

BÁO CÁO THỰC HÀNH LAB05

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục nội dung

Mục lục hình ảnh	2
1. Swing components	2
1.1. AWTAccumulator	2
1.2. Demo	4
2. Organizing Swing components with Layout Managers.....	4
2.1. Code	4
2.2. Demo	6
3. Create a graphical user interface for AIMS with SwingCreate class StoreScreen..	6
3.1. Create class MediaStore	6
3.2. Demo	8
4. JavaFX API.....	9
4.1. Create class Painter	9
4.2. Create Painter.fxml.....	9
4.3. Create class PainterController	10
4.4. Demo	11
5. View Cart Screen	12
5.1. Create cart.fxml	12
5.2. Create class CartScreen	14
5.3. Create class CartScreenController	15
5.4. Demo	17
6.2. Demo	19
7. Deleting a media	20
7.1. Code	20
7.2. Demo	20
8. Complete the Aims GUI application	22

9. Use case Diagram	25
10. Class Diagram.....	26

Mục lục hình ảnh

Figure 1: Demo Accumulator Input.....	4
Figure 2: Demo Accumulator Output.....	4
Figure 3: Demo NumberGrid Input 0-9	6
Figure 4: Demo NumberGrid DEL Button	6
Figure 5: Demo NumberGrid C Button	6
Figure 6: Demo Store GUI	8
Figure 7: Demo Painter Pen	11
Figure 8: Demo Painter Eraser	12
Figure 9: Demo Painter Clear	12
Figure 10: Demo Cart UI	17
Figure 11: Demo Cart UI Button _ Playable CD.....	19
Figure 12: Demo Cart UI Button _ Unplayable CD	20
Figure 13: Demo Cart UI Remove Button _ Selected CD	21
Figure 14: Demo Cart UI Remove Button _ Removed CD.....	21
Figure 15: Demo Aims GUI_ Store	22
Figure 16: Demo Aims GUI_ Add Book To Store	22
Figure 17: Demo Aims GUI_ Add CD To Store	23
Figure 18: Demo Aims GUI_ Add DVD To Store.....	23
Figure 19: Demo Aims GUI_ Cart	24
Figure 20: Use Case Diagram	25
Figure 21: Class Diagram.....	26

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.     private TextField tfInput;
9.     private TextField tfOutput;
10.    private int sum = 0;
11.
12.    // Constructor

```

```
13.     public AWTAccumulator() {
14.         setLayout(new GridLayout(2, 2));
15.
16.         add(new Label("Enter an Integer: "));
17.
18.         tfInput = new TextField(10);
19.         add(tfInput);
20.         tfInput. addActionListener(new TFInputListener());
21.
22.         add(new Label("The Accumulated Sum is: "));
23.
24.         tfOutput = new TextField(10);
25.         tfOutput. setEditable(false);
26.         add(tfOutput);
27.
28.         setTitle("AWT Accumulator");
29.         setSize(350, 120);
30.         setVisible(true);
31.     }
32.
33.     public static void main(String[] args) {
34.         new AWTAccumulator();
35.     }
36.     private class TFInputListener implements ActionListener {
37.         @Override
38.         public void actionPerformed(ActionEvent evt) {
39.             int numberIn = Integer.parseInt (tfInput.getText ());
40.             sum += numberIn;
41.             tfInput. setText("");
42.             tfOutput. setText(sum + "");
43.         }
44.     }
45. }
46. SwingAccumulator
47. package LAB05.GUIProject.swing;
48. import javax.swing.*;
49. import java.awt.*;
50. import java.awt.event.ActionEvent;
51. import java.awt.event.ActionListener;
52.
53. public class SwingAccumulator extends JFrame {
54.     // Attribute
55.     private JTextField tfInput;
56.     private JTextField tfOutput;
57.     private int sum = 0;
58.
59.     // Constructor
60.     public SwingAccumulator() {
61.         Container cp = getContentPane();
62.         cp.setLayout(new GridLayout(2, 2));
63.
64.         cp.add(new JLabel("Enter an Integer: "));
65.
66.         tfInput = new JTextField(10);
67.         cp.add(tfInput);
68.         tfInput.addActionListener(new TFInputListener());
69.
70.         cp.add(new JLabel("The Accumulated Sum is: "));
71.
72.         tfOutput = new JTextField(10);
73.         tfOutput. setEditable(false) ;
74.         cp.add(tfOutput);
75.
76.         setTitle("Swing Accumulator");
77.         setSize(350, 120);
78.         setVisible(true);
```

