

**BÁO CÁO THỰC HÀNH LAB 5  
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Mục lục nội dung**

1.	Swing components .....	4
1.1.	AWTAccumulator.....	4
1.2.	SwingAccumulator .....	5
1.3.	Demo.....	7
2.	Organizing Swing components with Layout Managers .....	8
2.1.	Code .....	8
2.2.	Demo.....	11
3.	Create a graphical user interface for AIMS with Swing .....	13
3.1.	Create class StoreScreen .....	13
3.2.	Create class MediaStore.....	19
3.3.	Demo.....	22
4.	JavaFX API.....	24
4.1.	Create class Painter .....	24
4.2.	Create Painter.fxml .....	25
4.3.	Create class PainterController.....	27
4.4.	Demo.....	28
5.	View Cart Screen .....	30
5.1.	Create cart.fxml.....	30
5.2.	Create class CartScreen.....	33
5.3.	Create class CartScreenController .....	35
5.4.	Demo.....	37
6.	Updating buttons based on selected item in TableView – ChangeListener .....	38
6.1.	Edit class CartScreenController .....	38
6.2.	Demo.....	41
7.	Deleting a media .....	42
7.1.	Code .....	42
7.2.	Demo.....	43
8.	Complete the Aims GUI application.....	44
9.	Use case Diagram .....	48
10.	Class Diagram.....	48

## Mục lục hình ảnh

Figure 1 Source code of AWTAccumulator .....	5
Figure 2 Demo of AWTAccumulator .....	5
Figure 3 Source code of SwingAccumulator .....	7
Figure 4 Demo of SwingAccumulator .....	7
Figure 5 Source code of NumberGrid.....	11
Figure 6 Demo buttons 0-9 .....	11
Figure 7 Demo button DEL .....	12
Figure 8 Demo button C .....	12
Figure 9 Source code of StoreScreen Class .....	18
Figure 10 Source code of MediaStore Class .....	22
Figure 11 Demo of Store Screen.....	22
Figure 12 Demo Add to cart button .....	22
Figure 13 Demo Play button .....	23
Figure 14 Demo View cart button.....	23
Figure 15 Source code of Painter Class .....	24
Figure 16 Source code of Painter.fxml .....	26
Figure 17 Source code of PainterController Class.....	28
Figure 18 Demo Pen .....	28
Figure 19 Demo Eraser .....	29
Figure 20 Demo Clear button .....	29
Figure 21 Source code of cart.fxml.....	32
Figure 22 Source code of CartScreen Class .....	34
Figure 23 Source code of CartScreenController Class .....	36
Figure 24 Demo View Cart Screen .....	37
Figure 25 Source code of CartScreenController Class .....	40
Figure 26 Demo media unplayable .....	41
Figure 27 Source code of btnRemovePressed Method .....	42
Figure 28 Demo button remove .....	43
Figure 29 Demo Store before add book .....	44
Figure 30 Demo add book.....	44
Figure 31 Demo Store after add book .....	45
Figure 32 Demo add CD .....	45
Figure 33 Demo Store after add CD .....	46
Figure 34 Demo Store add DVD .....	46
Figure 35 Demo Store after add DVD .....	47
Figure 36 Demo Cart .....	47
Figure 37 Catch Exception.....	47
Figure 38 Use case Diagram .....	48
Figure 39 Class Diagram .....	48

20215275

Trần Minh Huyền

Kỳ 20231

Thực hành lập trình hướng đối tượng

## 1. Swing components

### 1.1. AWTAccumulator

```
1 package lab05.GUIProject.swing;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5
6 ▷ public class AWTAccumulator extends Frame {
7     // Attribute
8     5 usages
9     private TextField tfInput;
10    4 usages
11    private TextField tfOutput;
12    2 usages
13    private int sum = 0;
14
15    // Constructor
16    1 usage
17    public AWTAccumulator() {
18        setLayout(new GridLayout( 2, 2));
19
20            add(new Label( text: "Enter an Integer: "));

```

```

22         add(new Label( text: "The Accumulated Sum is: "));
23
24         tfOutput = new TextField( columns: 10);
25         tfOutput.setEditable(false);
26         add(tfOutput);
27
28         setTitle("AWT Accumulator");
29         setSize( width: 350, height: 120);
30         setVisible(true);
31     }
32
33 >     public static void main(String[] args) {
34         new AWTAccumulator();
35     }
36
37     private class TFIInputListener implements ActionListener {
38         @Override
39         public void actionPerformed(ActionEvent evt) {
40             int numberIn = Integer.parseInt (tfInput.getText ());
41             sum += numberIn;
42             tfInput.setText("");
43             tfOutput.setText(sum + "");
44         }
45     }

```

Figure 1 Source code of AWTAccumulator

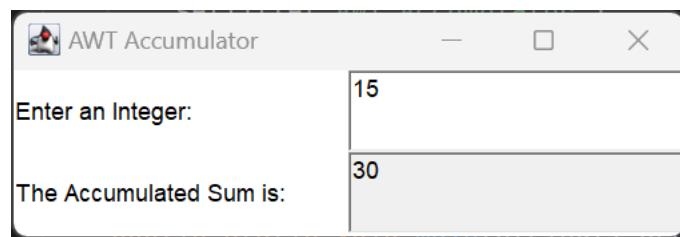


Figure 2 Demo of AWTAccumulator

## 1.2. SwingAccumulator

```
1 package lab05.GUIProject.swing;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6
7 ▷ public class SwingAccumulator extends JFrame {
8     // Attribute
9     5 usages
10    private JTextField tfInput;
11    4 usages
12    private JTextField tfOutput;
13    2 usages
14    private int sum = 0;
15
16    // Constructor
17    1 usage
18    public SwingAccumulator() {
19        Container cp = getContentPane();
20        cp.setLayout(new GridLayout( 2, 2));
21
22        cp.add(new JLabel( text: "Enter an Integer: "));
23
24        tfInput = new JTextField( columns: 10);
25        cp.add(tfInput);
```

```

22         tfInput.addActionListener(new TFInputListener());
23
24         cp.add(new JLabel( text: "The Accumulated Sum is: "));
25
26         tfOutput = new JTextField( columns: 10);
27         tfOutput.setEditable(false) ;
28         cp.add(tfOutput);
29
30         setTitle("Swing Accumulator");
31         setSize( width: 350, height: 120);
32         setVisible(true);
33     }
34
35 ▶ public static void main(String[] args) {
36     new SwingAccumulator();
37 }
38 1 usage
39 private class TFInputListener implements ActionListener {
40     @Override
41     public void actionPerformed(ActionEvent evt) {
42         int numberIn = Integer.parseInt(tfInput.getText ());
43         sum += numberIn;
44         tfInput.setText("");
45         tfOutput.setText(sum + "");
46     }
47 }

```

Figure 3 Source code of SwingAccumulator

### 1.3. Demo

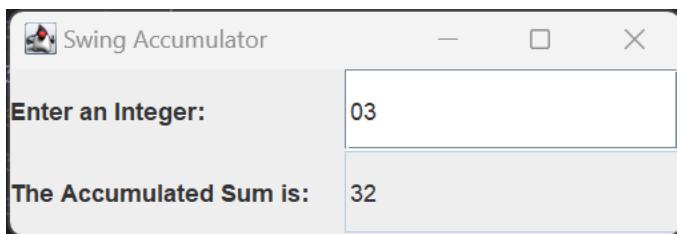


Figure 4 Demo of SwingAccumulator

## 2. Organizing Swing components with Layout Managers

### 2.1. Code

```
1 package lab05.GUIProject.swing;
2     import javax.swing.*;
3     import java.awt.*;
4     import java.awt.event.ActionEvent;
5     import java.awt.event.ActionListener;
6
7 ▶ public class NumberGrid extends JFrame {
8     // Attribute
9     6 usages
10    private JButton[] btnNumbers = new JButton[10];
11    3 usages
12    private JButton btnDelete, btnReset;
13    8 usages
14    private JTextField tfDisplay;
15
16    // Constructor
17    1 usage
18    public NumberGrid() {
19        tfDisplay = new JTextField();
20        tfDisplay.setComponentOrientation(  
            ComponentOrientation.RIGHT_TO_LEFT);  

21
22        JPanel panelButtons = new JPanel(new GridLayout( rows: 4, cols: 3));
23        addButtons(panelButtons);
```

```
22     Container cp = getContentPane();
23     cp.setLayout(new BorderLayout());
24     cp.add(tfDisplay, BorderLayout.NORTH);
25     cp.add(panelButtons, BorderLayout.CENTER);
26
27     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28     setTitle("Number Grid");
29     setSize( width: 200, height: 200);
30     setVisible(true);
31 }
32
33 // Method to add buttons
34 void addButtons(JPanel panelButtons) {
35     ButtonListener btnListener = new ButtonListener();
36     for (int i = 1; i <= 9; i++) {
37         btnNumbers[i] = new JButton( text: "" + i);
38         panelButtons.add(btnNumbers[i]);
39         btnNumbers[i].addActionListener(btnListener);
40     }
41
42     btnDelete = new JButton( text: "DEL");
43     panelButtons.add(btnDelete);
44     btnDelete.addActionListener(btnListener);
45 }
```

```

47     btnNumbers[0] = new JButton( text: "0");
48     panelButtons.add(btnNumbers[0]);
49     btnNumbers[0].addActionListener(btnListener);
50
51     btnReset = new JButton( text: "C");
52     panelButtons.add(btnReset);
53     btnReset.addActionListener(btnListener);
54 }
55
56     2 usages
57
58     private class ButtonListener implements ActionListener {
59         @Override
60         public void actionPerformed(ActionEvent e) {
61             String button = e.getActionCommand();
62             if (button.charAt(0) >= '0' && button.charAt(0) <= '9') {
63                 tfDisplay.setText(tfDisplay.getText() + button);
64             } else if (button.equals("DEL")) {
65                 String currentText = tfDisplay.getText();
66                 String text = currentText.substring(0, currentText.length() - 1);
67                 tfDisplay.setText(text);
68             } else {
69                 tfDisplay.setText("");
70             }
71     }
72     public static void main(String[] args) {
73         new NumberGrid();
74     }
75 }
```

