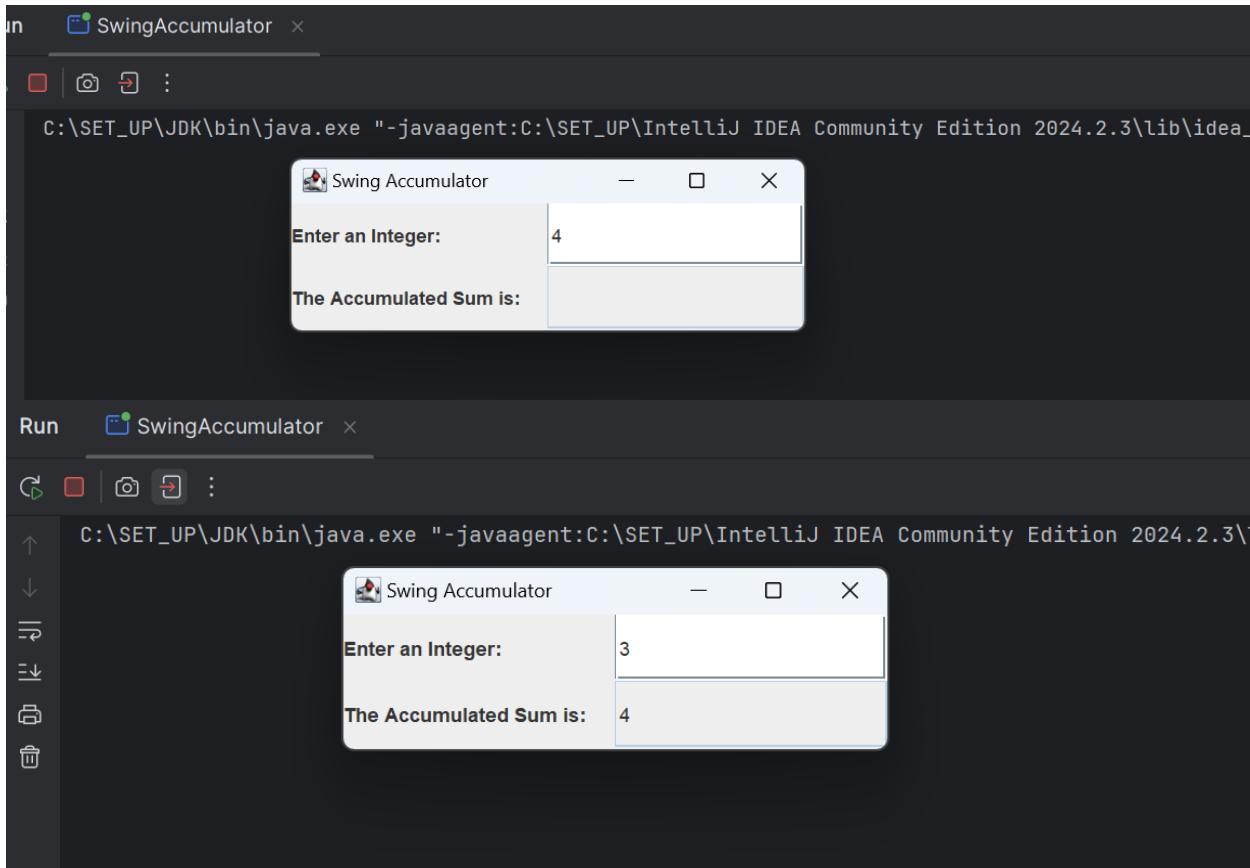
    }

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

    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 1.3 : Source code of SwingAccumulator



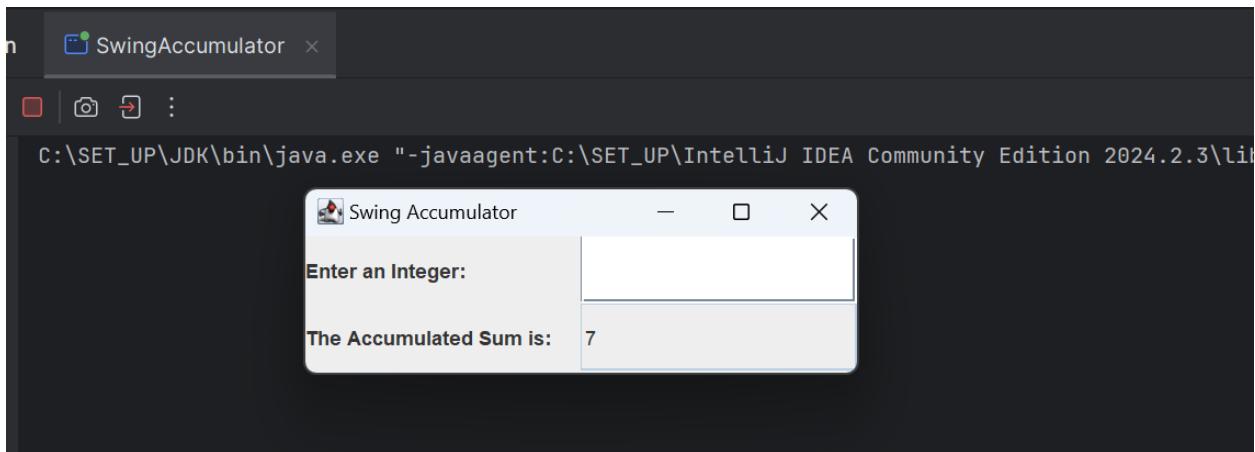


Figure 1.4: Demo of SwingAccumulator

## 2 Organizing Swing components with Layout Managers

### 2.1 Code

```
package hust.soict.hedspi.swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
//Trịnh Hữu An 20225593
> public class NumberGrid extends JFrame {
    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.add(tfDisplay, BorderLayout.NORTH);
        cp.add(panelButtons, BorderLayout.CENTER);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Number Grid");
        setSize( width: 400, height: 400);
        setVisible(true);
    }
}
```

```

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

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

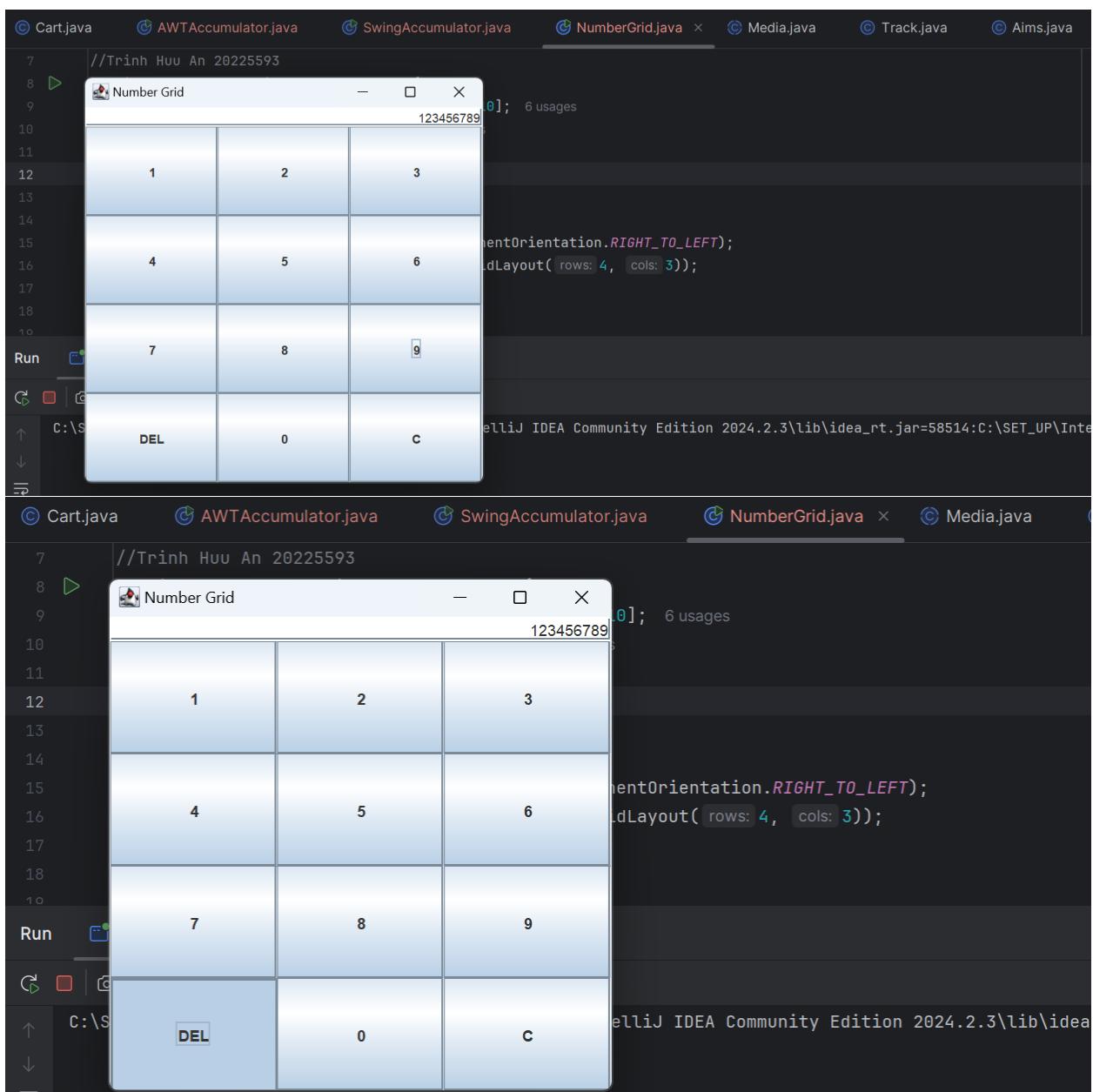
    btnReset = new JButton( text: "C");
    panelButtons.add(btnReset);
    btnReset.addActionListener(btnListener);
}

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 temp = tfDisplay.getText();
            try {
                tfDisplay.setText(temp.substring(0, temp.length() - 1));
            } catch (IndexOutOfBoundsException exception){
                exception.printStackTrace();
            }
        }
        else {
            tfDisplay.setText(null);
        }
    }
}
}

```

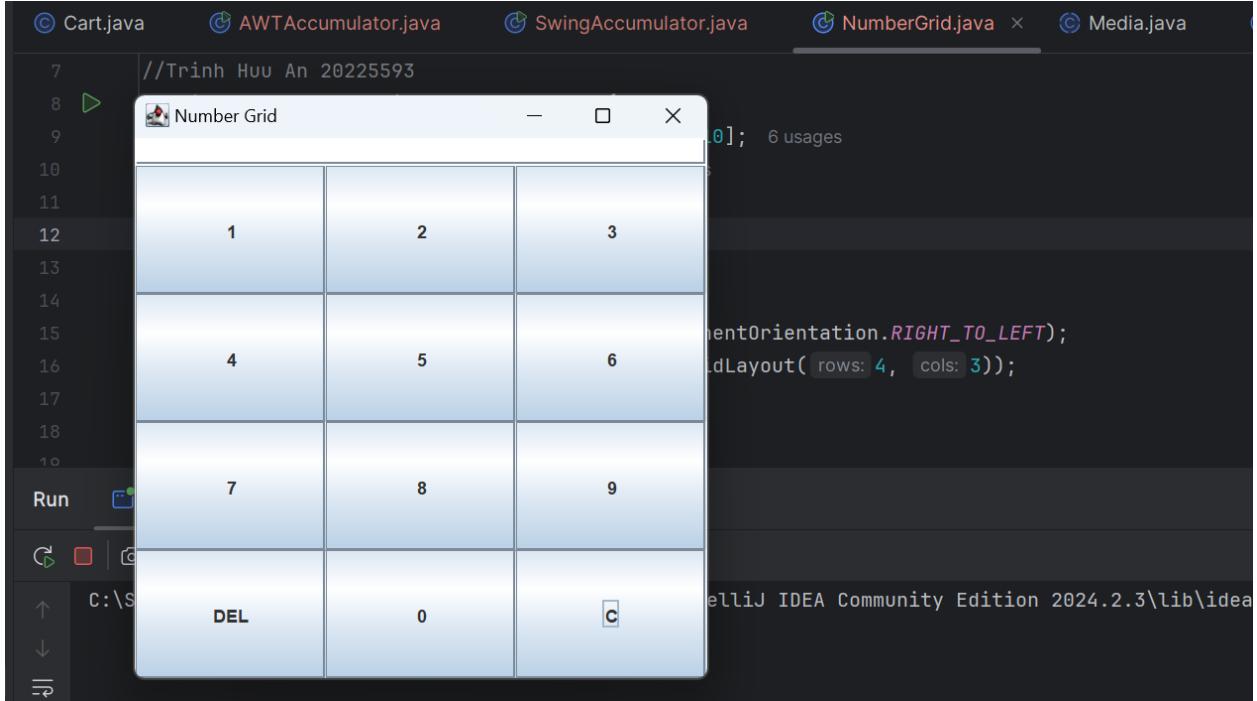
Figure 3: Source code of NumberGrid

## 2.2 Demo



The screenshot shows two instances of a Java application running in IntelliJ IDEA. The application displays a 4x3 grid of numbers (1-9) and a bottom row with 'DEL', '0', and 'c'. The code in the editor is identical for both runs.

```
//Trinh Huu An 20225593
Number Grid
1 2 3
4 5 6
7 8 9
DEL 0 c
```