```

79.     }
80.
81.     public static void main(String[] args) {
82.         new SwingAccumulator();
83.     }
84.     private class TFInputListener implements ActionListener {
85.         @Override
86.         public void actionPerformed(ActionEvent evt) {
87.             int numberIn = Integer.parseInt(tfInput.getText ());
88.             sum += numberIn;
89.             tfInput. setText("");
90.             tfOutput.setText(sum + "");
91.         }
92.     }
93. }

```

1.2. Demo

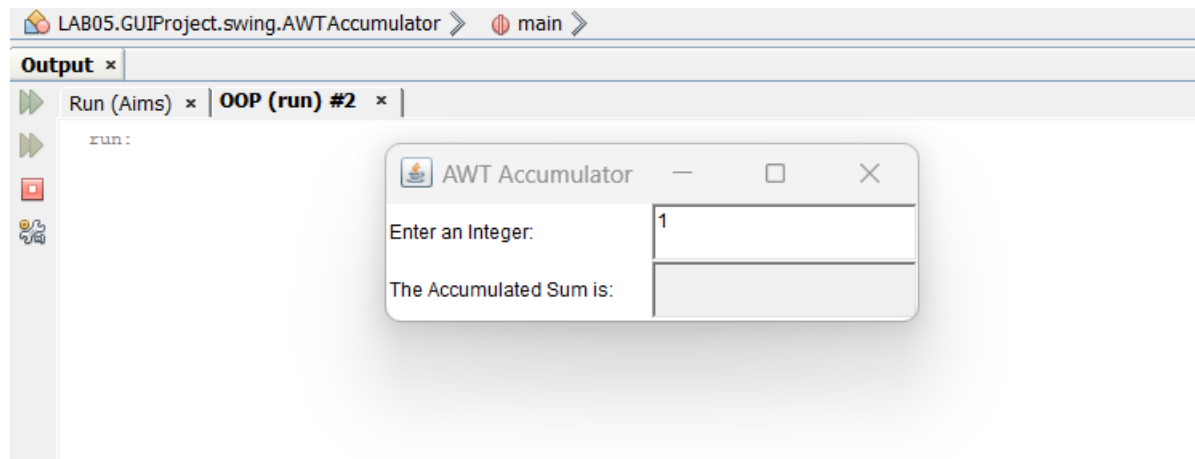


Figure 1: Demo Accumulator Input

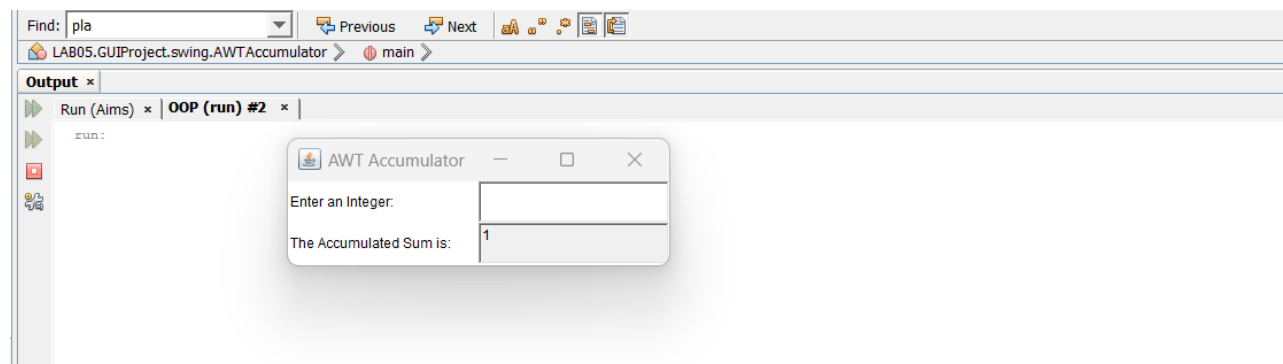


Figure 2: Demo Accumulator Output

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.     private JButton[] btnNumbers = new JButton[10];

```

```
10.     private JButton btnDelete, btnReset;
11.     private JTextField tfDisplay;
12.
13.     // Constructor
14.     public NumberGrid() {
15.         tfDisplay = new JTextField();
16.         tfDisplay.setComponentOrientation(
17.             ComponentOrientation.RIGHT_TO_LEFT);
18.
19.         JPanel panelButtons = new JPanel(new GridLayout(4, 3));
20.         addButtons(panelButtons);
21.
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(200, 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("" + i);
38.             panelButtons.add(btnNumbers[i]);
39.             btnNumbers[i].addActionListener(btnListener);
40.
41.         }
42.
43.         btnDelete = new JButton("DEL");
44.         panelButtons.add(btnDelete);
45.         btnDelete.addActionListener(btnListener);
46.
47.         btnNumbers[0] = new JButton("0");
48.         panelButtons.add(btnNumbers[0]);
49.         btnNumbers[0].addActionListener(btnListener);
50.
51.         btnReset = new JButton("C");
52.         panelButtons.add(btnReset);
53.         btnReset.addActionListener(btnListener);
54.     }
55.
56.     private class ButtonListener implements ActionListener {
57.         @Override
58.         public void actionPerformed(ActionEvent e) {
59.             String button = e.getActionCommand();
60.             if (button.charAt(0) >= '0' && button.charAt(0) <= '9') {
61.                 tfDisplay.setText(tfDisplay.getText() + button);
62.             } else if (button.equals("DEL")) {
63.                 String currentText = tfDisplay.getText();
64.                 String text = currentText.substring(0, currentText.length() - 1);
65.                 tfDisplay.setText(text);
66.             } else {
67.                 tfDisplay.setText("");
68.             }
69.         }
70.     }
71.
72.     public static void main(String[] args) {
73.         new NumberGrid();
74.     }
75. }
```