Figure 5 Source code of NumberGrid

## 2.2. Demo

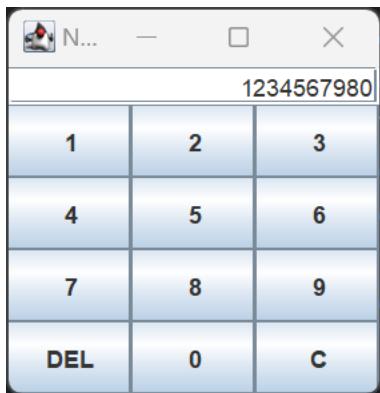


Figure 6 Demo buttons 0-9

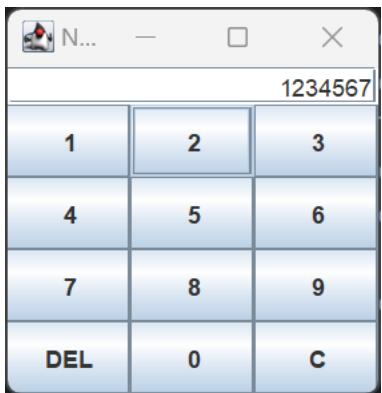


Figure 7 Demo button DEL

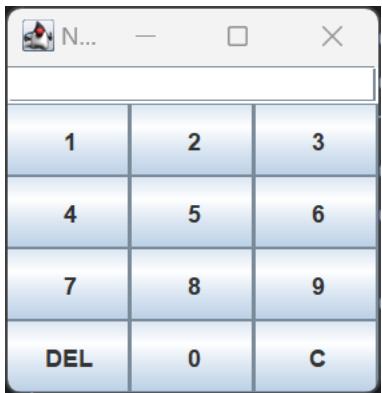


Figure 8 Demo button C

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

#### 3.1. Create class StoreScreen

```
1  package lab05.AimsProject.screen;
2  import javax.swing.*;
3  import java.awt.*;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.ActionListener;
6  import java.util.List;
7
8  import lab05.AimsProject.Cart;
9  import lab05.AimsProject.Store;
10 import lab05.AimsProject.media.*;
11
12 public class StoreScreen extends JFrame {
13     // Attribute
14     2 usages
15     private Store store;
16     3 usages
17     private Cart cart;
18
19     // Constructor
20     1 usage
21     public StoreScreen(Store store, Cart cart) {
22         this.store = store;
23         this.cart = cart;
24         Container cp = getContentPane();
25         cp.setLayout(new BorderLayout());
```

```
24         cp.add(createNorth(), BorderLayout. NORTH);
25         cp.add(createCenter(), BorderLayout. CENTER);
26
27         setVisible(true);
28         setTitle("Store");
29         setSize( width: 1024, height: 768);
30     }
31
32     // Method to create NORTH component
33     1 usage
34     JPanel createNorth() {
35         JPanel north = new JPanel();
36         north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
37         north.add(createMenuBar());
38         north.add(createHeader());
39         return north;
40     }
41
42     // Method to create Menu Bar
43     1 usage
44     JMenuBar createMenuBar() {
45         JMenu menu = new JMenu( s: "Options");
46
47         JMenu smUpdateStore = new JMenu( s: "Update Store");
48         smUpdateStore.add(new JMenuItem( text: "Add Book"));
```

```
47     smUpdateStore.add(new JMenuItem( text: "Add CD"));
48     smUpdateStore.add(new JMenuItem( text: "Add DVD"));
49
50     menu.add(smUpdateStore);
51     menu.add(new JMenuItem( text: "View store"));
52     menu.add(new JMenuItem( text: "View cart"));
53
54     JMenuBar menuBar = new JMenuBar();
55     menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
56     menuBar.add(menu);
57
58     return menuBar;
59 }
60
61 // Method to create Header
62 JPanel createHeader() {
63     JPanel header = new JPanel();
64     header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));
65
66     JLabel title = new JLabel( text: "AIMS");
67     title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size: 50));
68     title.setForeground(Color.CYAN);
69
70     ButtonListener buttonListener = new ButtonListener(cart);
```

```

71         JButton cart = new JButton("View cart");
72         cart.setPreferredSize(new Dimension(width: 100, height: 50));
73         cart.setMaximumSize(new Dimension(width: 100, height: 50));
74         cart.addActionListener(buttonListener);
75
76         header.add(Box.createRigidArea(new Dimension(width: 10, height: 10)));
77         header.add(title);
78         header.add(Box.createHorizontalGlue());
79         header.add(cart);
80         header.add(Box.createRigidArea(new Dimension(width: 10, height: 10)));
81
82         return header;
83     }
84
85     // Method to create Center
86     JPanel createCenter() {
87         JPanel center = new JPanel();
88         center.setLayout(new GridLayout(rows: 3, cols: 3, hgap: 2, vgap: 2));
89
90         List<Media> mediaInStore = store.getItemsInStore();
91         MediaStore.setCart(cart);
92         for (int i = 0; i < 9; i++) {
93             MediaStore cell = new MediaStore(mediaInStore.get(i));
94             center.add(cell);

```

```

95         }
96
97         return center;
98     }
99
100    public static void main(String[] args) {
101        Store store = new Store();
102        Cart cart = new Cart();
103        // Create new media and add them to the store
104        // Adding DVDs
105        Media dvd1 = new DigitalVideoDisc(id: 1, title: "Inception",
106                                         category: "Science Fiction", director: "Christopher Nolan", length: 148, cost: 19.99f);
107        store.addMedia(dvd1);
108
109        Media dvd2 = new DigitalVideoDisc(id: 2, title: "The Dark Knight",
110                                         category: "Action", director: "Christopher Nolan", length: 152, cost: 17.99f);
111        store.addMedia(dvd2);
112
113        Media dvd3 = new DigitalVideoDisc(id: 7, title: "Interstellar",
114                                         category: "Science Fiction", director: "Christopher Nolan", length: 169, cost: 21.99f);
115        store.addMedia(dvd3);
116
117        // Adding CDs
118        Media cd1 = new CompactDisc(id: 3, title: "Random Access Memories",
119                                   category: "Electronic", cost: 15.99f, director: "Daft Punk", length: 13, artist: "Daft Punk");

```

```
120     store.addMedia(cd1);
121
122     Media cd2 = new CompactDisc( id: 4, title: "25",
123         category: "Pop", cost: 14.99f, director: "Adele", length: 11, artist: "Adele");
124     store.addMedia(cd2);
125
126     Media cd3 = new CompactDisc( id: 8, title: "Lover",
127         category: "Pop", cost: 17.99f, director: "Taylor Swift", length: 18, artist: "Taylor Swift");
128     store.addMedia(cd3);
129
130     // Adding Books
131     Media book1 = new Book( id: 5, title: "The Silent Patient",
132         category: "Thriller", cost: 14.95f);
133     store.addMedia(book1);
134
135     Media book2 = new Book( id: 6, title: "Where the Crawdads Sing",
136         category: "Mystery", cost: 12.99f);
137     store.addMedia(book2);
138
139     Media book3 = new Book( id: 9, title: "Educated",
140         category: "Memoir", cost: 16.95f);
141     store.addMedia(book3);
142     new StoreScreen(store, cart);
143 }
```

```
145      // Button Listener for View Cart
146      2 usages
147      private class ButtonListener implements ActionListener {
148          // Attribute
149          2 usages
150          private Cart cart;
151          // Constructor
152          1 usage
153          public ButtonListener(Cart cart) {
154              super();
155              this.cart = cart;
156          }
157          @Override
158          public void actionPerformed(ActionEvent e) {
159              String button = e.getActionCommand();
160              if (button.equals("View cart")) {
161                  cart.printCart();
162              }
163          }
164      }
```

Figure 9 Source code of StoreScreen Class

### 3.2. Create class MediaStore

```
1  package lab05.AimsProject.screen;
2  ✓ import javax.swing.*;
3  import java.awt.*;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.ActionListener;
6  import java.util.List;
7
8  import lab05.AimsProject.Cart;
9  import lab05.AimsProject.media.*;
10
11 4 usages
12
13 public class MediaStore extends JPanel {
14     // Attribute
15     ⚡ 1 usage
16     private Media media;
17     3 usages
18     private static Cart cart;
19
20     // Constructor
21     1 usage
22     @ ✓ public MediaStore(Media media) {
23         this.media = media;
24         this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
25
26         JLabel title = new JLabel(media.getTitle());
```

```
22     title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size: 20));
23     title.setAlignmentX(CENTER_ALIGNMENT);
24
25     JLabel cost = new JLabel(text: "" + media.getCost() + " $");
26     cost.setAlignmentX(CENTER_ALIGNMENT);
27
28     JPanel container = new JPanel();
29     container.setLayout(new FlowLayout(FlowLayout.CENTER));
30
31     ButtonListener buttonListener = new ButtonListener(media);
32     JButton addButton = new JButton(text: "Add to cart");
33     container.add(addButton);
34     addButton.addActionListener(buttonListener);
35
36     if (media instanceof Playable) {
37         JButton playButton = new JButton(text: "Play");
38         container.add(playButton);
39         playButton.addActionListener(buttonListener);
40     }
41
42     this.add(Box.createVerticalGlue());
43     this.add(title);
44     this.add(cost);
45     this.add(Box.createVerticalGlue());
46     this.add(container);
```

```
48         this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
49     }
50
51     // Getter and Setter
52
53
54     public static Cart getCart() {
55         return cart;
56     }
57
58     public static void setCart(Cart cart) {
59         MediaStore.cart = cart;
60     }
61
62     // Button Listener for Add to Cart and Play
63     2 usages
64     private class ButtonListener implements ActionListener {
65         // Attribute
66         3 usages
67         private Media media;
68
69         // Constructor
70         1 usage
71         public ButtonListener(Media media) {
72             super();
73         }
74     }
75 }
```

```

70             this.media = media;
71         }
72
73     @Override
74     public void actionPerformed(ActionEvent e) {
75         String button = e.getActionCommand();
76         if (button.equals("Add to cart")) {
77             cart.addMedia(media);
78         } else if (media instanceof Disc disc) {
79             disc.play();
80         }
81     }
82 }
83
84