*Figure 4: Demo NumberGrid*

### 3 Create a graphical user interface for AIMS with Swing

### 3.1 Create class StoreScreen

```
package hust.soict.hedspi.aims.screen;

import hust.soict.hedspi.aims.media.*;
import hust.soict.hedspi.aims.store.Store;

import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
//Trinh Huu An 20225593
public class StoreScreen extends JFrame {  ↳ Hữu An
    private Store store;  2 usages

    JPanel createNorth() {  1 usage  ↳ Hữu An
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }

    JMenuBar createMenuBar() {  1 usage  ↳ Hữu An
        JMenu menu = new JMenu("Options");
    }
}
```

```
JMenu smUpdateStore = new JMenu( s: "Update Store");
smUpdateStore.add(new JMenuItem( text: "Add Book"));
smUpdateStore.add(new JMenuItem( text: "Add CD"));
smUpdateStore.add(new JMenuItem( text: "Add DVD"));

menu.add(smUpdateStore);
menu.add(new JMenuItem( text: "View Store"));
menu.add(new JMenuItem( text: "View Cart"));

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

return menuBar;
}

 JPanel createHeader() { 1 usage  ± Hữu An
 JPanel header = new JPanel();
 header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

 JPanel createHeader() { 1 usage  ± Hữu An
 JPanel header = new JPanel();
 header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

 JLabel title = new JLabel( text: "AIMS");
 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));

 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;
}
```

```

JPanel createCenter() { 1 usage  ± Hữu An
    JPanel center = new JPanel();
    center.setLayout(new GridLayout( rows: 3, cols: 3, hgap: 2, vgap: 2));

    ArrayList<Media> mediaInStore = store.getItemsInStore();
    for (int i = 0; i < 9; i++) {
        MediaStore cell = new MediaStore(mediaInStore.get(i));
        center.add(cell);
    }

    return center;
}

public StoreScreen(Store store) { 1 usage  ± Hữu An
    this.store = store;

    Container cp = getContentPane();
    cp.setLayout(new BorderLayout());

    cp.add(createNorth(), BorderLayout.NORTH);
    cp.add(createCenter(), BorderLayout.CENTER);

    setVisible(true);
    setTitle("Store");
    setSize( width: 800, height: 600);
}

public static void main(String[] args) { ± Hữu An
    Store store = new Store();

    DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allers",
    DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "George Lucas",
    DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", director: "Guy Ritchie", length: 89,
    store.addMedia(dvd1); store.addMedia(dvd2); store.addMedia(dvd3);

    // CD + Track
    CompactDisc cd1 = new CompactDisc( title: "Thriller", category: "Pop", artist: "Michael Jackson", cost: 9.99f);
    Track track1 = new Track( title: "Billie Jean", length: 236);
    Track track2 = new Track( title: "Beat It", length: 252);
    Track track3 = new Track( title: "Man in the Mirror", length: 331);
    cd1.addTrack(track1); cd1.addTrack(track2); cd1.addTrack(track3);

    CompactDisc cd2 = new CompactDisc( title: "Kind of Blue", category: "Jazz", artist: "Miles Davis", cost: 15.50f);
    Track track4 = new Track( title: "So What", length: 431);
    Track track5 = new Track( title: "Freddie Freeloader", length: 539);
    cd2.addTrack(track4); cd2.addTrack(track5);

    CompactDisc cd3 = new CompactDisc( title: "Bohemian Rhapsody", category: "Rock", artist: "Queen", cost: 11.99f);
    Track track6 = new Track( title: "Bohemian Rhapsody", length: 355);
    Track track7 = new Track( title: "Killer Queen", length: 205);
    Track track8 = new Track( title: "Somebody to Love", length: 296);
    cd3.addTrack(track6); cd3.addTrack(track7); cd3.addTrack(track8);

    store.addMedia(cd1); store.addMedia(cd2); store.addMedia(cd3);
}

```

```

    // book
    Book book1 = new Book( title: "Hannibal", category: "Crime Thriller", cost: 8.99f);
    book1.addAuthor( authorName: "Thomas Harris");
    Book book2 = new Book( title: "Red Dragon", category: "Crime Thriller", cost: 9.11f);
    book2.addAuthor( authorName: "Thomas Harris");
    Book book3 = new Book( title: "The Martian", category: "Science Fiction", cost: 8.97f);
    book3.addAuthor( authorName: "Andy Weir");
    store.addMedia(book1); store.addMedia(book2); store.addMedia(book3);

    new StoreScreen(store);
}
}

```

Figure 5: Class StoreScreen

### 3.2 Create class MediaStore

```

package hust.soict.hedspi.aims.screen;

import hust.soict.hedspi.aims.media.Media;
import hust.soict.hedspi.aims.media.Playable;
|
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
//Trinh Huu An 20225593
public class MediaStore extends JPanel { 4 usages ▾ Hữu An
    private Media media; 1 usage

    public MediaStore(Media media) { 1 usage ▾ Hữu An
        this.media = media;
        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");
        addToCartButton.setPreferredSize(new Dimension( width: 120, height: 40));
        addToCartButton.setMaximumSize(new Dimension( width: 120, height: 40));
        addToCartButton.addActionListener(new ActionListener() { ▾ Hữu An
            @Override ▾ Hữu An
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog( parentComponent: MediaStore.this, message: media.getTitle() + " added to cart.");
            }
        });
        container.add(addToCartButton);

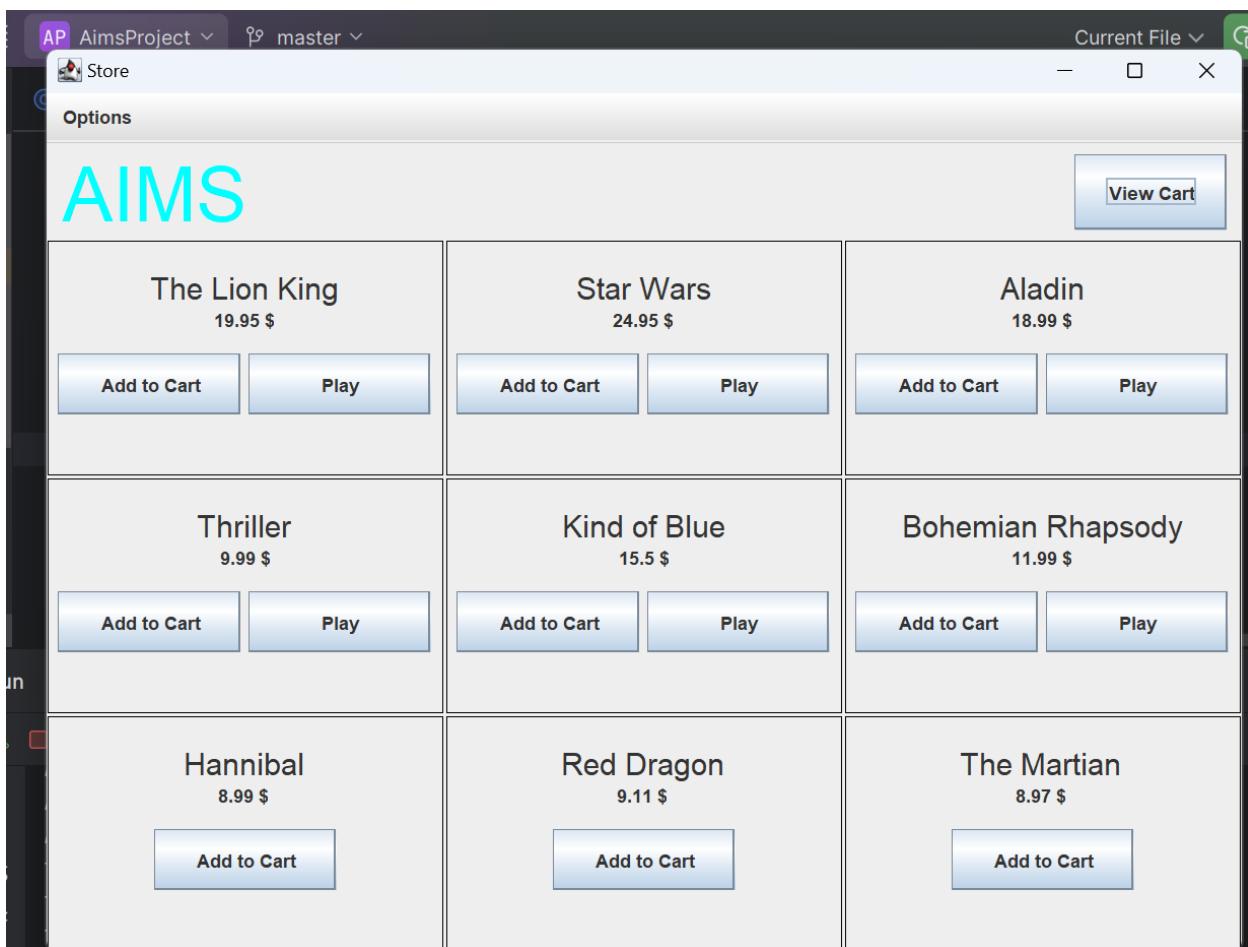
        container.add(addToCartButton);
    }
}

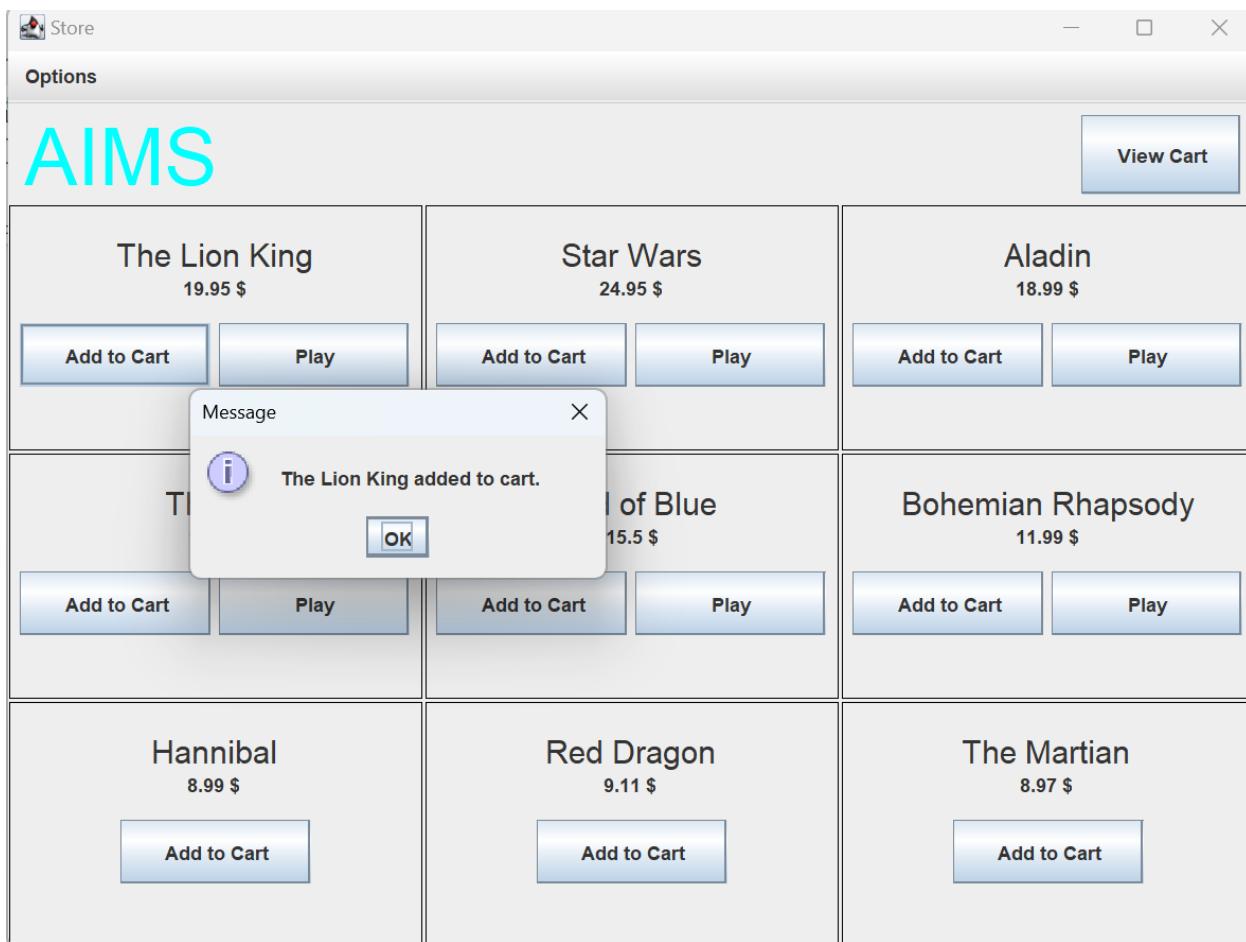
```

```
if (media instanceof Playable) {  
    JButton playButton = new JButton("Play");  
    playButton.setPreferredSize(new Dimension(120, 40));  
    playButton.setMaximumSize(new Dimension(120, 40));  
    playButton.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            JOptionPane.showMessageDialog(parentComponent: MediaStore.this, message: "Playing " + media.getTitle());  
        }  
    });  
    container.add(playButton);  
}  
  
this.add(Box.createVerticalGlue());  
this.add(title);  
this.add(cost);  
this.add(Box.createVerticalStrut(10));  
this.add(container);  
this.add(Box.createVerticalGlue());  
  
this.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
this.setPreferredSize(new Dimension(200, 250));  
}  
}
```