2.2. Demo

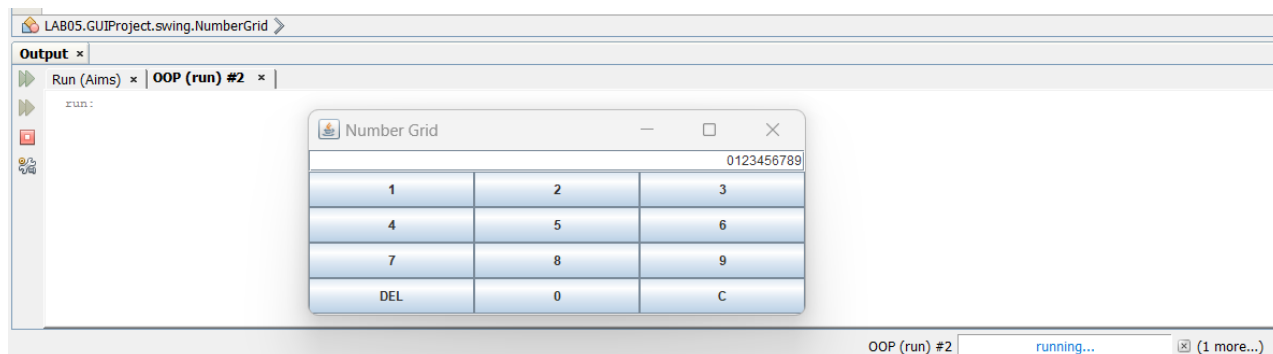


Figure 3: Demo NumberGrid Input 0-9

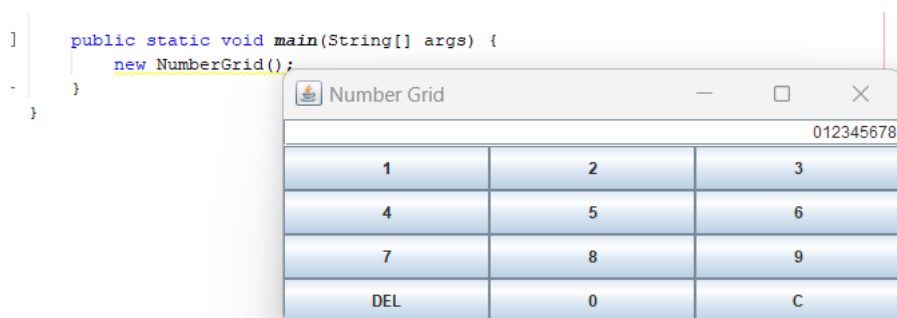


Figure 4: Demo NumberGrid DEL Button

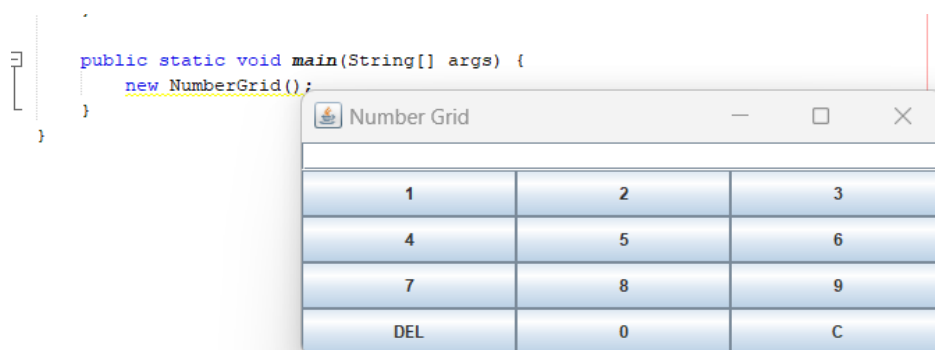


Figure 5: Demo NumberGrid C Button

3. Create a graphical user interface for AIMS with SwingCreate class StoreScreen

3.1. 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.exception.PlayerException;
10. import LAB05.AimsProject.media.*;
11. import java.util.logging.Level;
12. import java.util.logging.Logger;
13.

```

```
14. public class MediaStore extends JPanel {
15.     // Attribute
16.     private Media media;
17.     private static Cart cart;
18.
19.     // Constructor
20.     public MediaStore(Media media) {
21.         this.media = media;
22.         this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
23.
24.         JLabel title = new JLabel(media.getTitle());
25.         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 20));
26.         title.setAlignmentX(CENTER_ALIGNMENT);
27.
28.         JLabel cost = new JLabel("" + media.getCost() + " $");
29.         cost.setAlignmentX(CENTER_ALIGNMENT);
30.
31.         JPanel container = new JPanel();
32.         container.setLayout(new FlowLayout(FlowLayout.CENTER));
33.
34.         ButtonListener buttonListener = new ButtonListener(media);
35.         JButton addButton = new JButton("Add to cart");
36.         container.add(addButton);
37.         addButton.addActionListener(buttonListener);
38.
39.         if (media instanceof Playable) {
40.             JButton playButton = new JButton("Play");
41.             container.add(playButton);
42.             playButton.addActionListener(buttonListener);
43.         }
44.
45.         this.add(Box.createVerticalGlue());
46.         this.add(title);
47.         this.add(cost);
48.         this.add(Box.createVerticalGlue());
49.         this.add(container);
50.
51.         this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
52.     }
53.
54.     // Getter and Setter
55.
56.
57.     public static Cart getCart() {
58.         return cart;
59.     }
60.
61.     public static void setCart(Cart cart) {
62.         MediaStore.cart = cart;
63.     }
64.
65.     // Button Listener for Add to Cart and Play
66.     private class ButtonListener implements ActionListener {
67.         // Attribute
68.         private Media media;
69.         private Disc disc;
70.
71.         // Constructor
72.         public ButtonListener(Media media) {
73.             super();
74.             this.media = media;
75.         }
76.
77.         @Override
78.         public void actionPerformed(ActionEvent e) {
79.             String button = e.getActionCommand();
```