```

Figure 10 Source code of MediaStore Class

### 3.3. Demo

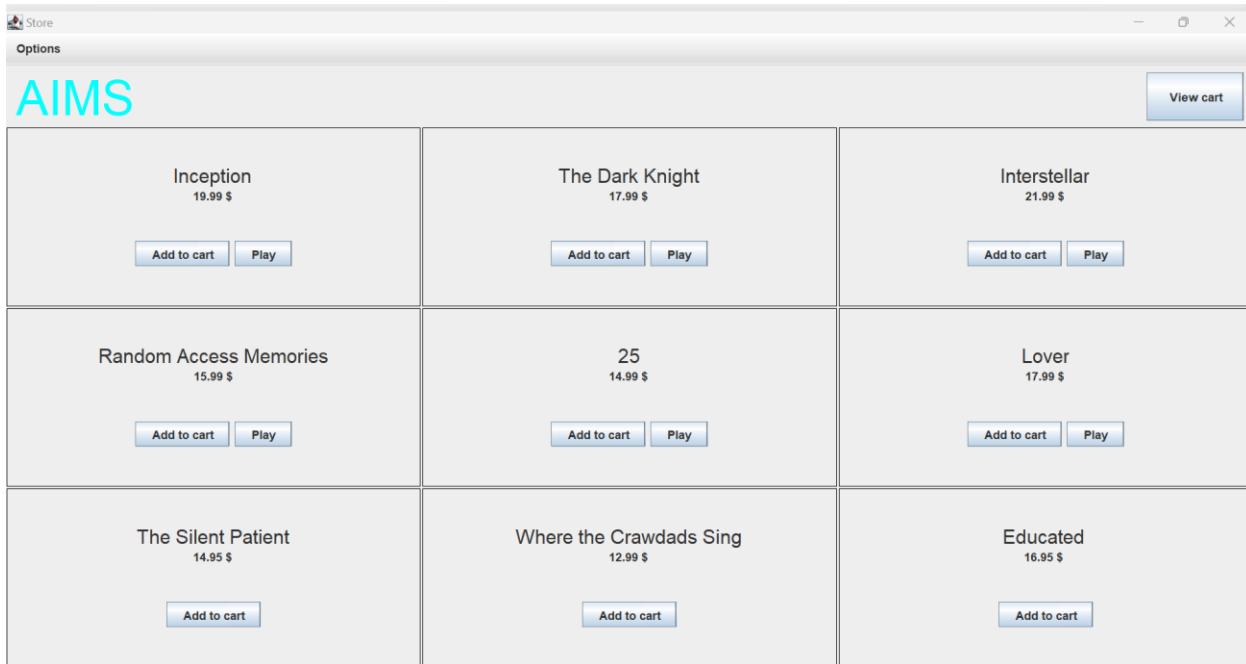


Figure 11 Demo of Store Screen

```

Added to cart successful
Added to cart successful
Added to cart successful

```

Figure 12 Demo Add to cart button

```
Playing DVD: Inception
DVD length: 148
Playing CD: 25
CD artist: Adele
CD length: 0
Playing DVD: Interstellar
DVD length: 169
```

Figure 13 Demo Play button

```
*****CART*****
Ordered Items:
1. DVD - Inception - Science Fiction - Christopher Nolan - 148: 19.99$
2. DVD - The Dark Knight - Action - Christopher Nolan - 152: 17.99$
7. DVD - Interstellar - Science Fiction - Christopher Nolan - 169: 21.99$
Total cost: 59.97
*****
```

Figure 14 Demo View cart button

## 4. JavaFX API

### 4.1. Create class Painter

```
1 package lab05.GUIProject.javafx;
2
3     import javafx.application.Application;
4     import javafx.fxml.FXMLLoader;
5     import javafx.scene.Parent;
6     import javafx.scene.Scene;
7     import javafx.stage.Stage;
8
9     public class Painter extends Application {
10         @Override
11         public void start(Stage stage) throws Exception {
12             Parent root = FXMLLoader.load(getClass()
13                 .getResource( name: "Painter.fxml"));
14             Scene scene = new Scene(root);
15             stage.setTitle("Painter");
16             stage.setScene(scene);
17             stage.show();
18         }
19
20     public static void main (String []args){
21         launch(args);
22     }
23 }
```

Figure 15 Source code of Painter Class

#### 4.2. Create Painter.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.RadioButton?>
6  <?import javafx.scene.control.TitledPane?>
7  <?import javafx.scene.control.ToggleGroup?>
8  <?import javafx.scene.layout.AnchorPane?>
9  <?import javafx.scene.layout.BorderPane?>
10 <?import javafx.scene.layout.Pane?>
11 <?import javafx.scene.layout.VBox?>
12
13
14 <AnchorPane prefHeight="400.0" prefWidth="600.0"
15     xmlns="http://javafx.com/javafx/21"
16     xmlns:fx="http://javafx.com/fxml/1"
17     fx:controller="lab05.GUIProject.javafx.PainterController">
18     <children>
19         <BorderPane layoutX="251.0" layoutY="4.0" maxHeight="-Infinity"
20             maxWidth="-Infinity" minHeight="-Infinity"
21             minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0">
22             <padding>
23                 <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
24
25                 <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
26             </padding>
27             <left>
28                 <VBox maxHeight="1.7976931348623157E308"
29                     maxWidth="-Infinity" prefWidth="70.0"
30                     spacing="8.0" BorderPane.alignment="CENTER">
31                     <BorderPane.margin>
32                         <Insets right="8.0" />
33                     </BorderPane.margin>
34                     <children>
35                         <TitledPane animated="false" text="Tools">
36                             <content>
37                                 <AnchorPane>
38                                     <children>
39                                         <RadioButton mnemonicParsing="false"
40                                             onAction="#penButtonPressed" selected="true" text="Pen">
41                                             <toggleGroup>
42                                                 <ToggleGroup fx:id="identical" />
43                                             <padding>
44                                                 <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
45                                             </padding>
46                                         </RadioButton>
47                                     </children>
48                                 </AnchorPane>
49                             </content>
50                         </TitledPane>
51                     </children>
52                 </VBox>
53             </left>
54         </BorderPane>
55     </children>
56 
```

```
46      <RadioButton layoutY="31.0" mnemonicParsing="false"
47          onAction="#eraserButtonPressed" selected="true"
48          text="Eraser" toggleGroup="$identical">
49          <padding>
50              <Insets left="8.0" right="8.0" top="8.0" />
51          </padding>
52      </RadioButton>
53  </children>
54  <padding>
55      <Insets bottom="4.0" left="4.0" right="4.0" top="4.0" />
56  </padding>
57  </AnchorPane>
58  </content>
59  </TitledPane>
60  <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false"
61      onAction="#clearButtonPressed" text="Clear">
62      <VBox.margin>
63          <Insets />
64      </VBox.margin>
65      <padding>
66          <Insets bottom="4.0" left="4.0" right="4.0" top="4.0" />
67      </padding>
68  </Button>
69      </children>
70  </VBox>
71  </left>
72  <center>
73      <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged"
74          style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
75  </center>
76  </BorderPane>
77  </children>
78  </AnchorPane>
```

Figure 16 Source code of Painter.fxml

#### 4.3. Create class PainterController

```
1 package lab05.GUIProject.javafx;
2
3 import javafx.event.ActionEvent;
4 import javafx.fxml.FXML;
5 import javafx.scene.control.ToggleGroup;
6 import javafx.scene.input.MouseEvent;
7 import javafx.scene.layout.Pane;
8 import javafx.scene.paint.Color;
9 import javafx.scene.paint.Paint;
10 import javafx.scene.shape.Circle;
11
12 1 usage
13 public class PainterController {
14     3 usages
15     private Paint penColor;
16     @FXML
17     private Pane drawingAreaPane;
18     @FXML
19     private ToggleGroup identical;
20     @FXML
21     void clearButtonPressed(ActionEvent event) {
22         drawingAreaPane.getChildren().clear();
23     }
```