Figure 6: Class MediaStore

### 3.3 Demo





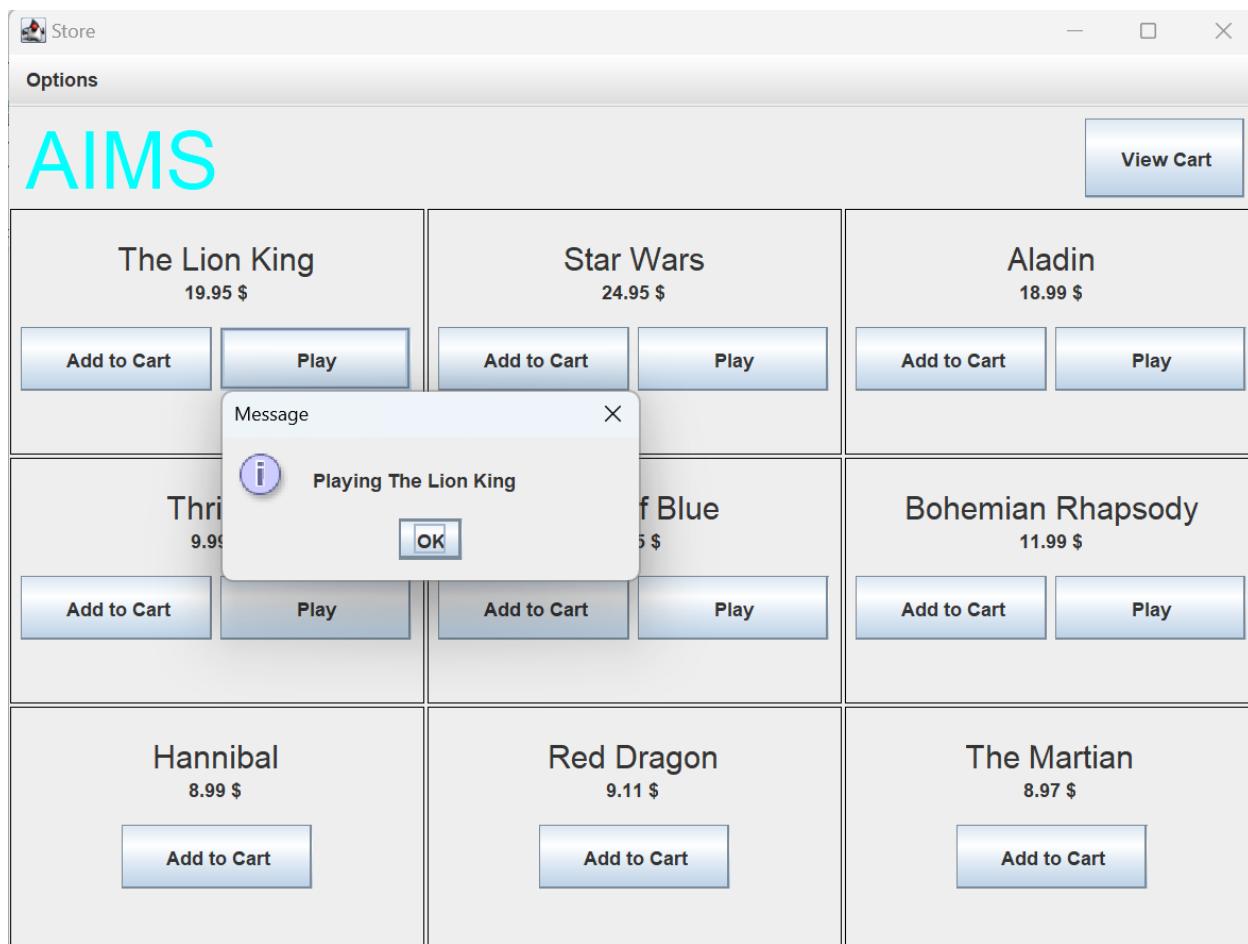


Figure 7: Demo

## 4 JavaFX API

### 4.1 Create class Painter

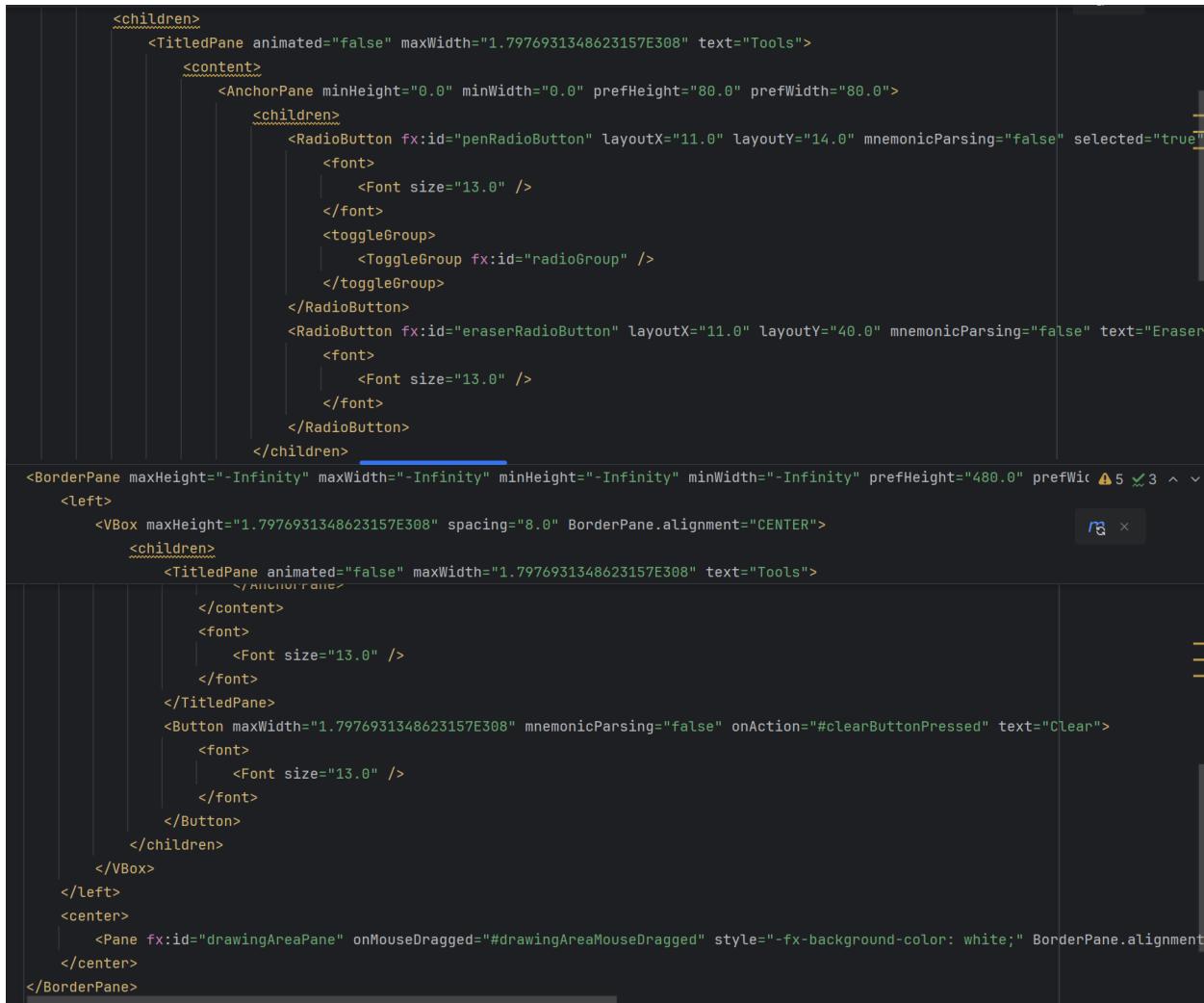
```
package hust.soict.hedspi.javafx;
|
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;
//Trinh Huu An 20225593
public class Painter extends Application { new *
    @Override new *
    public void start(Stage stage) {
        try {
            Parent root = FXMLLoader.load(getClass().getResource( name: "/hust/soict/hedspi/javafx/Painter.fxml"));
            Scene scene = new Scene(root);
            stage.setTitle("Painter");
            stage.setScene(scene);
            stage.show();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public static void main(String[] args) { new *
        public static void main(String[] args) { new *
            launch(args);
        }
    }
}
```

Figure 8: Class Painter

### 4.2 Create Painter.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import javafx.geometry.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>
|
<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640
<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">
                <content>
                    <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="80.0" prefWidth="80.0">

```



```

<children>
    <TitledPane animated="false" maxWidth="1.7976931348623157E308" text="Tools">
        <content>
            <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="80.0" prefWidth="80.0">
                <children>
                    <RadioButton fx:id="penRadioButton" layoutX="11.0" layoutY="14.0" mnemonicParsing="false" selected="true">
                        <font>
                            <Font size="13.0" />
                        </font>
                        <toggleGroup>
                            <ToggleGroup fx:id="radioGroup" />
                        </toggleGroup>
                    </RadioButton>
                    <RadioButton fx:id="eraserRadioButton" layoutX="11.0" layoutY="40.0" mnemonicParsing="false" text="Eraser">
                        <font>
                            <Font size="13.0" />
                        </font>
                    </RadioButton>
                </children>
            </AnchorPane>
        <left>
            <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
                <children>
                    <TitledPane animated="false" maxWidth="1.7976931348623157E308" text="Tools">
                        <content>
                            <font>
                                <Font size="13.0" />
                            </font>
                        </content>
                        <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear">
                            <font>
                                <Font size="13.0" />
                            </font>
                        </Button>
                    </TitledPane>
                </children>
            </VBox>
        </left>
        <center>
            <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
        </center>
    </TitledPane>

```

Figure 9: Painter.fxml

### 4.3 Create class PainterController

```
package hust.soict.hedspi.javafx;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
//Trịnh Hữu An 20225593
public class PainterController { new *
| @FXML
| private RadioButton penRadioButton;
|
| @FXML
| private Pane drawingAreaPane;
|
| @FXML
| private RadioButton eraserRadioButton;
|
| @FXML
| private ToggleGroup radioGroup;
|
| @FXML new *
| void drawingAreaMouseDragged(MouseEvent event) {
|     Color color = penRadioButton.isSelected() ? Color.BLACK : Color.WHITE;
|     Circle newCircle = new Circle(event.getX(), event.getY(), 4, color);
|     drawingAreaPane.getChildren().add(newCircle);
| }
|
| @FXML new *
| void clearButtonPressed(ActionEvent event) { drawingAreaPane.getChildren().clear(); }
}