```
80.         if (button.equals("Add to cart")) {
81.             try {
82.                 cart.addMedia(media);
83.             } catch (Exception ex) {
84.                 throw new RuntimeException(ex);
85.             }
86.         } else if (media != null && media instanceof Disc) {
87.             try {
88.                 disc.play();
89.             } catch (PlayerException ex) {
90.                 throw new RuntimeException(ex);
91.             }
92.         }
93.     }
94. }
95. }
96.
```

3.2. Demo

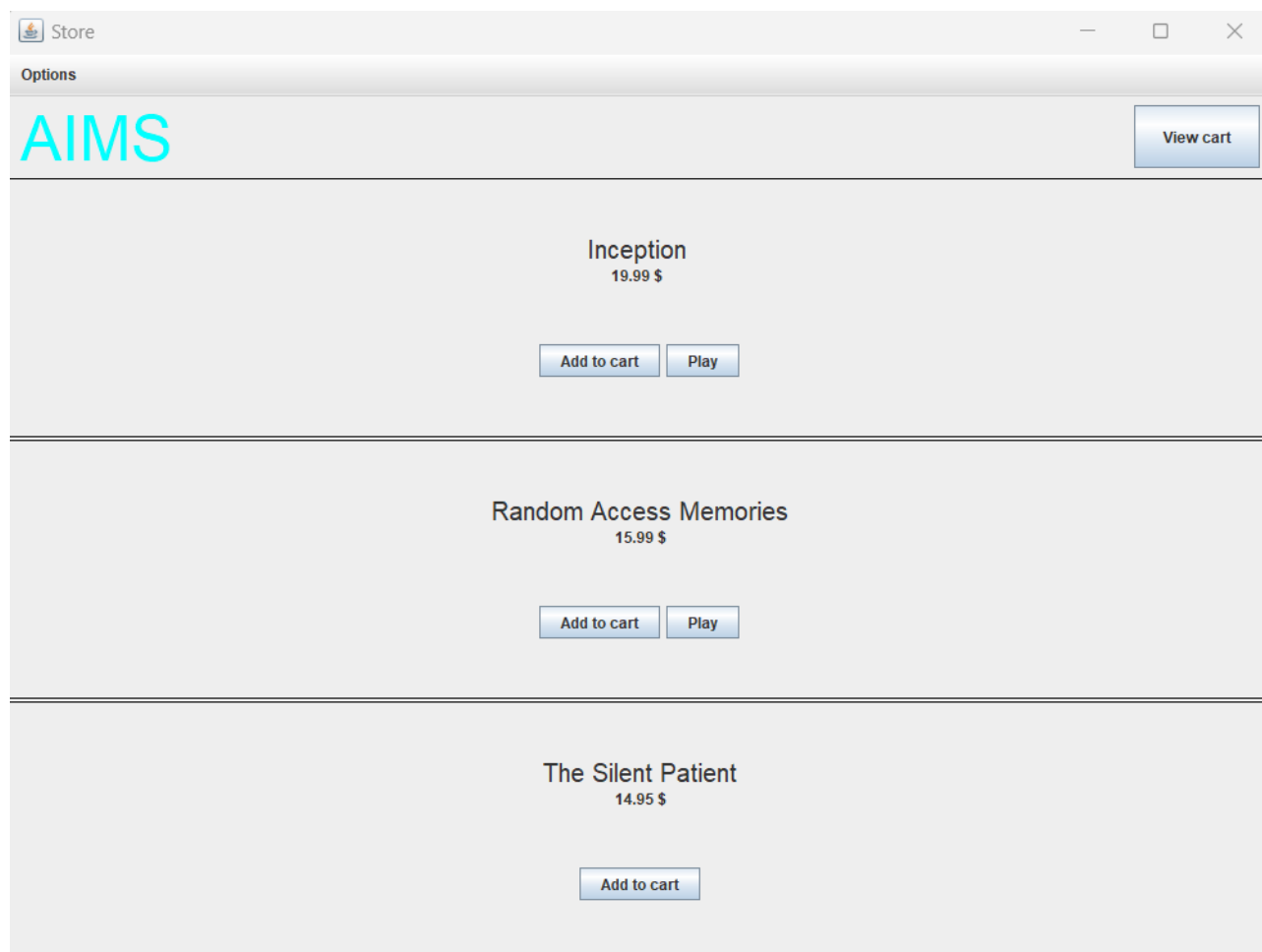


Figure 6: Demo Store GUI

4. JavaFX API

4.1. Create class Painter

```

1. package LAB05.GUIProject.javaafx;
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("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. }
24.

```

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. <AnchorPane prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8"
14.     xmlns:fx="http://javafx.com/fxml/1" fx:controller="lab05.GUIProject.javaafx.PainterController">
15.     <children>
16.         <BorderPane layoutX="251.0" layoutY="4.0" maxHeight="-Infinity" maxWidth="-Infinity"
17.             minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0">
18.             <padding>
19.                 <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
20.             </padding>
21.             <left>
22.                 <VBox maxHeight="1.7976931348623157E308" maxWidth="-Infinity" prefWidth="70.0"
23.                     spacing="8.0" BorderPane.alignment="CENTER">
24.                     <BorderPane.margin>
25.                         <Insets right="8.0" />
26.                     </BorderPane.margin>
27.                     <children>
28.                         <TitledPane animated="false" text="Tools">
29.                             <content>
30.                                 <AnchorPane>
31.                                     <children>
32.                                         <RadioButton mnemonicParsing="false" onAction="#penButtonPressed"
33.                                             selected="true" text="Pen">
34.                                             <toggleGroup>
35.                                                 <ToggleGroup fx:id="identical" />
36.                                             </toggleGroup>
37.                                         </RadioButton>
38.                                     </children>
39.                                 </AnchorPane>
40.                             </content>
41.                         </TitledPane>
42.                     </children>
43.                 </left>
44.             </BorderPane>
45.         </children>
46.     </AnchorPane>
47. </fx:root>
48.

```

```

33.                <padding>
34.                <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
35.                </padding>
36.            </RadioButton>
37.            <RadioButton layoutY="31.0" mnemonicParsing="false"
onAction="#eraserButtonPressed" selected="true" text="Eraser" toggleGroup="$identical">
38.                <padding>
39.                <Insets left="8.0" right="8.0" top="8.0" />
40.                </padding>
41.            </RadioButton>
42.        </children>
43.        <padding>
44.        <Insets bottom="4.0" left="4.0" right="4.0" top="4.0" />
45.        </padding>
46.    </AnchorPane>
47.    </content>
48.    </TitledPane>
49.    <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false"
onAction="#clearButtonPressed" text="Clear">
50.        <VBox.margin>
51.        <Insets />
52.    </VBox.margin>
53.    <padding>
54.    <Insets bottom="4.0" left="4.0" right="4.0" top="4.0" />
55.    </padding>
56.    </Button>
57.    </children>
58.    </VBox>
59.    </left>
60.    <center>
61.        <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" style="-fx-
background-color: white;" BorderPane.alignment="CENTER" />
62.    </center>
63.    </BorderPane>
64.    </children>
65.    </AnchorPane>
66.