```
25     @FXML
26     void drawingAreaMouseDragged(MouseEvent event) {
27         Circle newCircle = new Circle(event.getX(),
28             event.getY(), 2, penColor);
29         drawingAreaPane.getChildren().add(newCircle);
30     }
31
32     @FXML
33     void eraserButtonPressed(ActionEvent event) {
34         penColor = Color.WHITE;
35     }
36
37     @FXML
38     void penButtonPressed(ActionEvent event) {
39         penColor = Color.BLACK;
40     }
41 }
```

Figure 17 Source code of PainterController Class

#### 4.4. Demo



Figure 18 Demo Pen



Figure 19 Demo Eraser

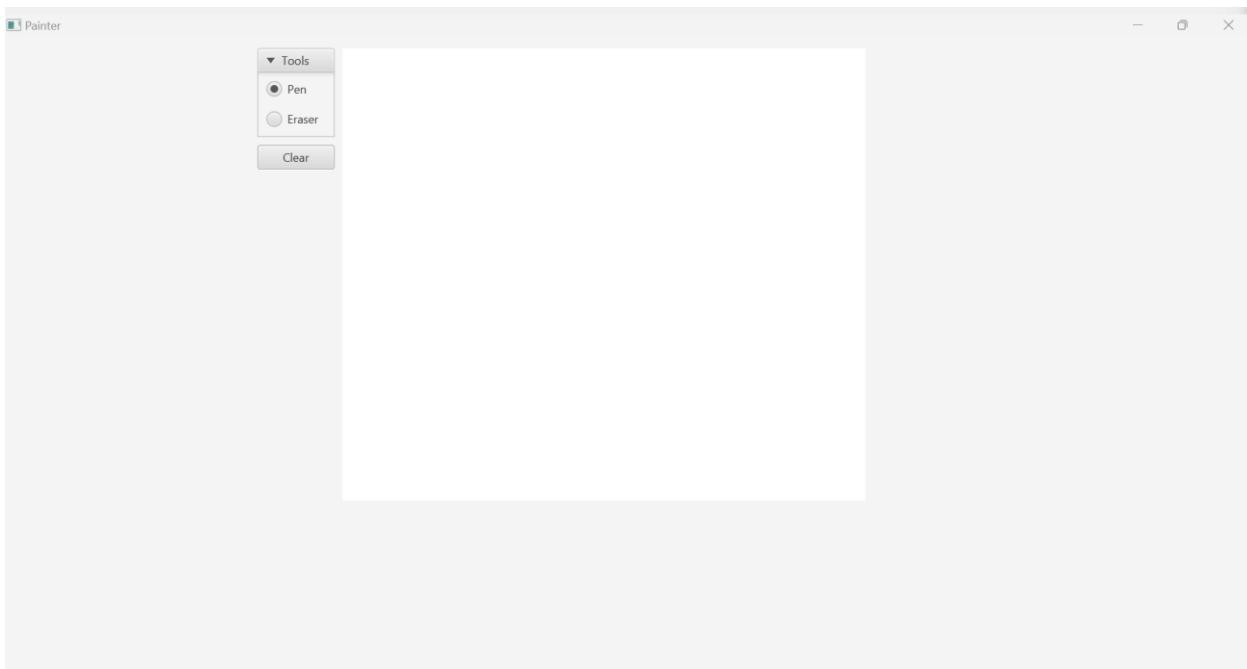


Figure 20 Demo Clear button

## 5. View Cart Screen

### 5.1. Create cart.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.ButtonBar?>
6  <?import javafx.scene.control.Label?>
7  <?import javafx.scene.control.Menu?>
8  <?import javafx.scene.controlMenuBar?>
9  <?import javafx.scene.control.MenuItem?>
10 <?import javafx.scene.control.RadioButton?>
11 <?import javafx.scene.control.TableColumn?>
12 <?import javafx.scene.control TableView?>
13 <?import javafx.scene.control.TextField?>
14 <?import javafx.scene.control.ToggleGroup?>
15 <?import javafx.scene.layout.AnchorPane?>
16 <?import javafx.scene.layout.BorderPane?>
17 <?import javafx.scene.layout.HBox?>
18 <?import javafx.scene.layout VBox?>
19 <?import javafx.scene.text.Font?>
20
21 <AnchorPane prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/21"
22   xmlns:fx="http://javafx.com/fxml/1">
23   |   <children>

```

```

24   |       <BorderPane layoutX="79.0" layoutY="14.0" prefHeight="768.0" prefWidth="1024.0">
25   |           <top>
26   |               <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
27   |                   <children>
28   |                       <MenuBar>
29   |                           <menus>
30   |                               <Menu mnemonicParsing="false" text="Options">
31   |                                   <items>
32   |                                       <Menu mnemonicParsing="false" text="Update Store">
33   |                                           <items>
34   |                                               <MenuItem mnemonicParsing="false" text="Add Book" />
35   |                                               <MenuItem mnemonicParsing="false" text="Add CD" />
36   |                                               <MenuItem mnemonicParsing="false" text="Add DVD" />
37   |                                       </items>
38   |                                   </Menu>
39   |                               <MenuItem mnemonicParsing="false" text="View Store" />
40   |                               <MenuItem mnemonicParsing="false" text="View Cart" />
41   |                           </items>
42   |                       </MenuBar>
43   |                   <Label text="CART" textFill="AQUA">
44   |                       <font>
45   |                           </font>
46   |                   </Label>

```

```

47           <Font size="50.0" />
48       </font>
49   <VBox.margin>
50     <Insets />
51   </VBox.margin>
52   <padding>
53     <Insets left="10.0" />
54   </padding>
55   <Label>
56     <children>
57   </VBox>
58 </top>
59 <center>
60   <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
61     <padding>
62       <Insets left="10.0" />
63     </padding>
64     <children>
65       <HBox alignment="CENTER_LEFT" prefWidth="200.0" spacing="10.0">
66         <padding>
67           <Insets bottom="10.0" top="10.0" />
68         </padding>
69         <children>
70           <Label text="Filter" />
71           <TextField />
72           <RadioButton mnemonicParsing="false" selected="true" text="By ID">
73             <toggleGroup>
74               <ToggleGroup fx:id="filterCategory" />
75             </toggleGroup>
76           </RadioButton>
77           <RadioButton mnemonicParsing="false" text="By Title"
78                         toggleGroup="$filterCategory" />
79         </children>
80       </HBox>
81       <TableView fx:id="tblMedia">
82         <columns>
83           <TableColumn fx:id="colMediaTitle" prefWidth="75.0" text="Title" />
84           <TableColumn fx:id="colMediaCategory" prefWidth="75.0" text="Category" />
85           <TableColumn fx:id="colMediaCost" prefWidth="75.0" text="Cost" />
86         </columns>
87         <columnResizePolicy>
88           <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
89         </columnResizePolicy>
90       </TableView>
91       <ButtonBar prefHeight="40.0" prefWidth="200.0">
92         <buttons>

```

```

93           <Button mnemonicParsing="false" text="Play" />
94           <Button mnemonicParsing="false" text="Remove" />
95       
```

```

96       </children>
97   
```

```

98   </VBox>
99   
```

```

100  </center>
101  <right>
102  
```

```

103      <VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
104          <padding>
105              <Insets top="50.0" />
106          </padding>
107          <children>
108              <HBox alignment="CENTER">
109                  <children>
110                      <Label lineSpacing="10.0" text="Total:">
111                          <font>
112                              <Font size="24.0" />
113                      </font>
114                      <Label text="0 $" textFill="AQUA">
115                          <font>
116                              <Font size="24.0" />
117                      </font>
118                  </children>
119              
```

```

120      <Button mnemonicParsing="false" style="-fx-background-color: red;" 
121          text="Place Order" textFill="WHITE">
122          <font>
123              <Font size="24.0" />
124          </font>
125      
```

```

126      </children>
127  
```

```

128  </VBox>
129  
```

```

130  </right>
131  
```

```

132  </BorderPane>
133  
```

```

134  </children>
135  
```

Figure 21 Source code of cart.fxml

### 5.2. Create class CartScreen

```
1 package lab05.AimsProject.screen;
2
3 import javafx.application.Platform;
4 import javafx.embed.swing.JFXPanel;
5 import javafx.fxml.FXMLLoader;
6 import javafx.scene.Parent;
7 import javafx.scene.Scene;
8 import lab05.AimsProject.Cart;
9
10 import javax.swing.*;
11 import java.io.IOException;
12 import java.net.URL;
13
14 1 usage
15 public class CartScreen extends JFrame {
16     1 usage
17     private Cart cart;
18
19     1 usage
20     public CartScreen(Cart cart) {
21         super();
22
23         this.cart = cart;
24
25         JFXPanel fxPanel = new JFXPanel();
```

```
23     this.add(fxPanel);
24
25     this.setTitle("Cart");
26     this.setVisible(true);
27     Platform.runLater(new Runnable() {
28         @Override
29         public void run() {
30             try {
31                 FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "cart.fxml"));
32                 CartScreenController controller =
33                     new CartScreenController(cart);
34                 loader.setController(controller);
35                 Parent root = loader.load();
36                 fxPanel.setScene(new Scene(root));
37             } catch (IOException e) {
38                 e.printStackTrace();
39             }
40         }
41     });
42 }
43 }
```

Figure 22 Source code of CartScreen Class

### 5.3. Create class CartScreenController

```
1 package lab05.AimsProject.screen;
2 import javafx.fxml.FXML;
3 import javafx.fxml.FXMLLoader;
4 import javafx.scene.control.TableColumn;
5 import javafx.scene.control.TableView;
6 import lab05.AimsProject.Cart;
7 import lab05.AimsProject.media.Media;
8 import javafx.scene.control.cell.PropertyValueFactory;
9
10    2 usages
10 public class CartScreenController {
11      2 usages
11     private Cart cart;
12
12         no usages
13     private FXMLLoader loader;
14
14         1 usage
15     @FXML
16     private TableView<Media> tblMedia;
17
17         1 usage
18     @FXML
18     private TableColumn<Media, String> colMediaTitle;
```

```
21      @FXML
22      private TableColumn<Media, String> colMediaCategory;
23
24      1 usage
25      @FXML
26      private TableColumn<Media, Float> colMediaCost;
27
28      1 usage
29      public CartScreenController(Cart cart) {
30          super();
31          this.cart = cart;
32          //        loader.load();
33      }
34
35      no usages
36      @FXML
37      private void initialize() {
38          colMediaTitle.setCellValueFactory(
39              new PropertyValueFactory<Media, String>( s: "title"));
40          colMediaCategory.setCellValueFactory(
41              new PropertyValueFactory<Media, String>( s: "category"));
42          colMediaCost.setCellValueFactory(
43              new PropertyValueFactory<Media, Float>( s: "cost"));
44          tblMedia.setItems(this.cart.getItemsOrdered());
45      }
46  }
```