}
```

Figure 10: PainterController

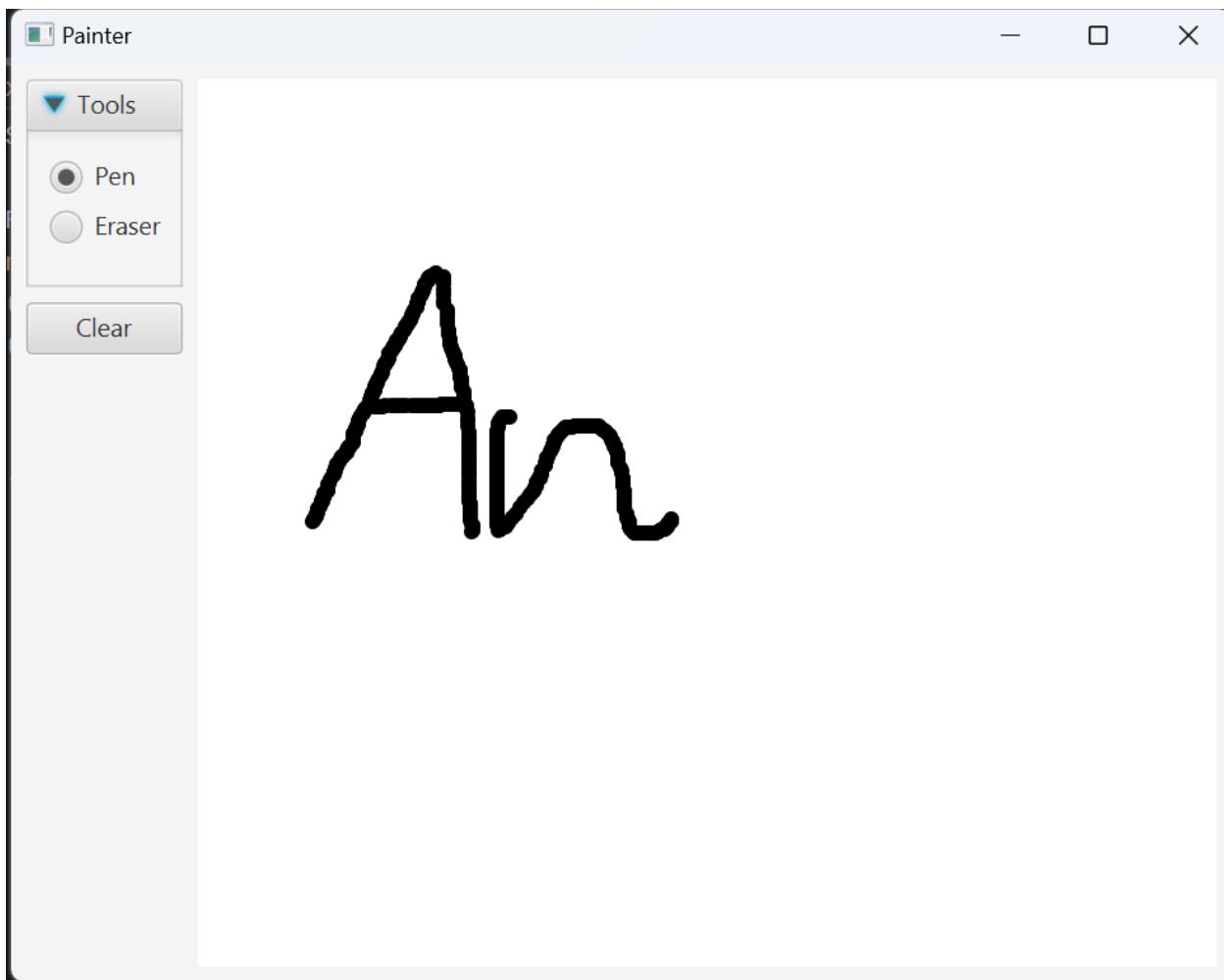


Figure 11: Use pen

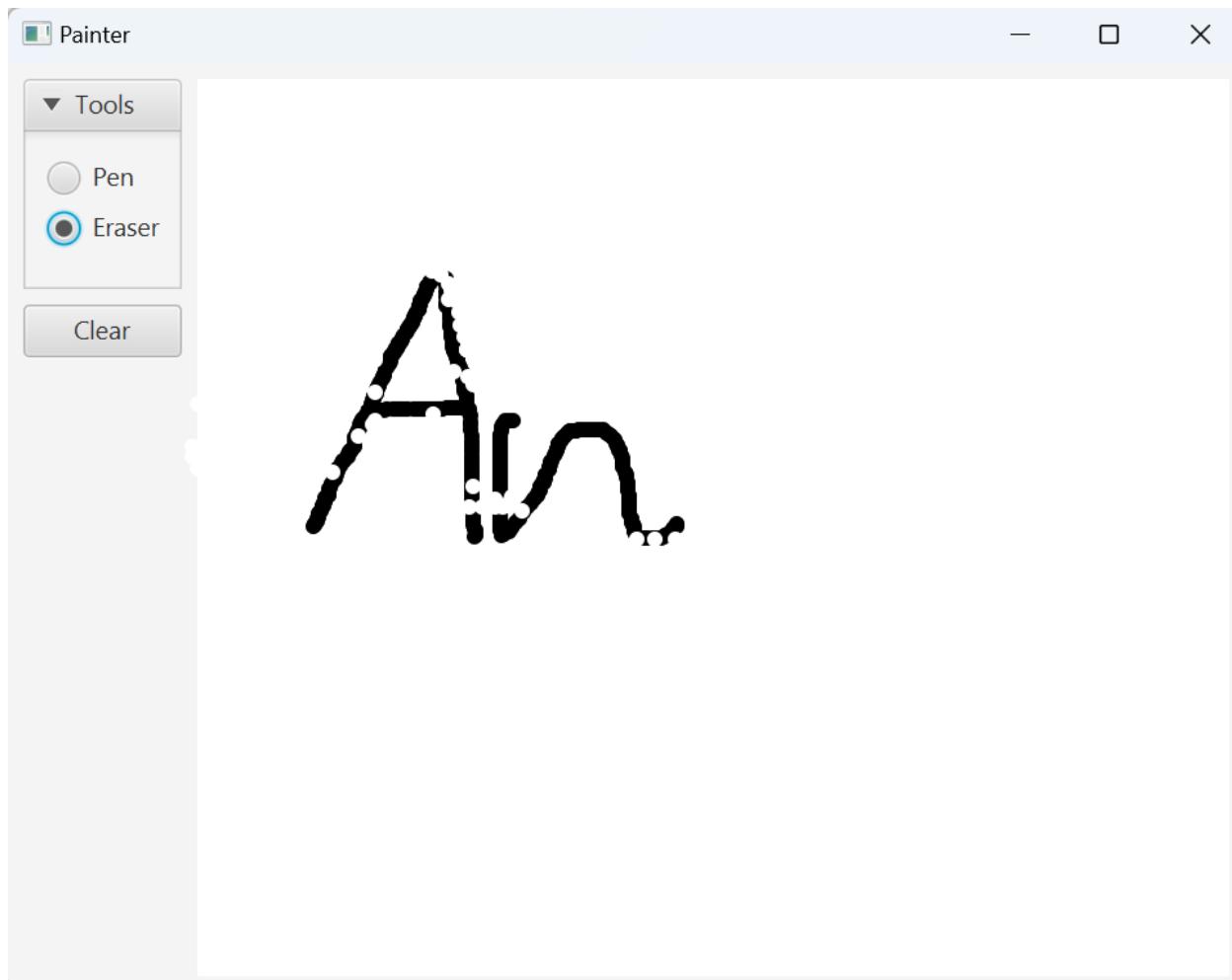


Figure 12: Use eraser

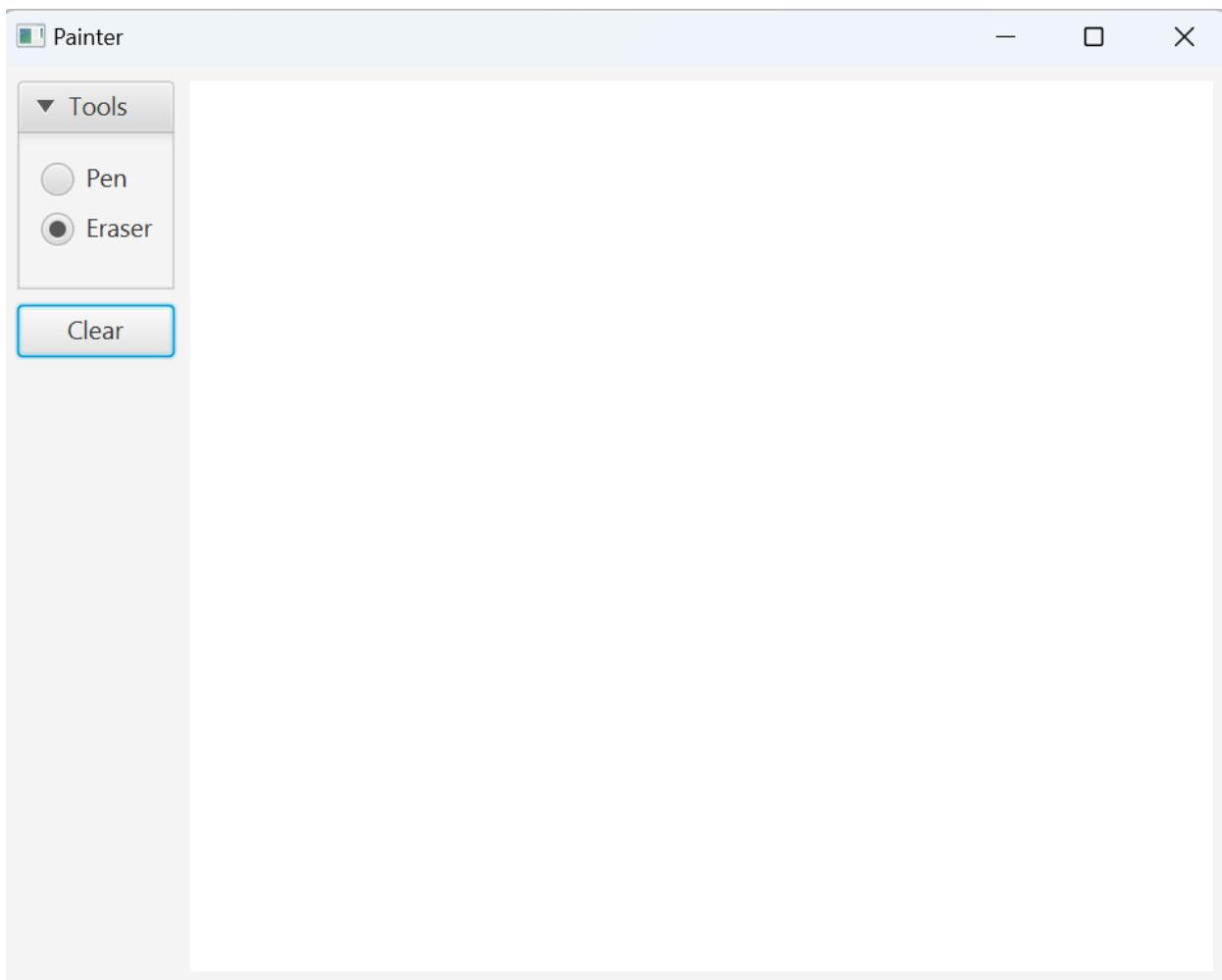


Figure 13: Clear Button

## 5 View Cart Screen

### 5.1 Create cart.fxml

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

<?import javafx.geometry.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<BorderPane prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" 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>
                            </items>
                        </Menu>
                    </menus>
                </MenuBar>
                <Label text="CART" textFill="AQUA">
                    <font>
                        <Font size="50.0" />
                    </font>
                    <padding>
                        <Insets left="10.0" />
                    </padding>
                </Label>
            </children>
        </VBox>
    </top>
</BorderPane>
```

```

<center>
    <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
        <padding>
            <Insets left="10.0" />
        </padding>
        <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">
                        <font>
                            <Font size="13.0" />
                        </font>
                    </TextField>
                    <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" mnemonicParsing="false" onAction="#btnSortPressed" text="Sort by Title">
                        <HBox.margin>
                            <Insets left="250.0" />
                        </HBox.margin>
                    </Button>
                    <Button mnemonicParsing="false" onAction="#btnSortPressed" text="Sort by Cost" />
                </children>
            </HBox>
            <TableView fx:id="tblMedia">
                <columns>
                    <TableColumn fx:id="colMediaTitle" prefWidth="75.0" text="Title" />
                <TableView fx:id="tblMedia">
                    <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>
                    <VBox.margin>
                        <Insets />
                    </VBox.margin>
                </TableView>
                <HBox alignment="TOP_RIGHT" prefHeight="100.0" prefWidth="200.0" spacing="10.0">
                    <children>
                        <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed" text="Play" />
                        <Button fx:id="btnRemove" mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
                        <Button fx:id="btnDetails" mnemonicParsing="false" onAction="#btnDetailsPressed" text="Details" />
                    </children>
                </HBox>
            </TableView>
        </children>
    </VBox>

```

```

<HBox alignment="TOP_RIGHT" prefHeight="100.0" prefWidth="200.0" spacing="10.0">
    <children>
        <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed" text="Play" />
        <Button fx:id="btnRemove" mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
        <Button fx:id="btnDetails" mnemonicParsing="false" onAction="#btnDetailsPressed" text="Details" />
    </children>
    <padding>
        <Insets top="10.0" />
    </padding>
</HBox>
</VBox>
</children>
</VBox>
</center>
</right>
<VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
    <padding>
        <Insets left="20.0" right="20.0" top="100.0" />
    </padding>
    <children>
        <HBox alignment="CENTER" spacing="10.0">
            <children>
                <HBox alignment="CENTER" spacing="10.0">
                    <children>
                        <Label text="Total:>">
                            <font>
                                <Font size="24.0" />
                            </font>
                        </Label>
                        <Label fx:id="lblCost" text="0 $" textFill="AQUA">
                            <font>
                                <Font size="24.0" />
                            </font>
                        </Label>
                    </children>
                </HBox>
                <Button fx:id="btnPlaceOrder" mnemonicParsing="false" onAction="#btnPlaceOrderPressed" style="-fx-background-color:
                    <font>
                        <Font size="24.0" />
                    </font>
                    <~/font>
                    <~color>
                        <Font size="24.0" />
                    </font>
                    <VBox.margin>
                        <Insets top="40.0" />
                    </VBox.margin>
                </Button>
            </children>
        </VBox>
    </children>
</VBox>
</right>
</BorderPane>