```

4.3. Create class PainterController

```

1. package LAB05.GUIProject.javaafx;
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. public class PainterController {
13.     private Paint penColor;
14.     @FXML
15.     private Pane drawingAreaPane;
16.
17.     @FXML
18.     private ToggleGroup identical;
19.
20.     @FXML
21.     void clearButtonPressed(ActionEvent event) {
22.         drawingAreaPane.getChildren().clear();
23.     }
24.
25.     @FXML
26.     void drawingAreaMouseDragged(MouseEvent event) {

```

```
27.         Circle newCircle = new Circle(event.getX(),
28.             event.getY(), 4, 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. }
42.
```

4.4. Demo



Figure 7: Demo Painter Pen



Figure 8: Demo Painter Eraser

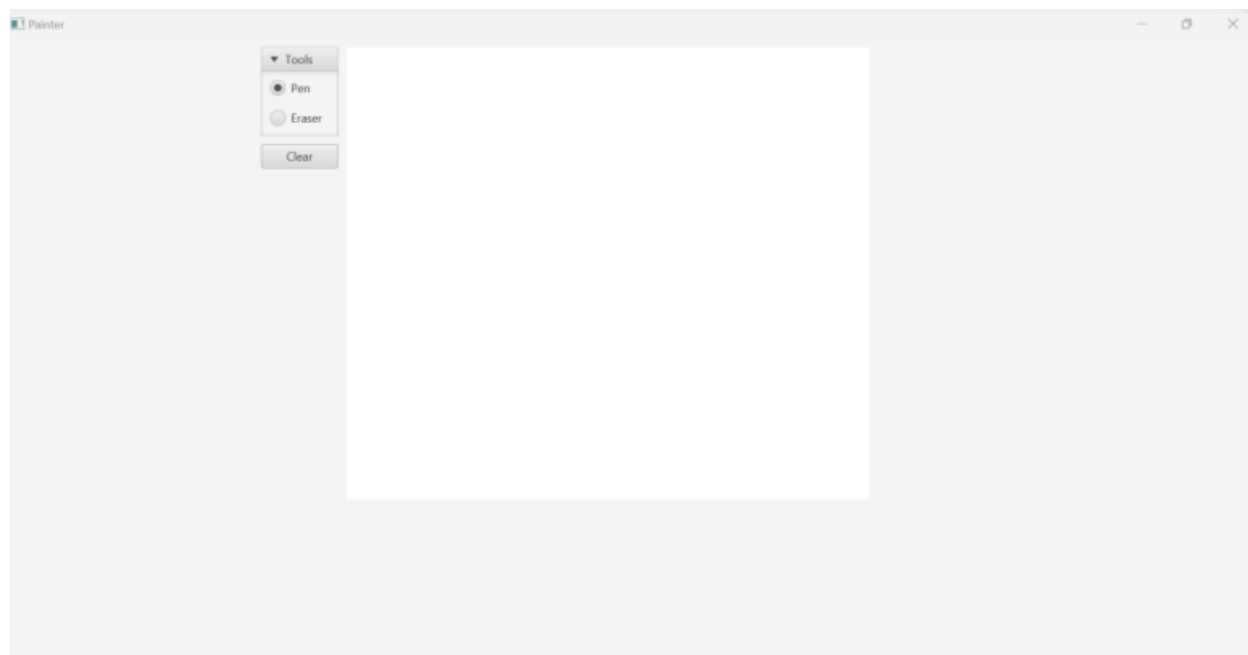


Figure 9: Demo Painter Clear

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.control.MenuBar?>
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/8"
xmlns:fx="http://javafx.com/fxml/1">
22.     <children>
23.         <BorderPane layoutX="79.0" layoutY="14.0" prefHeight="768.0" prefWidth="1024.0">
24.             <top>
25.                 <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
26.                     <children>
27.                         <MenuBar>
28.                             <menus>
29.                                 <Menu mnemonicParsing="false" text="Options">
30.                                     <items>
31.                                         <Menu mnemonicParsing="false" text="Update Store">
32.                                             <items>
33.                                                 <MenuItem mnemonicParsing="false" text="Add Book" />
34.                                                 <MenuItem mnemonicParsing="false" text="Add CD" />
35.                                                 <MenuItem mnemonicParsing="false" text="Add DVD" />
36.                                             </items>
37.                                         </Menu>
38.                                         <MenuItem mnemonicParsing="false" text="View Store" />
39.                                         <MenuItem mnemonicParsing="false" text="View Cart" />
40.                                     </items>
41.                                 </Menu>
42.                             </menus>
43.                         </MenuBar>
44.                         <Label text="CART" textFill="AQUA">
45.                             <font>
46.                                 <Font size="50.0" />
47.                             </font>
48.                             <VBox.margin>
49.                                 <Insets />
50.                             </VBox.margin>
51.                             <padding>
52.                                 <Insets left="10.0" />
53.                             </padding>
54.                         </Label>
55.                     </children>
56.                 </VBox>
57.             </top>
58.             <center>
59.                 <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
60.                     <padding>
61.                         <Insets left="10.0" />
62.                     </padding>
63.                     <children>
64.                         <HBox alignment="CENTER_LEFT" prefWidth="200.0" spacing="10.0">
65.                             <padding>
66.                                 <Insets bottom="10.0" top="10.0" />
67.                             </padding>
68.                             <children>
69.                                 <Label text="Filter" />
70.                                 <TextField fx:id="tfFilter" />
71.                                 <RadioButton fx:id="radioBtnFilterId" mnemonicParsing="false"
selected="true" text="By ID">

```