Figure 23 Source code of CartScreenController Class

#### 5.4. Demo

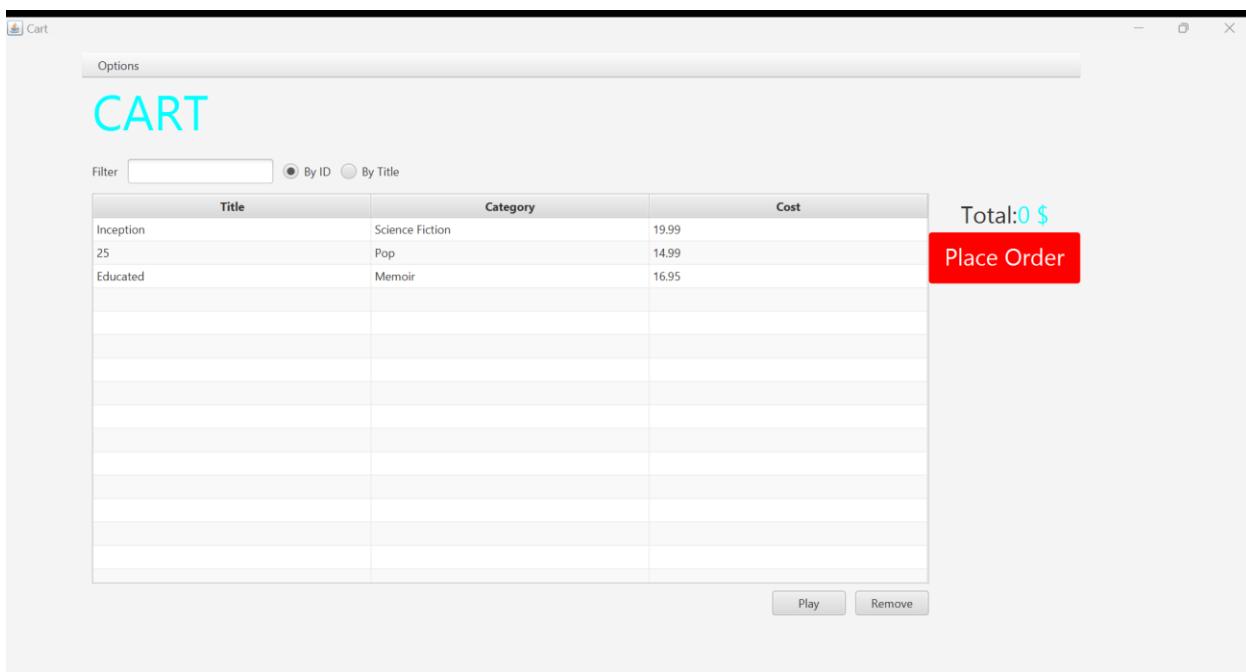


Figure 24 Demo View Cart Screen

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

### 6.1. Edit class CartScreenController

```
1  package lab05.AimsProject.screen;
2  import javafx.beans.value.ChangeListener;
3  import javafx.beans.value.ObservableValue;
4  import javafx.fxml.FXML;
5  import javafx.fxml.FXMLLoader;
6  import javafx.scene.control.Button;
7  import javafx.scene.control.TableColumn;
8  import javafx.scene.control.TableView;
9  import lab05.AimsProject.Cart;
10 import lab05.AimsProject.media.Media;
11 import javafx.scene.control.cell.PropertyValueFactory;
12 import lab05.AimsProject.media.Playable;
13
14 2 usages
15 public class CartScreenController {
16     2 usages
17     private Cart cart;
18
19     no usages
20     private FXMLLoader loader;
21
22     2 usages
23     @FXML
24     private TableView<Media> tblMedia;
```

```
23     @FXML  
24     private TableColumn<Media, String> colMediaTitle;  
25  
26     1 usage  
27     @FXML  
28     private TableColumn<Media, String> colMediaCategory;  
29  
30     1 usage  
31     @FXML  
32     private Button btnPlay;  
33  
34     2 usages  
35     @FXML  
36     private Button btnRemove;  
37     1 usage  
38     public CartScreenController(Cart cart) {  
39         super();  
40         this.cart = cart;  
41         //         loader.load();  
42     }
```

```

43
44 @FXML
45     private void initialize() {
46         colMediaTitle.setCellValueFactory(
47             new PropertyValueFactory<Media, String>( s: "title"));
48         colMediaCategory.setCellValueFactory(
49             new PropertyValueFactory<Media, String>( s: "category"));
50         colMediaCost.setCellValueFactory(
51             new PropertyValueFactory<Media, Float>( s: "cost"));
52         tblMedia.setItems(this.cart.getItemsOrdered());
53
54         btnPlay.setVisible(false);
55         btnRemove.setVisible(false);
56
57         tblMedia.getSelectionModel().selectedItemProperty().addListener(
58             new ChangeListener<Media>() {
59             @Override
60             public void changed(ObservableValue<? extends Media> observable, Media oldValue,
61                                 Media newValue) {
62                 if (newValue != null) {
63                     updateButtonBar(newValue);
64                 }
65             }
66         });
67     }
68
69     void updateButtonBar(Media media){
70         btnRemove.setVisible(true);
71         btnPlay.setVisible(media instanceof Playable);
72     }

```

Figure 25 Source code of CartScreenController Class

## 6.2. Demo

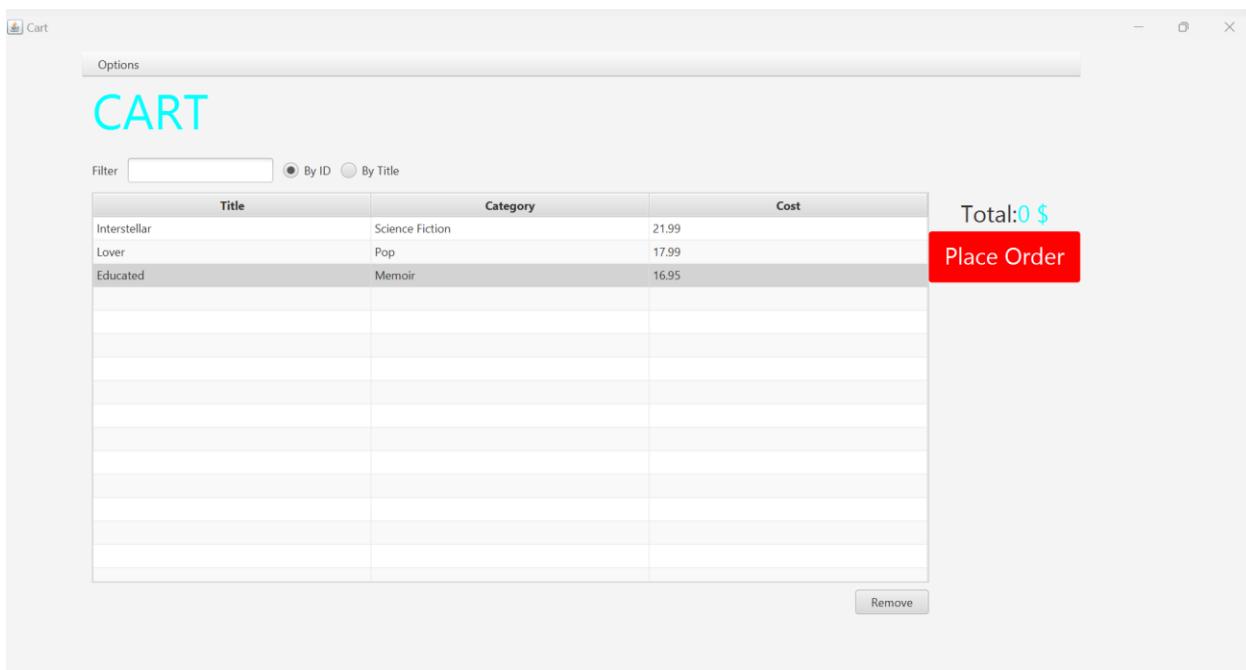
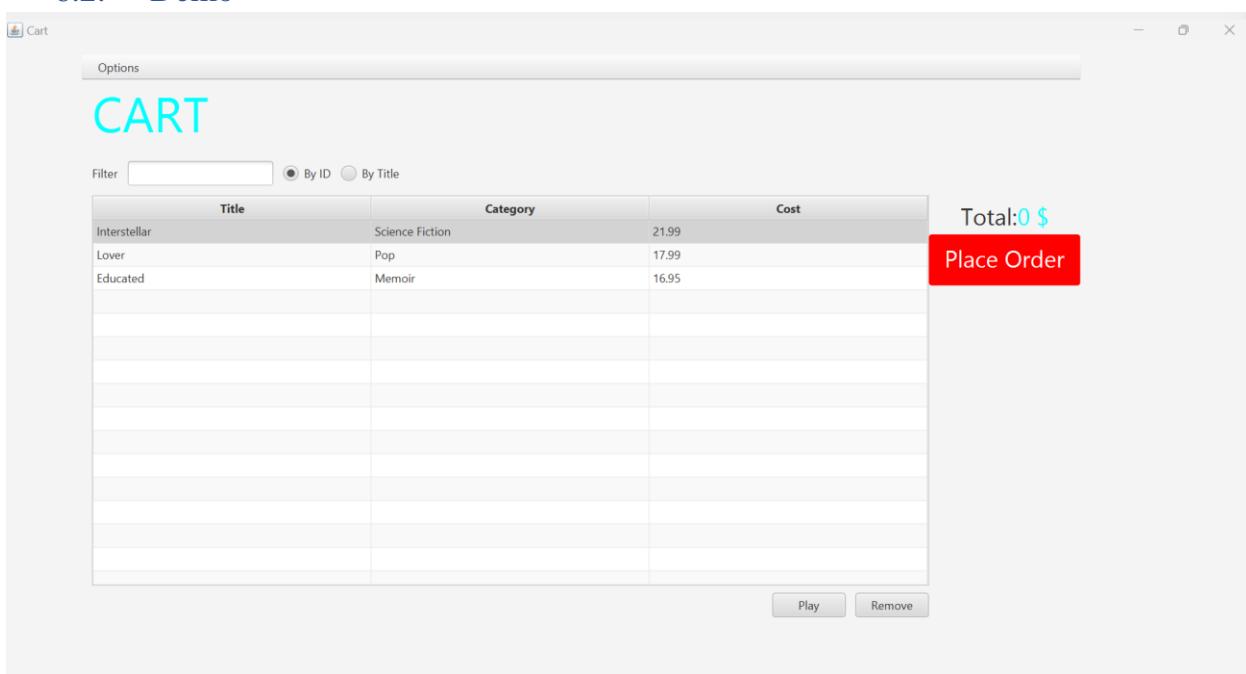