```

Figure 14: cart.fxml

## 5.2 Create class CartScreen

```

1 package hust.soict.hedspi.aims.screen;
2
3 import hust.soict.hedspi.aims.cart.*;
4 import hust.soict.hedspi.aims.media.Book;
5 import hust.soict.hedspi.aims.media.CompactDisc;
6 import hust.soict.hedspi.aims.media.DigitalVideoDisc;
7 import javafx.application.Platform;
8 import javafx.embed.swing.JFXPanel;
9 import javafx.fxml.FXMLLoader;
10 import javafx.scene.Parent;
11 import javafx.scene.Scene;
12
13 import javax.swing.*;
14 import java.awt.*;
15 import java.io.IOException;
16 //Trịnh Hữu An 20225593
17 public class CartScreen extends JFrame {
18     private Cart cart; 1 usage
19
20     public CartScreen(Cart cart) { 2 usages
21         super();
22         this.cart = cart;
23         JFXPanel fxPanel = new JFXPanel();
24         this.add(fxPanel);
25
26
27     this.setSize(new Dimension( width: 1024, height: 768));
28     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29     Platform.runLater(new Runnable() {
30
31         @Override
32         public void run() {
33             try {
34                 FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/hust/soict/hedspi/aims/screen/cart.fxml"));
35                 CartScreenController controller = new CartScreenController(cart);
36                 loader.setController(controller);
37                 Parent root = loader.load();
38                 Scene scene = new Scene(root);
39                 fxPanel.setScene(scene);
40
41             } catch (IOException e) {
42                 e.printStackTrace();
43             }
44         }
45     });
46 };

```

```

    }

    public static void main(String[] args) {
        Cart cart = new Cart();
        // Compact Discs
        CompactDisc cd = new CompactDisc( title: "Melodic Journey", category: "Pop", cost: 11.49f, director: "Liam Rhodes", length: 180, artist: "Liam Rhodes");
        CompactDisc cd1 = new CompactDisc( title: "Classical Masterpieces", category: "Classical", cost: 15.99f, director: "Various Artists", length: 200, artist: "Various Artists");
        CompactDisc cd2 = new CompactDisc( title: "Electronic Beats", category: "Electronic", cost: 9.99f, director: "DJ Zane", length: 220, artist: "DJ Zane");
        CompactDisc cd3 = new CompactDisc( title: "Smooth Jazz Nights", category: "Jazz", cost: 13.49f, director: "John Coltrane", length: 200, artist: "John Coltrane");
        CompactDisc cd4 = new CompactDisc( title: "Greatest Hits 2024", category: "Pop", cost: 10.49f, director: "Ariana Grande", length: 150, artist: "Ariana Grande");

        // Digital Video Discs
        DigitalVideoDisc dvd = new DigitalVideoDisc( title: "The Dark Knight", category: "Action", cost: 14.99f, director: "Christopher Nolan", length: 120);
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "Avatar", category: "Sci-fi", cost: 18.99f, director: "James Cameron", length: 162);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "The Grand Budapest Hotel", category: "Comedy", cost: 12.49f, director: "Wes Anderson", length: 90);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", cost: 16.99f, director: "Disney", length: 88);
        DigitalVideoDisc dvd4 = new DigitalVideoDisc( title: "The Avengers", category: "Action", cost: 19.49f, director: "Joss Whedon", length: 140);

        // Books
        Book book = new Book( title: "The Silent Patient", category: "Thriller", cost: 12.99f);
        Book book1 = new Book( title: "The Great Gatsby", category: "Classic", cost: 9.49f);
        Book book2 = new Book( title: "1984", category: "Dystopian", cost: 8.99f);
        Book book3 = new Book( title: "Moby Dick", category: "Adventure", cost: 11.29f);
        Book book4 = new Book( title: "The Catcher in the Rye", category: "Classic", cost: 7.79f);

        Cart.addMedia(cd);
        Cart.addMedia(cd1);
        Cart.addMedia(cd2);
        Cart.addMedia(cd3);
        Cart.addMedia(cd4);

        Cart.addMedia(dvd);
        Cart.addMedia(dvd2);
        Cart.addMedia(dvd3);
        Cart.addMedia(dvd4);
        Cart.addMedia(dvd1);

        Cart.addMedia(book);
        Cart.addMedia(book1);
        Cart.addMedia(book2);
        Cart.addMedia(book3);
        Cart.addMedia(book4);

        new CartScreen(cart);
    }
}

```

Figure 15: Create CartScreen

### 5.3 Create class CartScreenController

```

@FXML 1 usage
private TableColumn<Media, String> colMediaCategory;
@FXML 3 usages
private Button btnPlay;
@FXML 2 usages
private Button btnRemove;
@FXML 2 usages
private Button btnDetails;
@FXML 1 usage
private RadioButton radioBtnFilterId;
@FXML no usages
private RadioButton radioBtnFilterTitle;
@FXML 1 usage
private TextField tfFilter;
@FXML 13 usages
private TableView<Media> tblMedia;

@FXML no usages
private ToggleGroup filterCategory;

@FXML no usages
private Button btnPlaceOrder;
@FXML 1 usage
private Label lblCost;
public CartScreenController(Cart cart) { 1 usage
    super();
    this.cart = cart;
}
@FXML no usages
private void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "title"));
    colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "cost"));
    tblMedia.setItems(cart.getItemsOrdered());
    btnDetails.setVisible(false);
    btnPlay.setVisible(false);
    btnRemove.setVisible(false);
    tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {
        @Override
        public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
            if (newValue != null) {
                updateButtonBar(newValue);
            }
        }
    });
}

```

```

        }
    }
});

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

updateCost();
}

void updateCost() { lblCost.setText(String.format("%.2f $", cart.totalCost())); }

void updateButtonBar(Media media) { usage
    btnRemove.setVisible(true);
    if (media instanceof Playable) {
        btnPlay.setVisible(true);
    } else {
        btnPlay.setVisible(false);
        btnPlay.setVisible(false);
    }
    btnDetails.setVisible(true);
}

void showFilteredMedia(String input) { usage
    if (input == "") {
        tblMedia.setItems(cart.getItemsOrdered());
        return;
    }

    FilteredList<Media> filteredList = new FilteredList<>(cart.getItemsOrdered());
    if (radioBtnFilterId.isSelected())
        filteredList.setPredicate((it) -> it.getId() == Integer.parseInt(input));
    else
        filteredList.setPredicate((it) -> it.isMatch(input));

    tblMedia.setItems(filteredList);
}
}

```

```

@FXML no usages
void btnSortPressed(ActionEvent event) {
    tblMedia.getSortOrder().clear();

    colMediaCost.setSortType(TableColumn.SortType.DESCENDING);

    if (event.getSource().toString().split("::")[1].equals("Sort by Title")) {
        tblMedia.getSortOrder().add(colMediaTitle);
        tblMedia.getSortOrder().add(colMediaCost);
    } else {
        tblMedia.getSortOrder().add(colMediaCost);
        tblMedia.getSortOrder().add(colMediaTitle);
    }

    tblMedia.sort();
}

@FXML no usages
void btnDetailsPressed(ActionEvent event) { new DetailScreen(tblMedia.getSelectionModel().getSelectedItem()); }

@FXML no usages
void btnPlayPressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    updateCost();
}

@FXML no usages
void btnPlaceOrderPressed(ActionEvent event) {
    if (cart.getItemsOrdered().isEmpty())
        JOptionPane.showMessageDialog( parentComponent: null, message: "Cart is empty!", title: "Error", JOptionPane.ERROR_MESSAGE);
    else {
        new PlaceOrderScreen();
        updateCost();
    }
}

@FXML no usages
void menuAddBook(ActionEvent event) { new AddBookToStoreScreen(); }

@FXML no usages
void menuAddCd(ActionEvent event) { new AddCompactDiscToStoreScreen(); }

@FXML no usages
void menuAddDvd(ActionEvent event) { new AddDigitalVideoDiscToStoreScreen(); }

@FXML no usages
void menuViewStore(ActionEvent event) {
    Aims.closeCartScreen();
    Aims.openStoreScreen();
}
}

```

Figure 16: Create CartScreenController

## 5.4 Demo

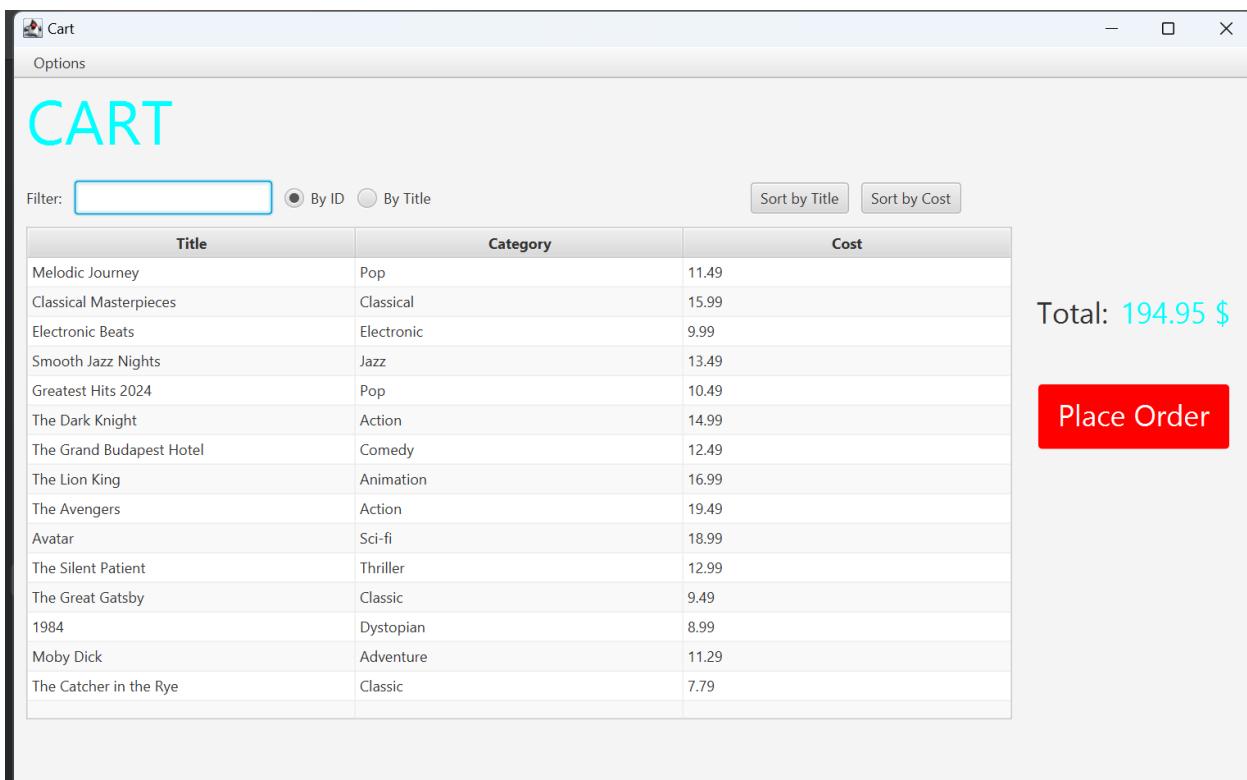


Figure 17: Demo

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

## 6.1 Edit class CartScreenController

```

@FXML no usages
private void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
    colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, String>("cost"));
    tblMedia.setItems(cart.getItemsOrdered());
    btnDetails.setVisible(false);
    btnPlay.setVisible(false);
    btnRemove.setVisible(false);
    tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {
        @Override
        public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
            if (newValue != null) {
                updateButtonBar(newValue);
            }
        }
    });
    tfFilter.textProperty().addListener(new ChangeListener<String>() {
        @Override
        public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
            showFilteredMedia(newValue);
        }
    });
    updateCost();
}

void updateCost() { lblCost.setText(String.format("%.2f $", cart.totalCost())); }

void updateButtonBar(Media media) { 1 usage
    btnRemove.setVisible(true);
    if (media instanceof Playable) {
        btnPlay.setVisible(true);
    } else {
        btnPlay.setVisible(false);
    }
    btnDetails.setVisible(true);
}

void showFilteredMedia(String input) { 1 usage
    if (input == "") {
        tblMedia.setItems(cart.getItemsOrdered());
        return;
    }
}