```

72.                <toggleGroup>
73.                    <ToggleGroup fx:id="filterCategory" />
74.                </toggleGroup>
75.            </RadioButton>
76.            <RadioButton fx:id="radioBtnFilterTitle" mnemonicParsing="false" text="By
Title" toggleGroup="$filterCategory" />
77.        </children>
78.    </HBox>
79.    <TableView fx:id="tblMedia">
80.        <columns>
81.            <TableColumn fx:id="colMediaTitle" prefWidth="75.0" text="Title" />
82.            <TableColumn fx:id="colMediaCategory" prefWidth="75.0" text="Category" />
83.            <TableColumn fx:id="colMediaCost" prefWidth="75.0" text="Cost" />
84.        </columns>
85.        <columnResizePolicy>
86.            <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
87.        </columnResizePolicy>
88.    </TableView>
89.    <ButtonBar prefHeight="40.0" prefWidth="200.0">
90.        <buttons>
91.            <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed"
text="Play" />
92.            <Button fx:id="btnRemove" mnemonicParsing="false"
onAction="#btnRemovePressed" text="Remove" />
93.        </buttons>
94.    </ButtonBar>
95.    </children>
96.    </VBox>
97.    </center>
98.    <right>
99.        <VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
100.            <padding>
101.                <Insets top="50.0" />
102.            </padding>
103.            <children>
104.                <HBox alignment="CENTER">
105.                    <children>
106.                        <Label lineSpacing="10.0" text="Total:">
107.                            <font>
108.                                <Font size="24.0" />
109.                            </font>
110.                        </Label>
111.                        <Label fx:id="lbTotalCost" text="0 $" textFill="AQUA">
112.                            <font>
113.                                <Font size="24.0" />
114.                            </font>
115.                        </Label>
116.                    </children>
117.                </HBox>
118.                <Button mnemonicParsing="false" onAction="#btnPlaceOrderPressed" style="-fx-
background-color: red;" text="Place Order" textFill="WHITE">
119.                    <font>
120.                        <Font size="24.0" />
121.                    </font>
122.                </Button>
123.            </children>
124.        </VBox>
125.    </right>
126.    </BorderPane>
127.    </children>
128.    </AnchorPane>

```

129.

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. public class CartScreen extends JFrame {
15.     private Cart cart;
16.
17.     public CartScreen(Cart cart) {
18.         super();
19.
20.         this.cart = cart;
21.
22.         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("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. }
44.

```

5.3. Create class CartScreenController

```

1. package LAB05.AimsProject.screen;
2. import javafx.beans.value.ChangeListener;
3. import javafx.beans.value.ObservableValue;
4. import javafx.event.ActionEvent;
5. import javafx.fxml.FXML;
6. import javafx.fxml.FXMLLoader;
7. import javafx.scene.control.*;
8. import LAB05.AimsProject.Cart;
9. import LAB05.AimsProject.media.Media;
10. import javafx.scene.control.cell.PropertyValueFactory;
11. import LAB05.AimsProject.media.Playable;
12.
13. public class CartScreenController {
14.     private Cart cart;
15.
16.     @FXML
17.     private TableView<Media> tblMedia;
18.
19.     @FXML
20.     private TableColumn<Media, String> colMediaTitle;

```

```

21.
22.     @FXML
23.     private TableColumn<Media, String> colMediaCategory;
24.
25.     @FXML
26.     private TableColumn<Media, Float> colMediaCost;
27.
28.     @FXML
29.     private Button btnPlay;
30.
31.     @FXML
32.     private Button btnRemove;
33.
34.     @FXML
35.     private Label lbTotalCost;
36.
37.     public CartScreenController(Cart cart) {
38.         super();
39.         this.cart = cart;
40.         // loader.load();
41.     }
42.
43.     @FXML
44.     private void initialize() {
45.         colMediaTitle.setCellValueFactory(
46.             new PropertyValueFactory<Media, String>("title"));
47.         colMediaCategory.setCellValueFactory(
48.             new PropertyValueFactory<Media, String>("category"));
49.         colMediaCost.setCellValueFactory(
50.             new PropertyValueFactory<Media, Float>("cost"));
51.         tblMedia.setItems(this.cart.getItemsOrdered());
52.         lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");
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.     }
73.
74.     @FXML
75.     void btnRemovePressed (ActionEvent event) throws Exception {
76.         Media media = tblMedia.getSelectionModel().getSelectedItem();
77.         cart.removeMedia(media);
78.         lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");
79.     }
80.
81.     @FXML
82.     void btnPlaceOrderPressed(ActionEvent event) {
83.
84.     }
85.

```



```

86.     @FXML
87.     void btnPlayPressed(ActionEvent event) {
88.
89.     }
90. }
91.