Figure 26 Demo media unplayable

## 7. Deleting a media

### 7.1. Code

```
@FXML  
void btnRemovePressed (ActionEvent event) {  
    Media media = tblMedia.getSelectionModel().getSelectedItem();  
    cart.removeMedia(media);  
}
```

Figure 27 Source code of btnRemovePressed Method

## 7.2. Demo

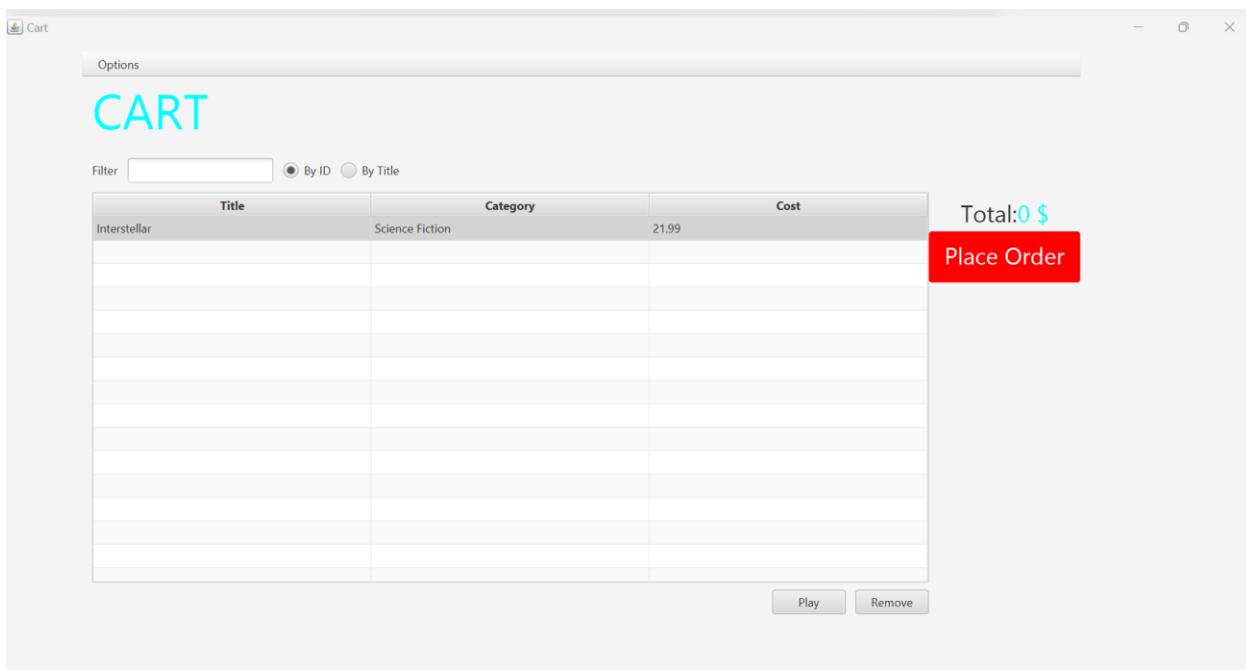
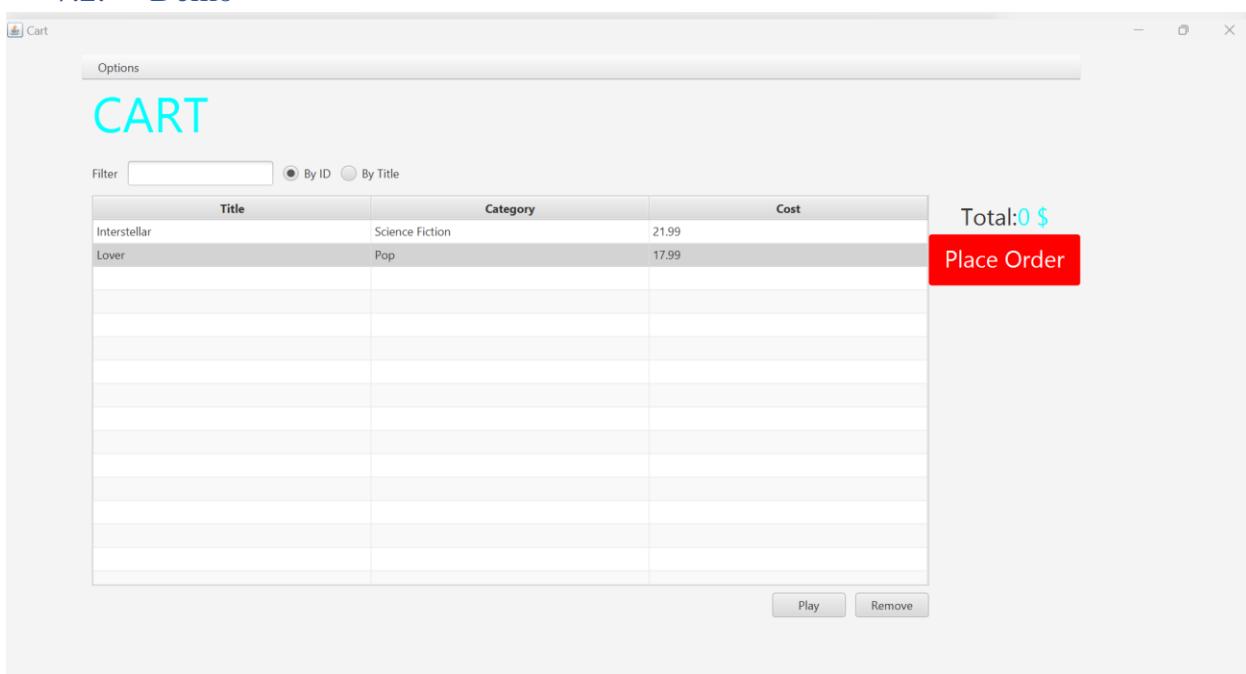