```

Figure 18: Edit class CartScreenController

## 6.2 Demo

**CART**

Filter:   By ID  By Title Sort by Title Sort by Cost

| Title                    | Category   | Cost  |
|--------------------------|------------|-------|
| Melodic Journey          | Pop        | 11.49 |
| Classical Masterpieces   | Classical  | 15.99 |
| Electronic Beats         | Electronic | 9.99  |
| Smooth Jazz Nights       | Jazz       | 13.49 |
| Greatest Hits 2024       | Pop        | 10.49 |
| Vibes of Nature          | Relaxation | 8.49  |
| Summer Beats             | Dance      | 12.99 |
| Indie Hits               | Indie      | 14.29 |
| Retro Classics           | Classical  | 16.99 |
| Top 100                  | Pop        | 17.99 |
| Ultimate Jazz Collection | Jazz       | 20.99 |
| The Dark Knight          | Action     | 14.99 |
| Avatar                   | Sci-fi     | 18.99 |
| The Grand Budapest Hotel | Comedy     | 12.49 |
| The Lion King            | Animation  | 16.99 |
| The Avengers             | Action     | 19.49 |

Total: 308.70 \$

Place Order

**CART**

Filter:   By ID  By Title Sort by Title Sort by Cost

| Title                    | Category   | Cost  |
|--------------------------|------------|-------|
| Greatest Hits 2024       | Pop        | 10.49 |
| Vibes of Nature          | Relaxation | 8.49  |
| Summer Beats             | Dance      | 12.99 |
| Indie Hits               | Indie      | 14.29 |
| Retro Classics           | Classical  | 16.99 |
| Top 100                  | Pop        | 17.99 |
| Ultimate Jazz Collection | Jazz       | 20.99 |
| The Dark Knight          | Action     | 14.99 |
| Avatar                   | Sci-fi     | 18.99 |
| The Grand Budapest Hotel | Comedy     | 12.49 |
| The Lion King            | Animation  | 16.99 |
| The Avengers             | Action     | 19.49 |
| The Matrix               | Sci-fi     | 14.49 |
| Interstellar             | Sci-fi     | 22.99 |
| The Godfather            | Drama      | 18.29 |
| Pulp Fiction             | Crime      | 16.79 |

Total: 308.70 \$

Place Order

Figure 19: Demo

## 7 Deleting a media

### 7.1 Code

```
@FXML no usages
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    updateCost();
}
```

Figure 20: btnRemovePressed method

### 7.2 Demo

The screenshot shows a JavaFX application window titled "CART". At the top, there is a menu bar with "Cart" and "Options" items. Below the menu is a large title "CART" in blue. Underneath the title is a search bar labeled "Filter:" with a text input field and two radio buttons: "By ID" (selected) and "By Title". To the right of the search bar are two buttons: "Sort by Title" and "Sort by Cost". The main area contains a table with three columns: "Title", "Category", and "Cost". The table lists various media items with their details. At the bottom right of the table are three buttons: "Play", "Remove", and "Details".

| Title                    | Category   | Cost  |
|--------------------------|------------|-------|
| Greatest Hits 2024       | Pop        | 10.49 |
| Vibes of Nature          | Relaxation | 8.49  |
| Summer Beats             | Dance      | 12.99 |
| Indie Hits               | Indie      | 14.29 |
| Retro Classics           | Classical  | 16.99 |
| Top 100                  | Pop        | 17.99 |
| Ultimate Jazz Collection | Jazz       | 20.99 |
| The Dark Knight          | Action     | 14.99 |
| Avatar                   | Sci-fi     | 18.99 |
| The Grand Budapest Hotel | Comedy     | 12.49 |
| The Lion King            | Animation  | 16.99 |
| The Avengers             | Action     | 19.49 |
| The Matrix               | Sci-fi     | 14.49 |
| Interstellar             | Sci-fi     | 22.99 |
| The Godfather            | Drama      | 18.29 |
| Pulp Fiction             | Crime      | 16.79 |

Options

# CART

Filter:   By ID  By Title Sort by Title  Sort by Cost

| Title                    | Category   | Cost  |
|--------------------------|------------|-------|
| Greatest Hits 2024       | Pop        | 10.49 |
| Vibes of Nature          | Relaxation | 8.49  |
| Summer Beats             | Dance      | 12.99 |
| Indie Hits               | Indie      | 14.29 |
| Retro Classics           | Classical  | 16.99 |
| Top 100                  | Pop        | 17.99 |
| Ultimate Jazz Collection | Jazz       | 20.99 |
| The Dark Knight          | Action     | 14.99 |
| Avatar                   | Sci-fi     | 18.99 |
| The Grand Budapest Hotel | Comedy     | 12.49 |
| The Lion King            | Animation  | 16.99 |
| The Avengers             | Action     | 19.49 |
| The Matrix               | Sci-fi     | 14.49 |
| Interstellar             | Sci-fi     | 22.99 |
| The Godfather            | Drama      | 18.29 |
| Inception                | Thriller   | 17.99 |