```

5.4. Demo

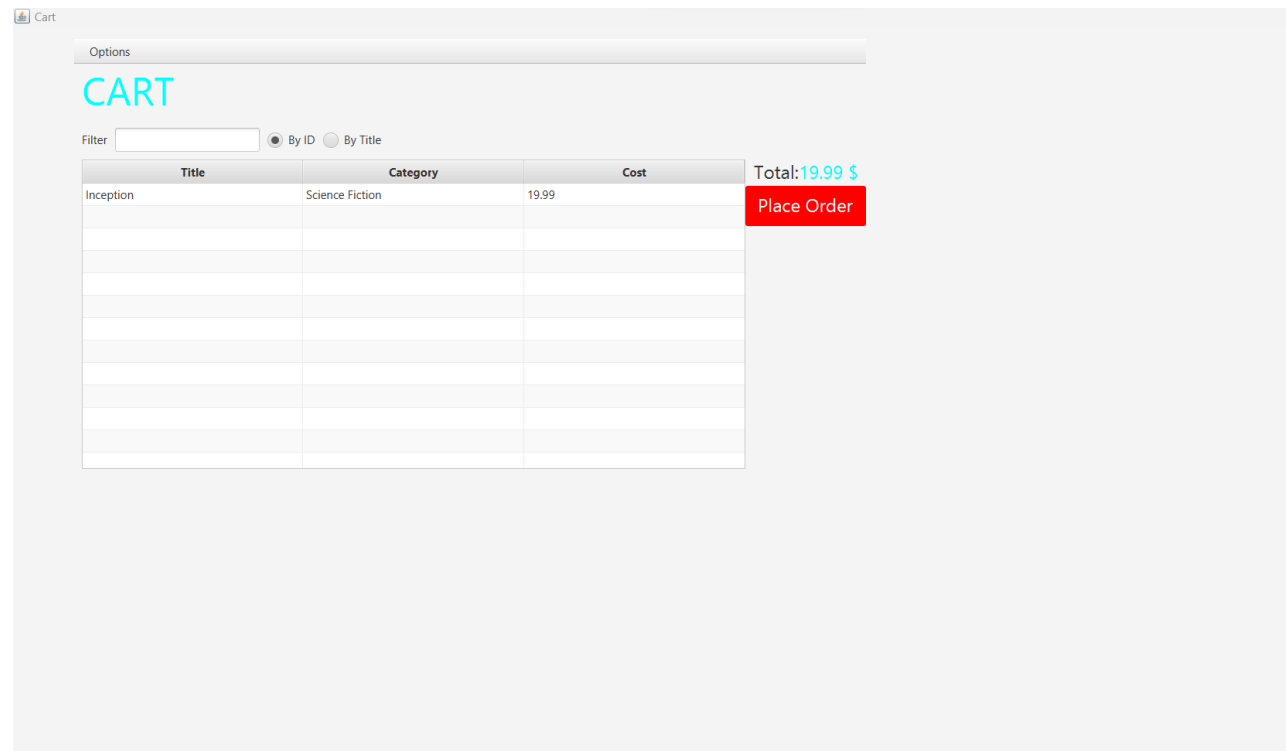


Figure 10: Demo Cart UI

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.event.ActionEvent;
5. import javafx.fxml.FXML;
6. import javafx.fxml.FXMLLoader;
7. import javafx.scene.control.*;
8. import LAB05.AimsProject.Cart;
9. import LAB05.AimsProject.media.Media;
10. import javafx.scene.control.cell.PropertyValueFactory;
11. import LAB05.AimsProject.media.Playable;
12.
13. public class CartScreenController {
14.     private Cart cart;
15.
16.     @FXML
17.     private TableView<Media> tblMedia;
18.

```

```

19.    @FXML
20.    private TableColumn<Media, String> colMediaTitle;
21.
22.    @FXML
23.    private TableColumn<Media, String> colMediaCategory;
24.
25.    @FXML
26.    private TableColumn<Media, Float> colMediaCost;
27.
28.    @FXML
29.    private Button btnPlay;
30.
31.    @FXML
32.    private Button btnRemove;
33.
34.    @FXML
35.    private Label lbTotalCost;
36.
37.    public CartScreenController(Cart cart) {
38.        super();
39.        this.cart = cart;
40.        // loader.load();
41.    }
42.
43.    @FXML
44.    private void initialize() {
45.        colMediaTitle.setCellValueFactory(
46.            new PropertyValueFactory<Media, String>("title");
47.        colMediaCategory.setCellValueFactory(
48.            new PropertyValueFactory<Media, String>("category");
49.        colMediaCost.setCellValueFactory(
50.            new PropertyValueFactory<Media, Float>("cost");
51.        tblMedia.setItems(this.cart.getItemsOrdered());
52.        lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");
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.    }
73.
74.    @FXML
75.    void btnRemovePressed (ActionEvent event) throws Exception {
76.        Media media = tblMedia.getSelectionModel().getSelectedItem();
77.        cart.removeMedia(media);
78.        lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");
79.    }
80.
81.    @FXML
82.    void btnPlaceOrderPressed(ActionEvent event) {
83.

```