Figure 28 Demo button remove

## 8. Complete the Aims GUI application

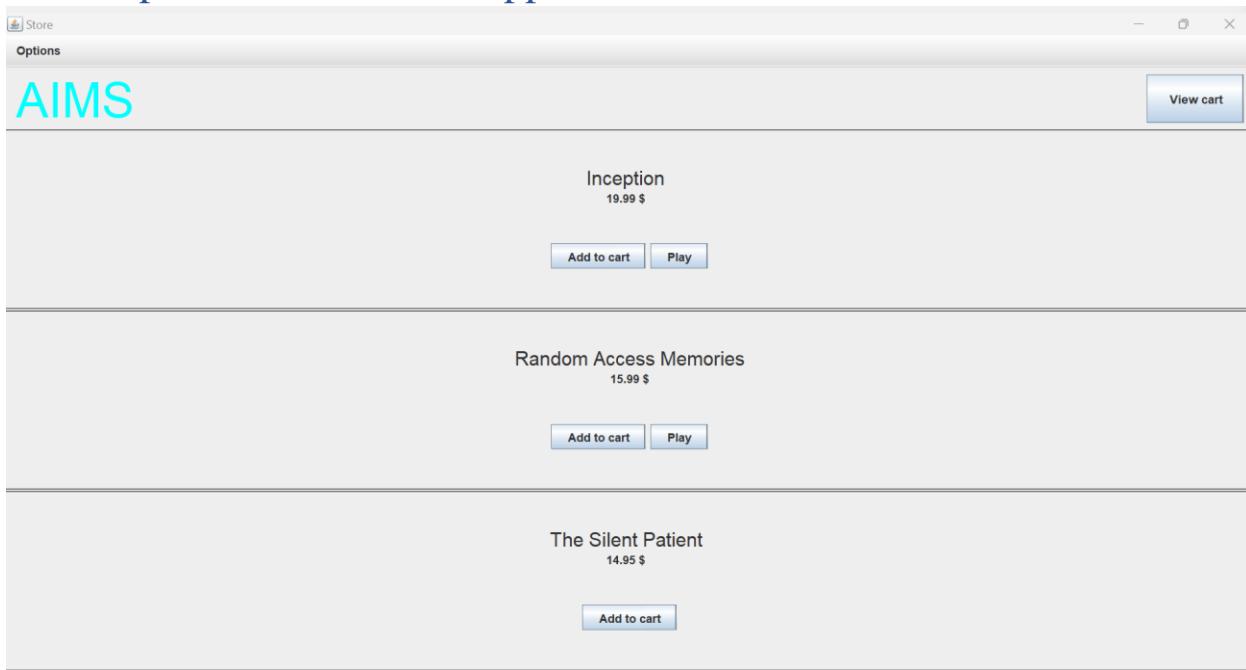


Figure 29 Demo Store before add book

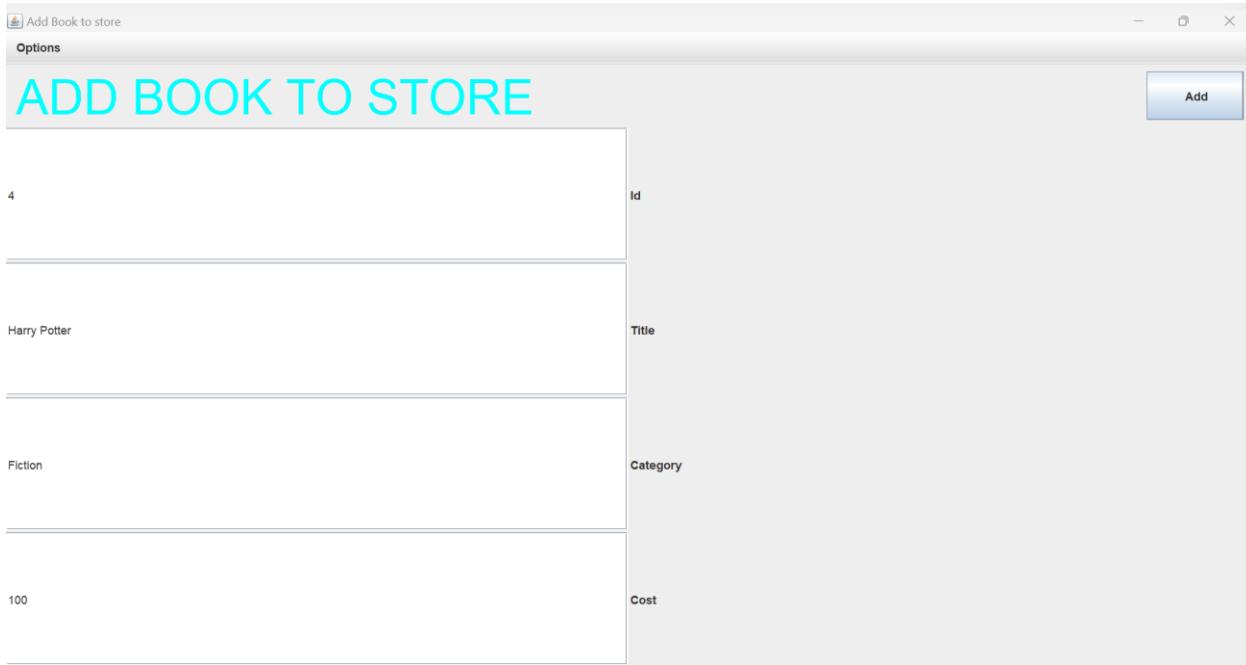


Figure 30 Demo add book

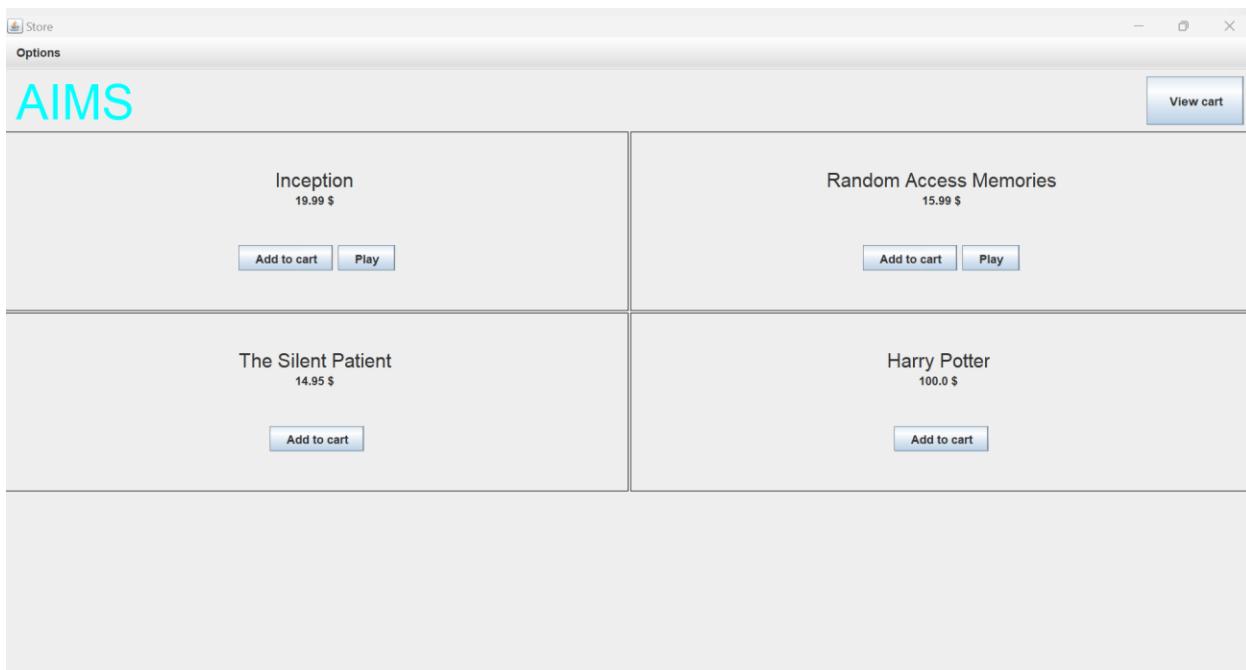


Figure 31 Demo Store after add book

5	Id
Nam cham	Title
Rock	Category
90	Cost
Buc tuong	Director
5	Length
Buc tuong	Artist

Figure 32 Demo add CD

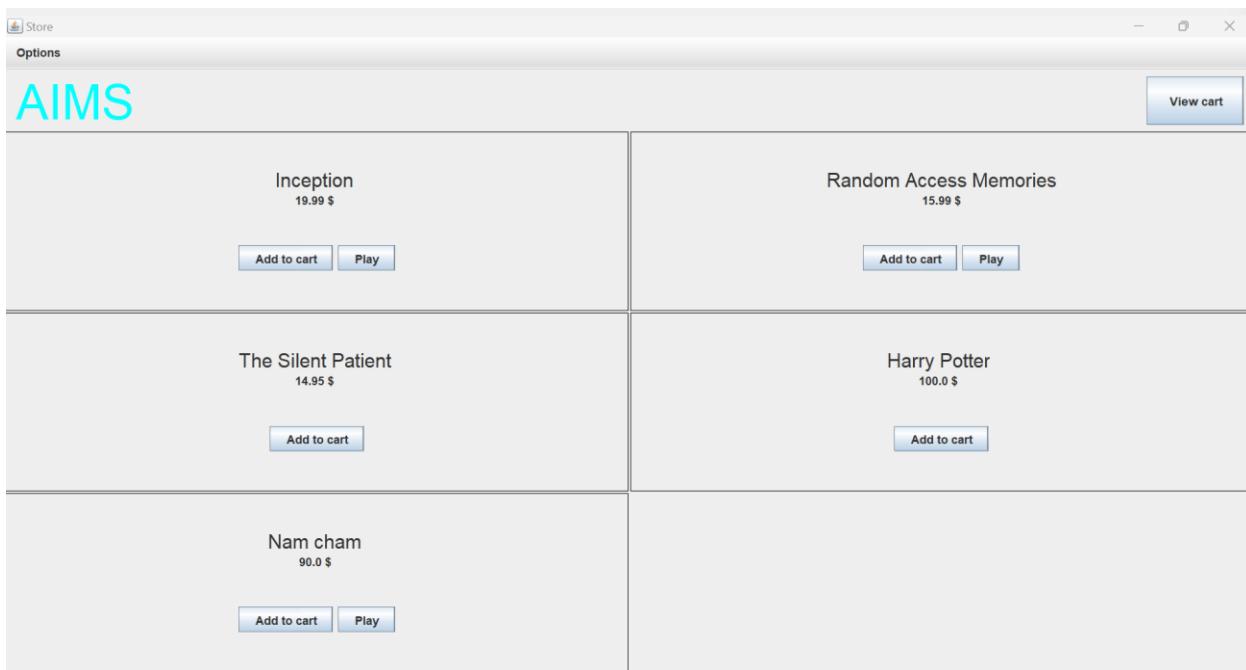


Figure 33 Demo Store after add CD

The screenshot shows a software application window titled "Add DVD to store". At the top left is a "Store" icon, and at the top right is a "Close" button. The main area is titled "ADD DVD TO STORE" in large blue letters. On the right side is a "Add" button. The form consists of six input fields:

Field	Value
Id	6
Title	The Pacific
Category	Documentation
Cost	80
Director	HBO
Length	1000