Figure 21: Demo

## 8 Complete the Aims GUI application

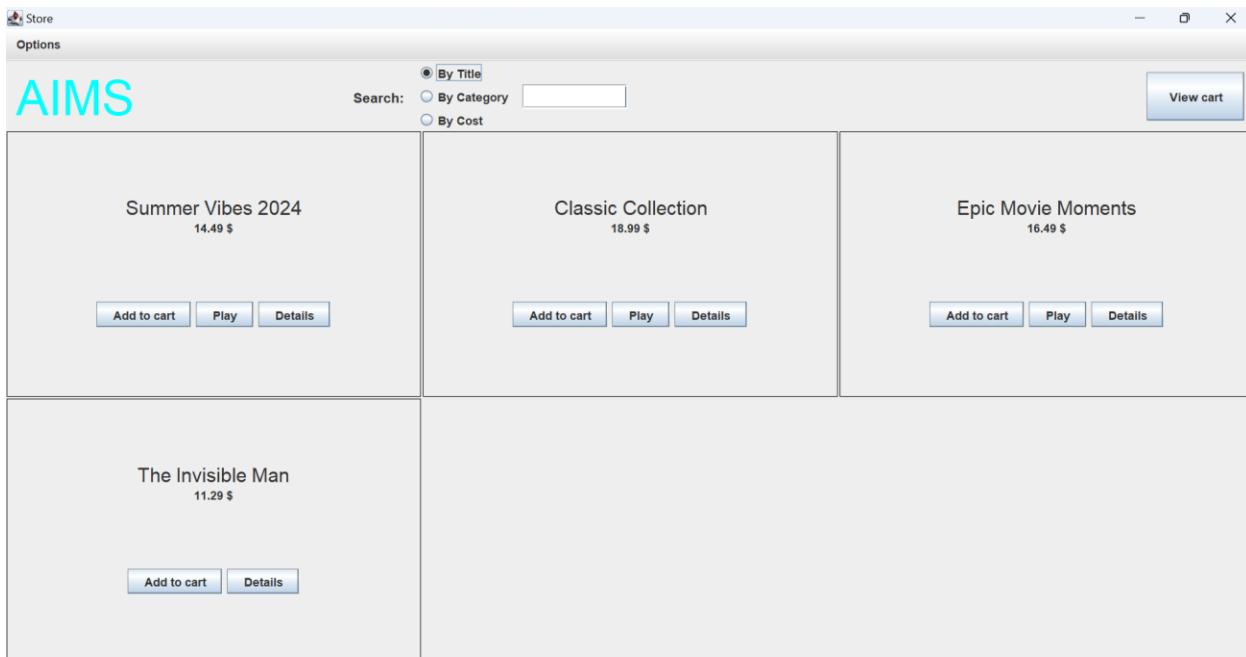


Figure 22: Store before add book

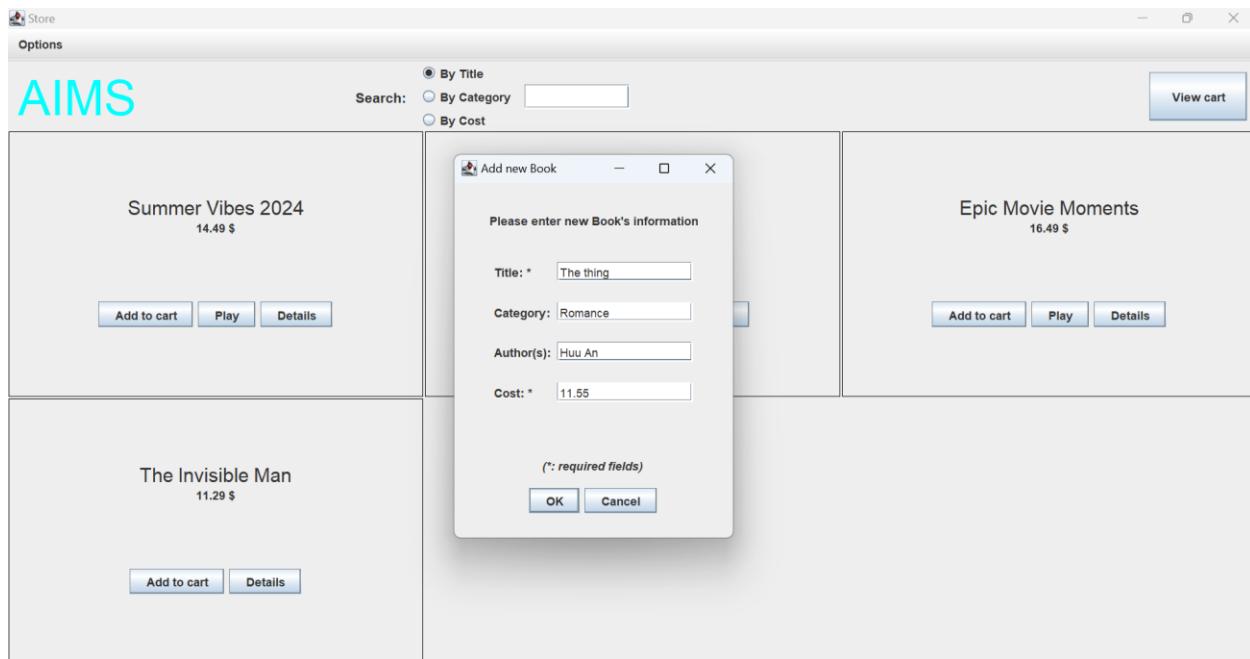


Figure 23: Add book

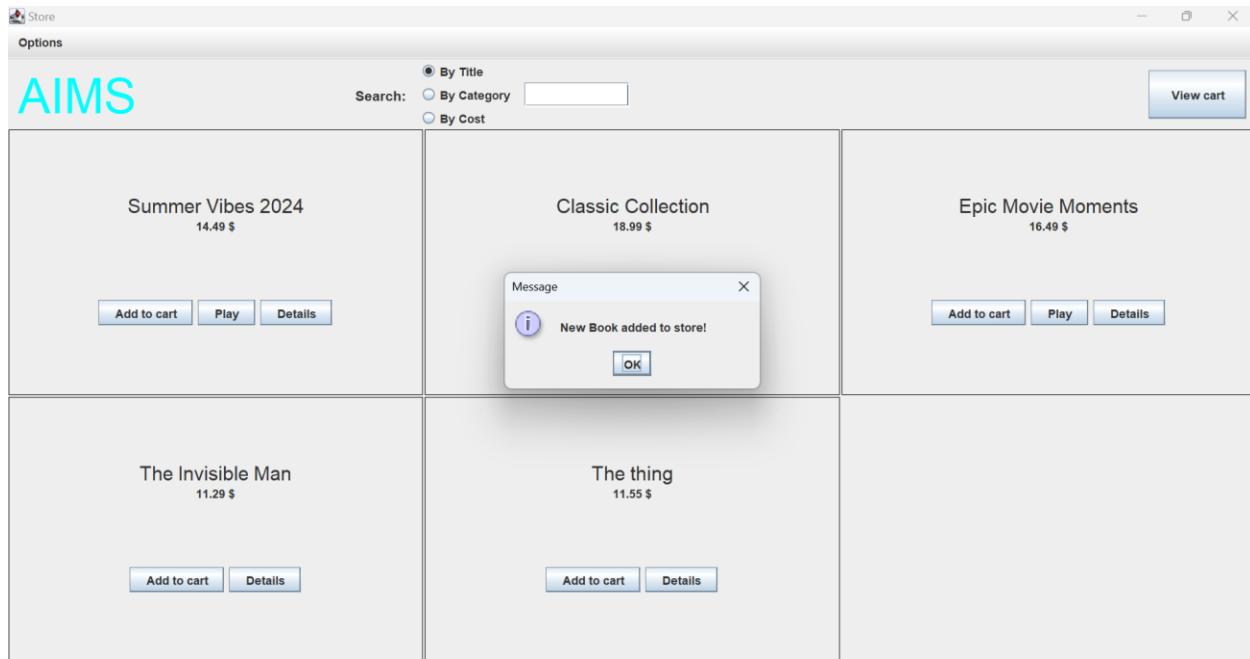


Figure 24: Store after add book

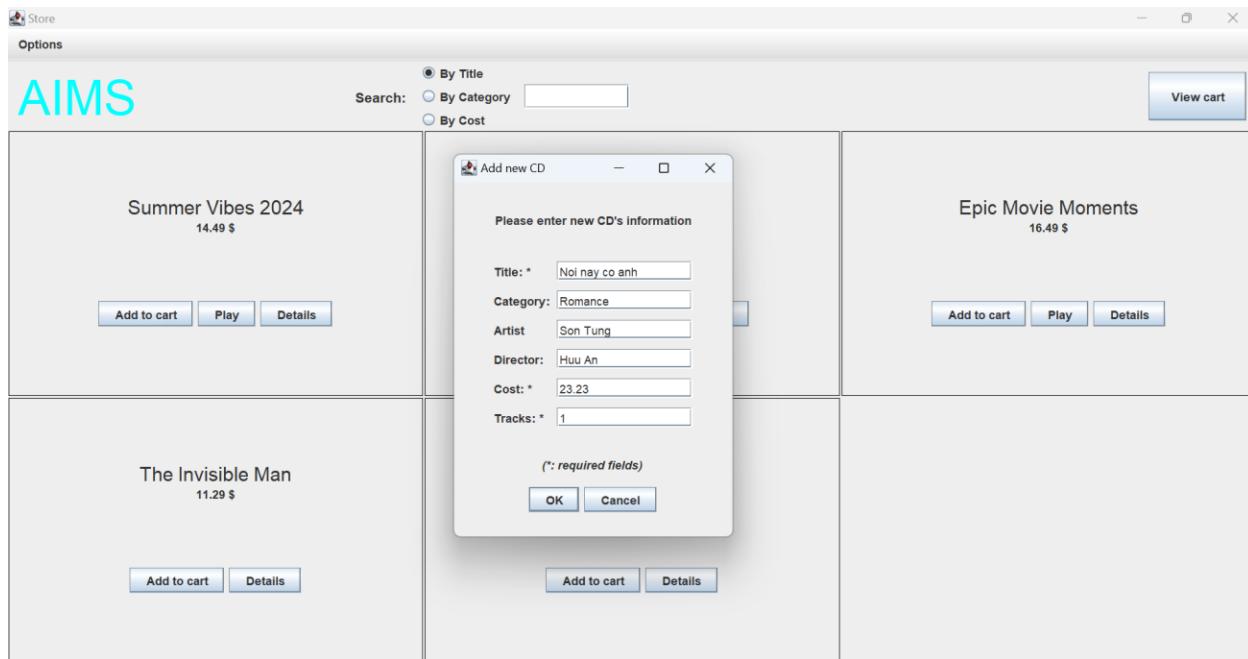


Figure 25: Add CD

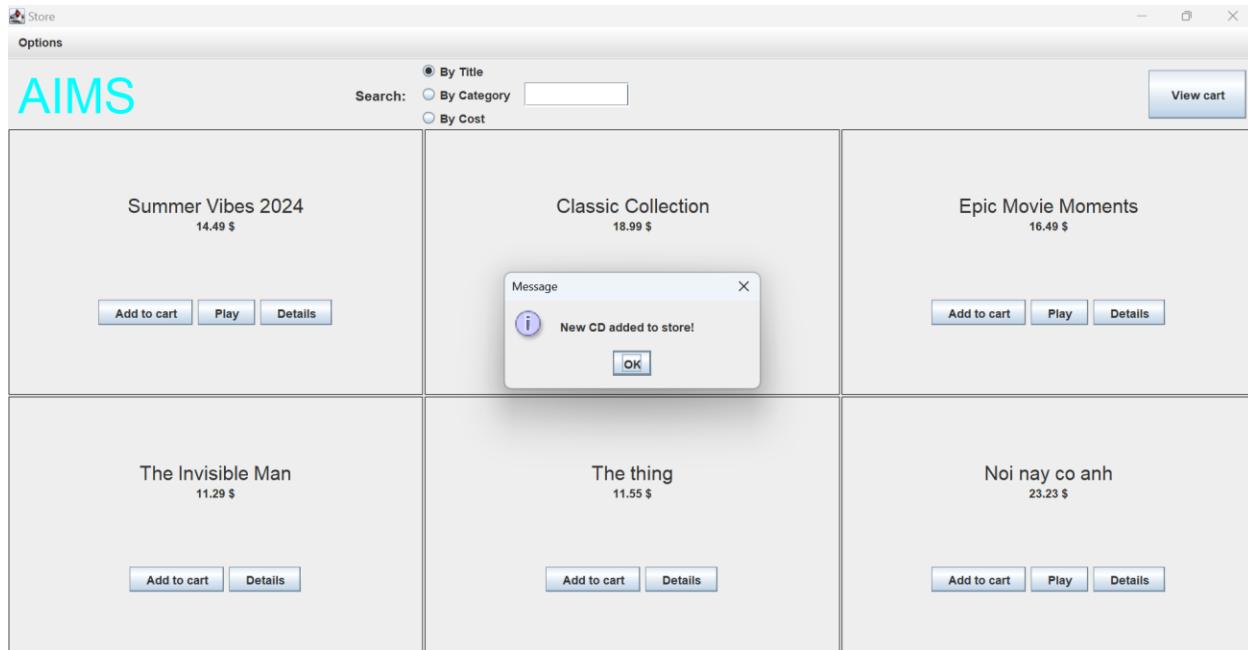


Figure 26: Store after add CD

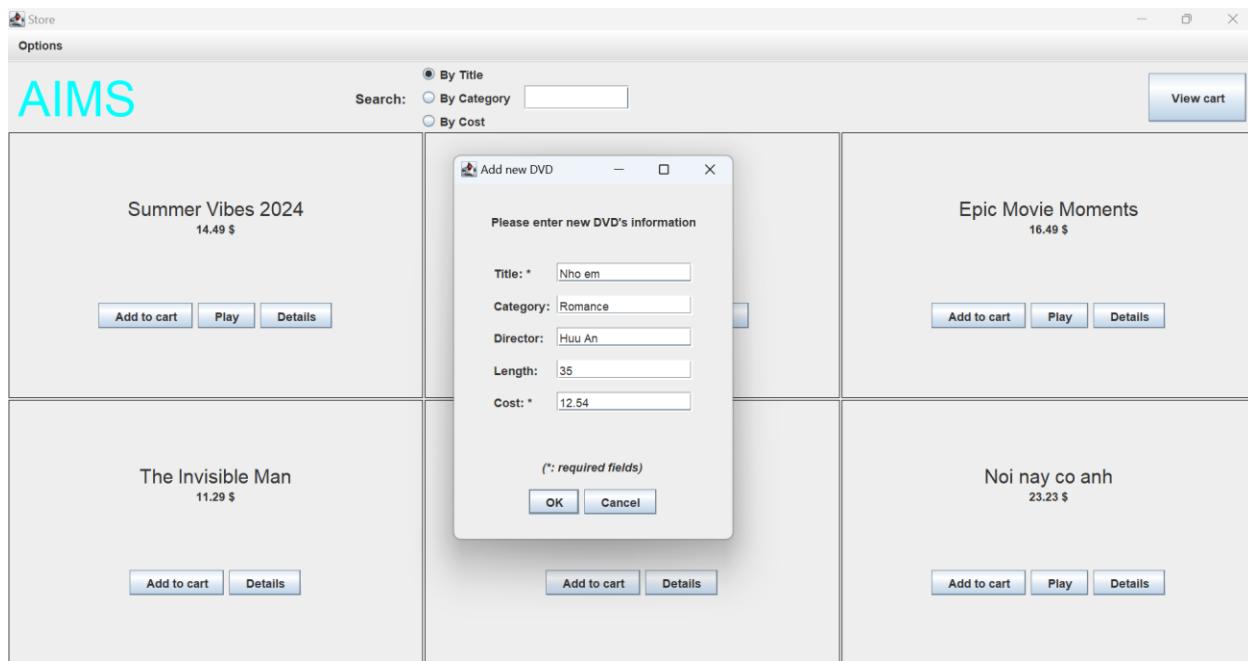


Figure 27: Add DVD

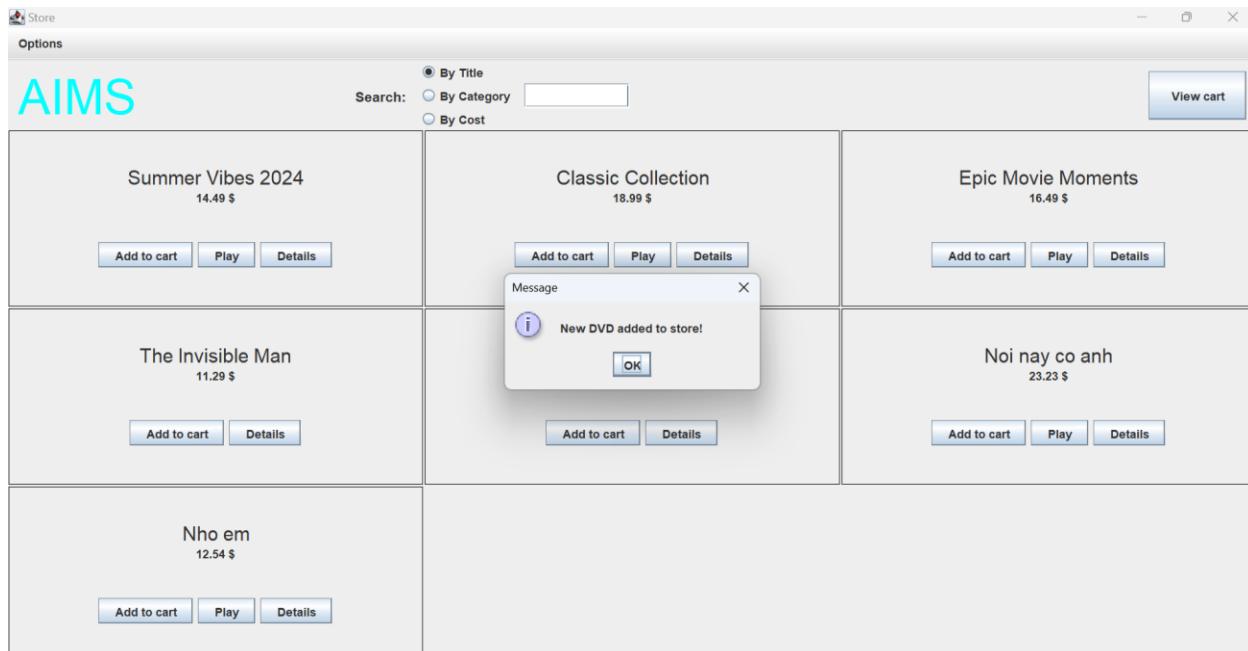
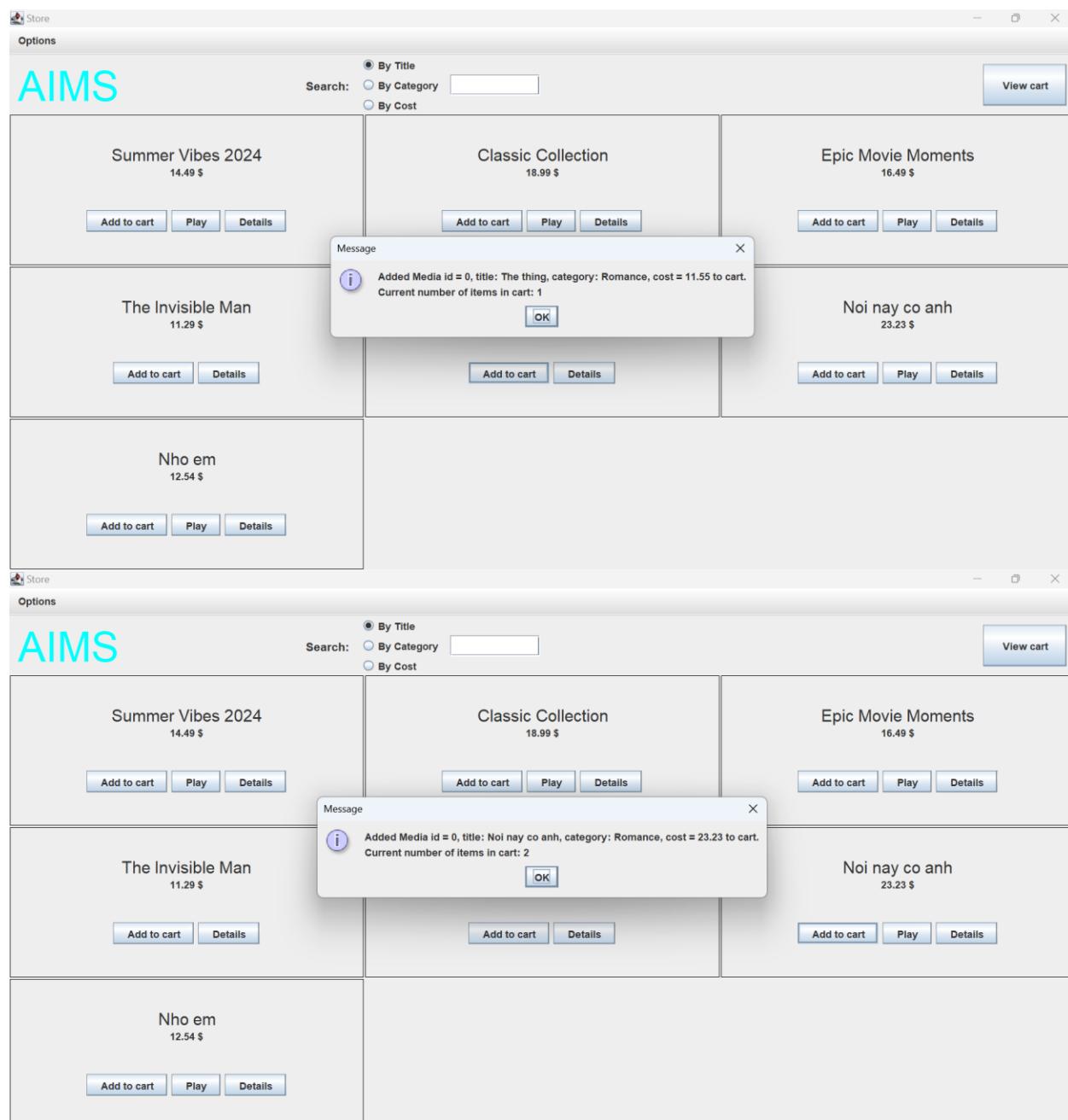