Figure 34 Demo Store add DVD

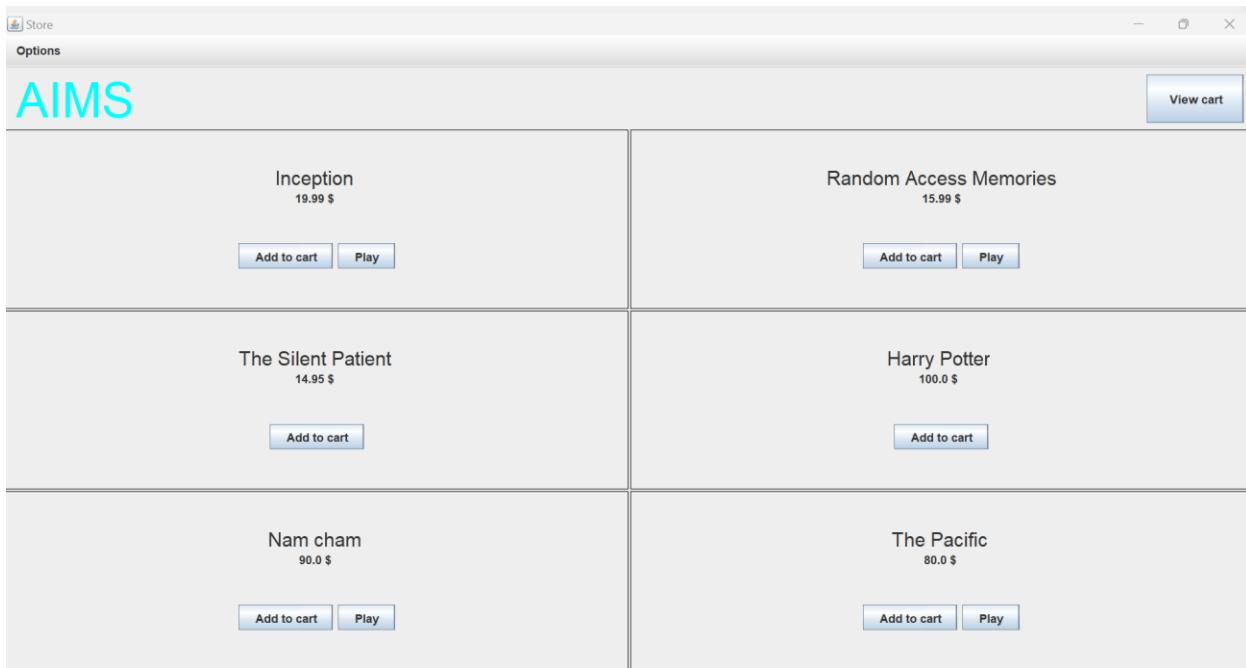


Figure 35 Demo Store after add DVD

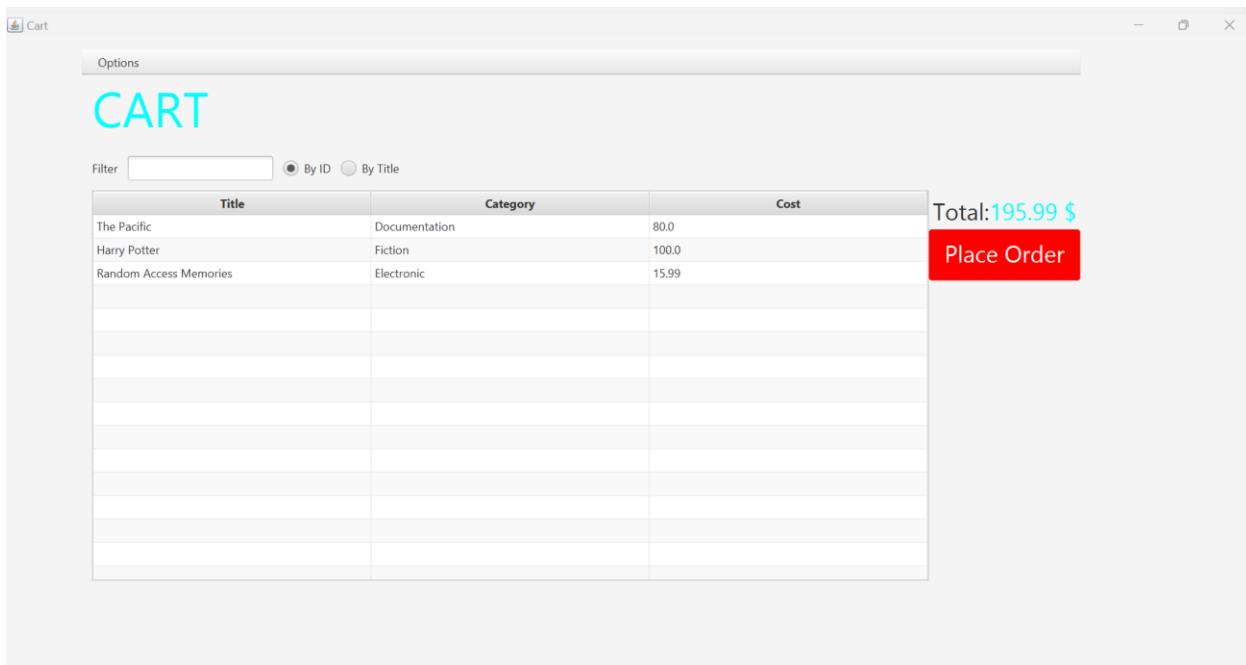


Figure 36 Demo Cart

```
! "AWT-EventQueue-0" java.lang.RuntimeException Create breakpoint : lab05.AimsProject.exception.PlayerException: ERROR: CD length is non-positive!
project.screen.MediaStore$ButtonListener.actionPerformed(MediaStore.java:87) <4 internal lines>
/javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:279) <30 internal lines>
AimsProject.exception.PlayerException Create breakpoint : ERROR: CD length is non-positive!
project.media.CompactDisc.play(CompactDisc.java:73)
project.screen.MediaStore$ButtonListener.actionPerformed(MediaStore.java:85)
```

Figure 37 Catch Exception

## 9. Use case Diagram

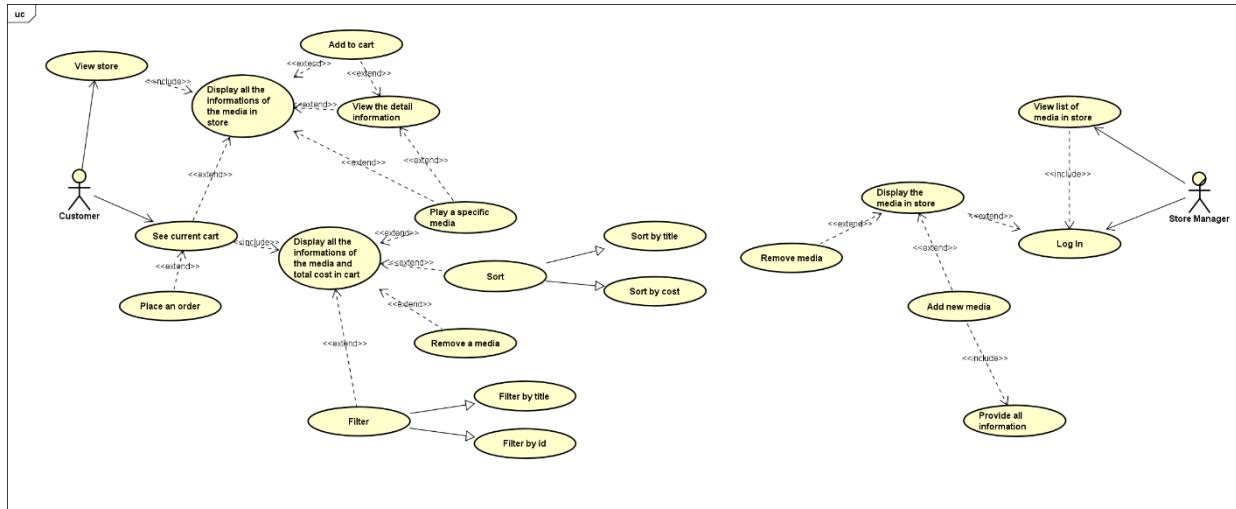


Figure 38 Use case Diagram

## 10. Class Diagram

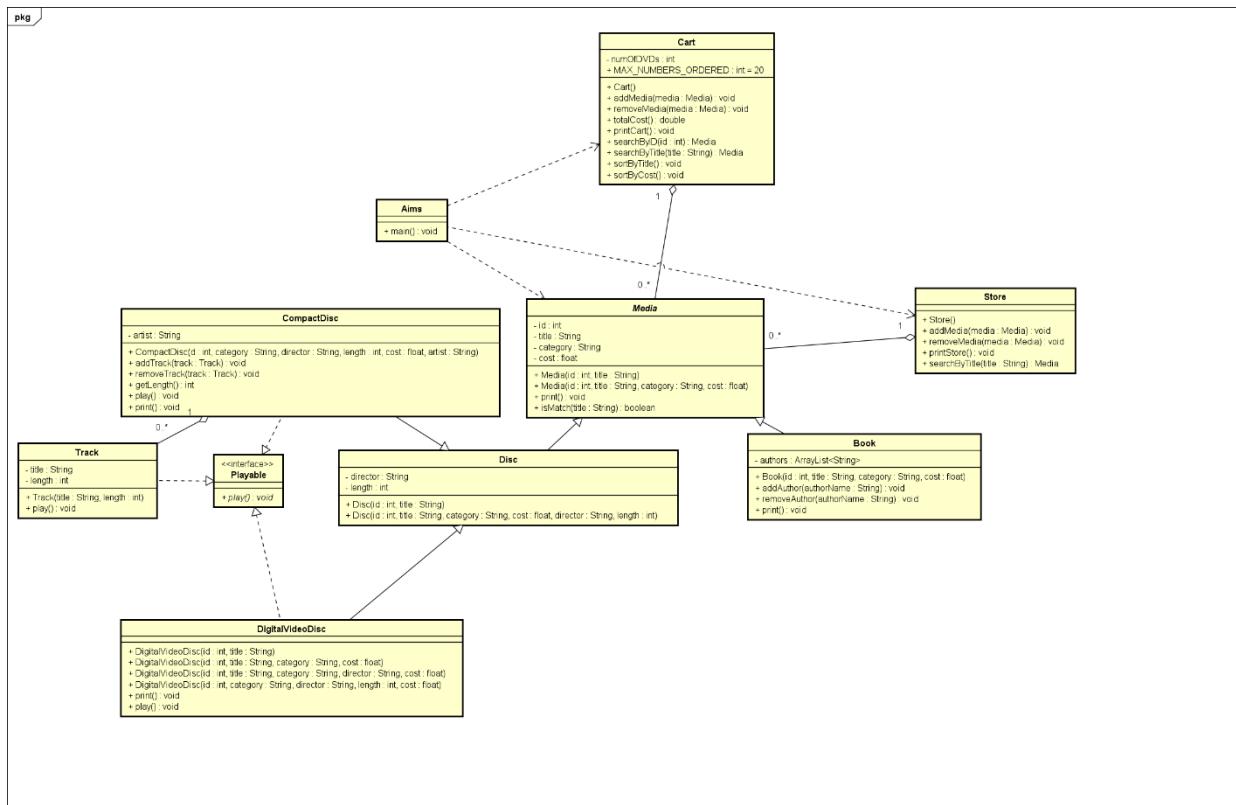


Figure 39 Class Diagram