Figure 28: Store after add DVD



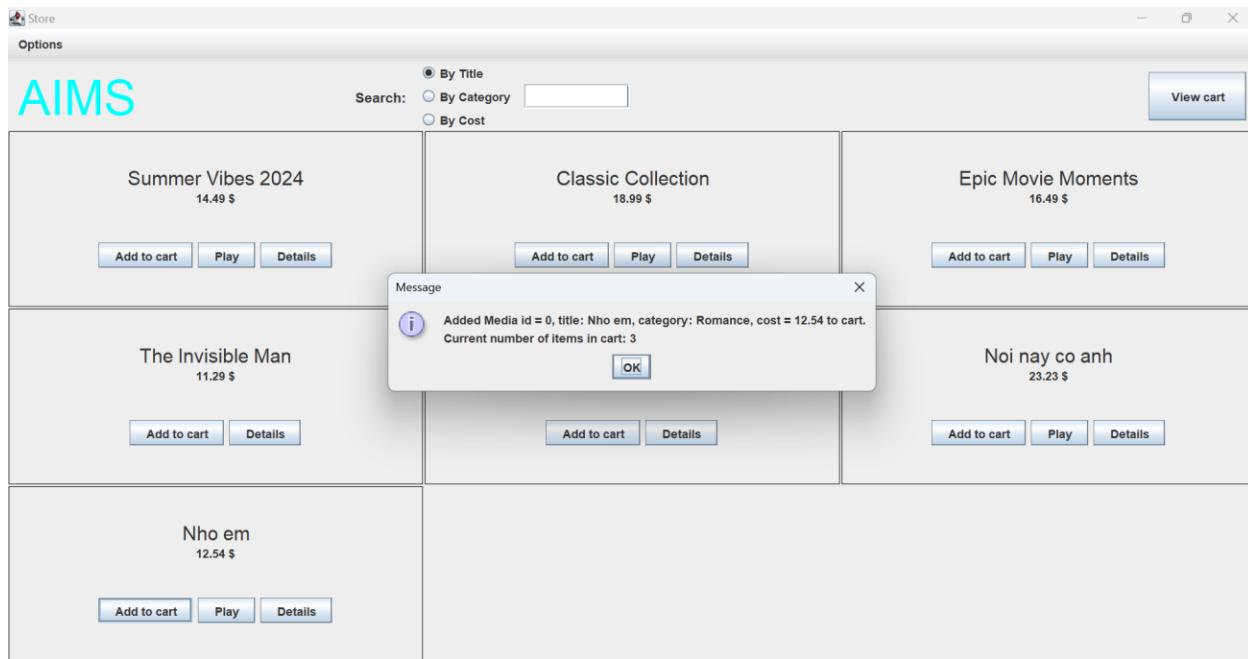


Figure 29: Add to Cart

The screenshot shows the CART application interface. At the top, there are filter options: 'Filter' (empty), 'By ID' (radio button selected), 'By Title' (radio button), and sorting options: 'Sort by Title' and 'Sort by Cost'. Below these, a table displays the items in the cart:

| Title          | Category | Cost  |
|----------------|----------|-------|
| The thing      | Romance  | 11.55 |
| Noi nay co anh | Romance  | 23.23 |
| Nho em         | Romance  | 12.54 |

Total: 47.32 \$

Buttons on the right: Place Order, Remove, Details.

Figure 30: Cart

```
@Override 3 usages
public void play() throws PlayerException {
    if (this.getLength() <= 0) {
        throw new PlayerException("Track length is invalid!");
    }
    System.out.println("Playing Track: " + this.getTitle());
    System.out.println("Track length: " + this.getLength());
}

}

💡 @Override 3 usages
public void play() throws PlayerException {
    if (getLength() <= 0) {
        throw new PlayerException("ERROR: DVD length is non-positive!");
    } else {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
}
```

Figure 31: Exception

## 9 Use case Diagram

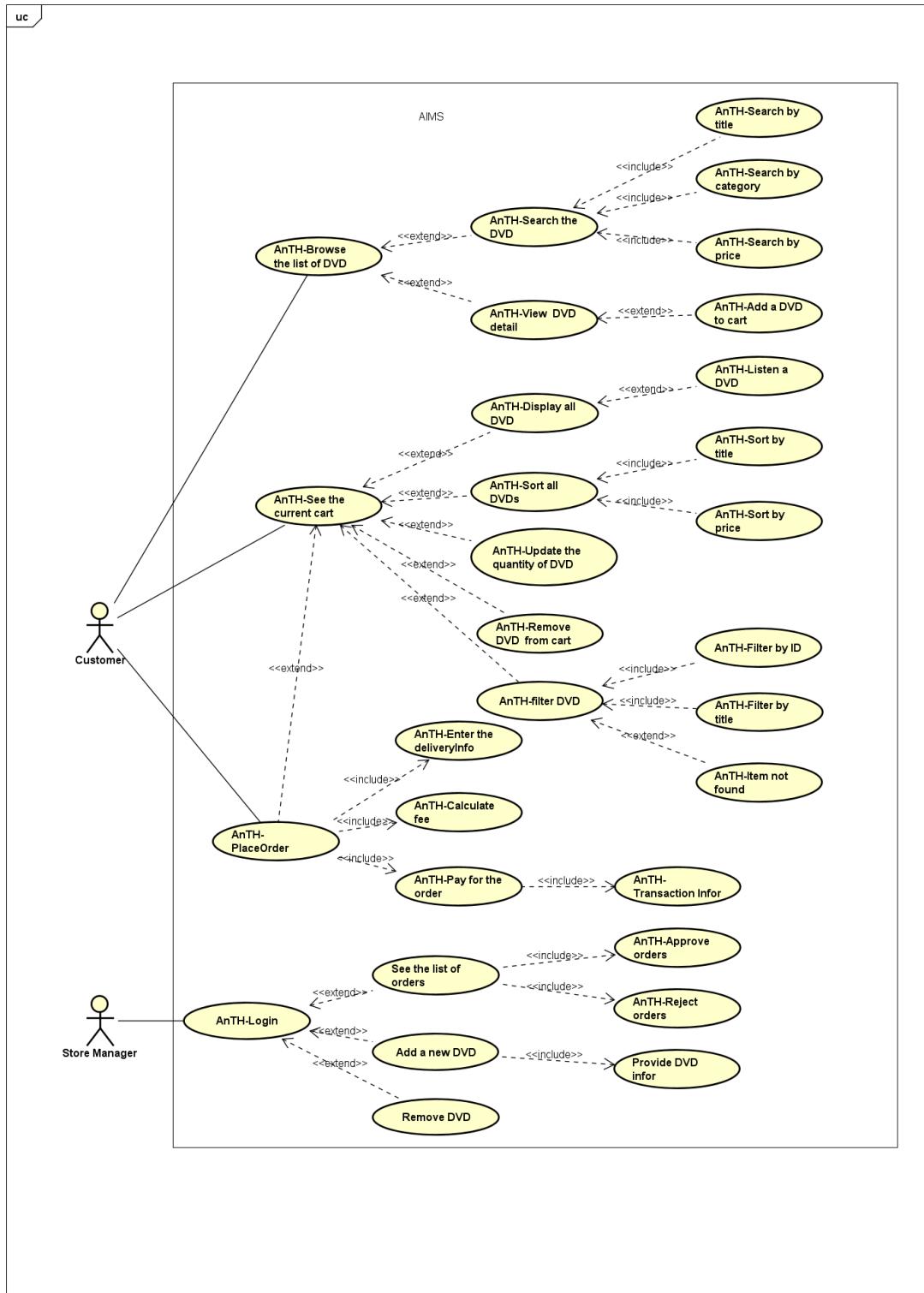


Figure 32: Use case diagram

## 10 Class Diagram

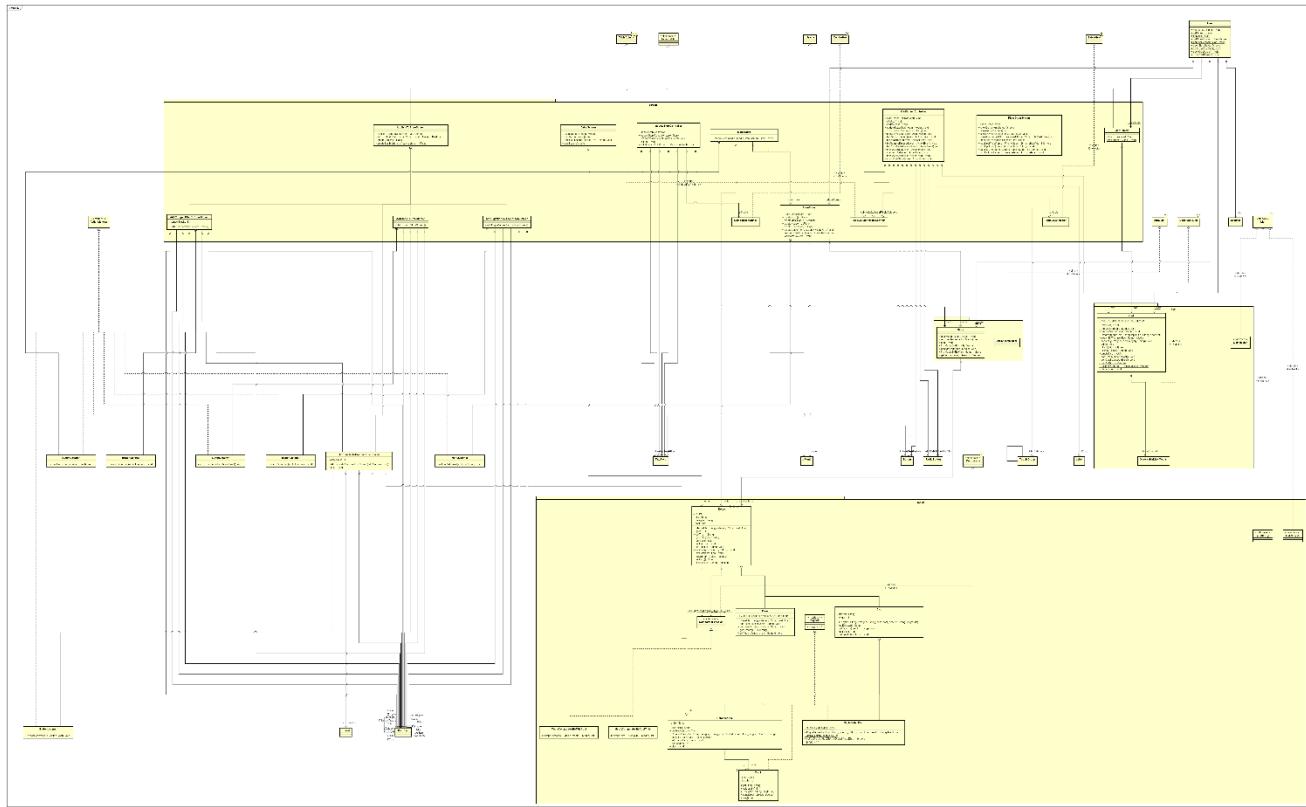


Figure 33: Class Diagram