```
84.     }
85.
86.     @FXML
87.     void btnPlayPressed(ActionEvent event) {
88.
89.     }
90. }
```

91.

6.2. Demo

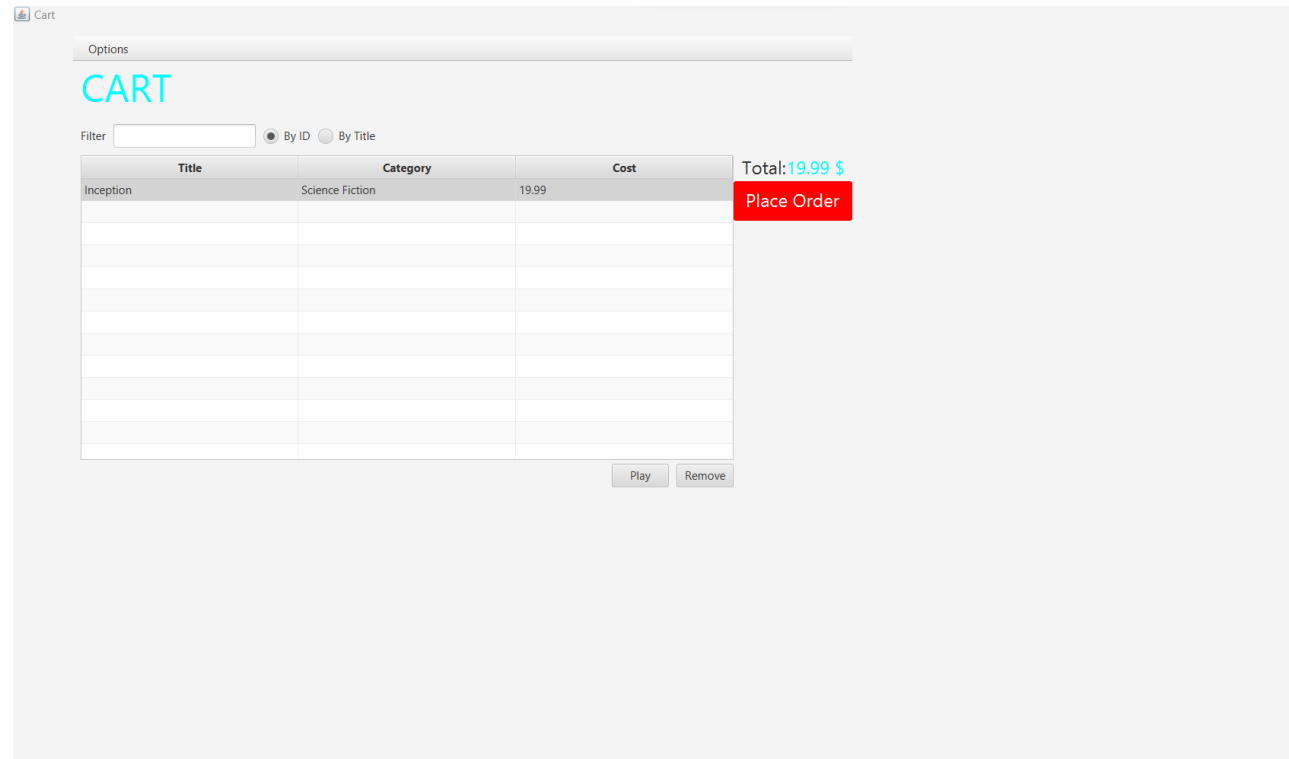


Figure 11: Demo Cart UI Button _ Playable CD

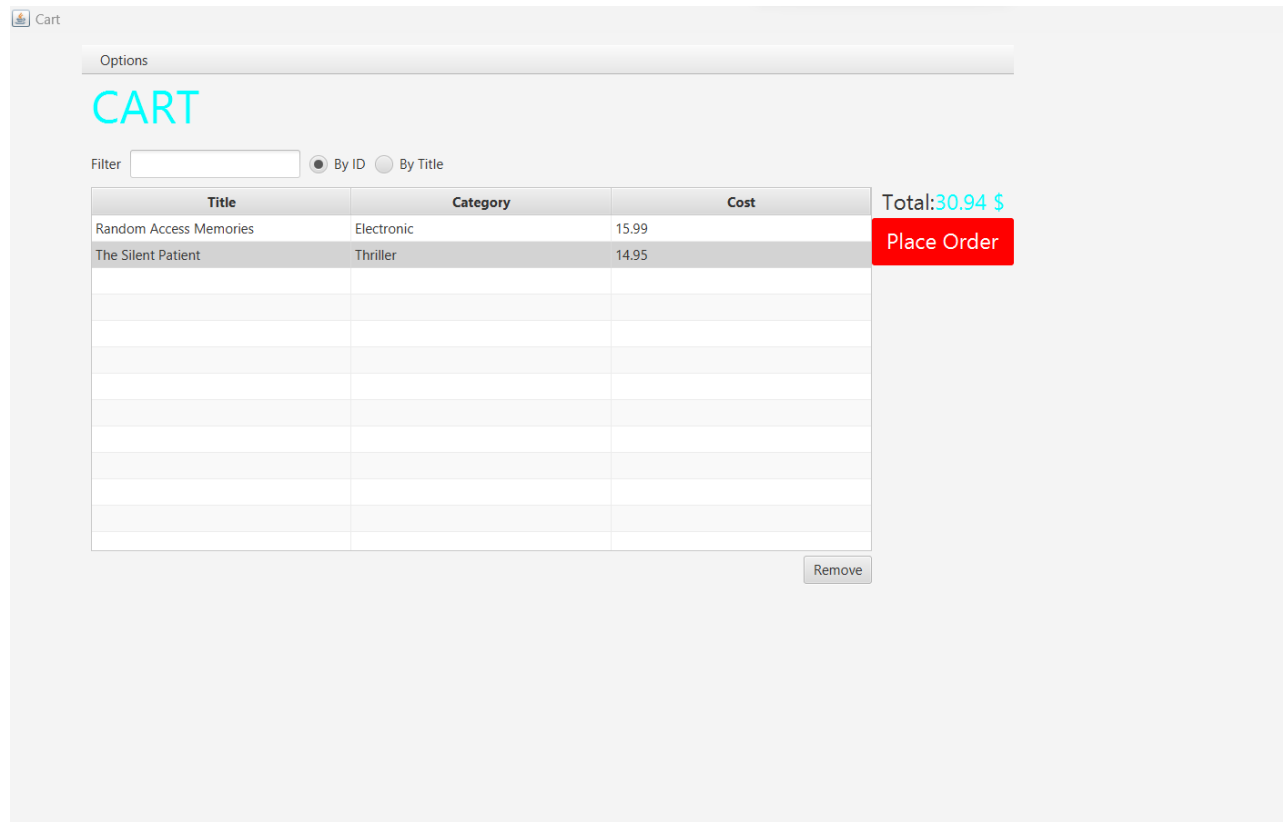


Figure 12: Demo Cart UI Button _ Unplayable CD

7. Deleting a media

7.1. Code

```

1. @FXML
2. void btnRemovePressed (ActionEvent event) throws Exception {
3.     Media media = tblMedia.getSelectionModel().getSelectedItem();
4.     cart.removeMedia(media);
5.     lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");
6. }

```

7.2. Demo

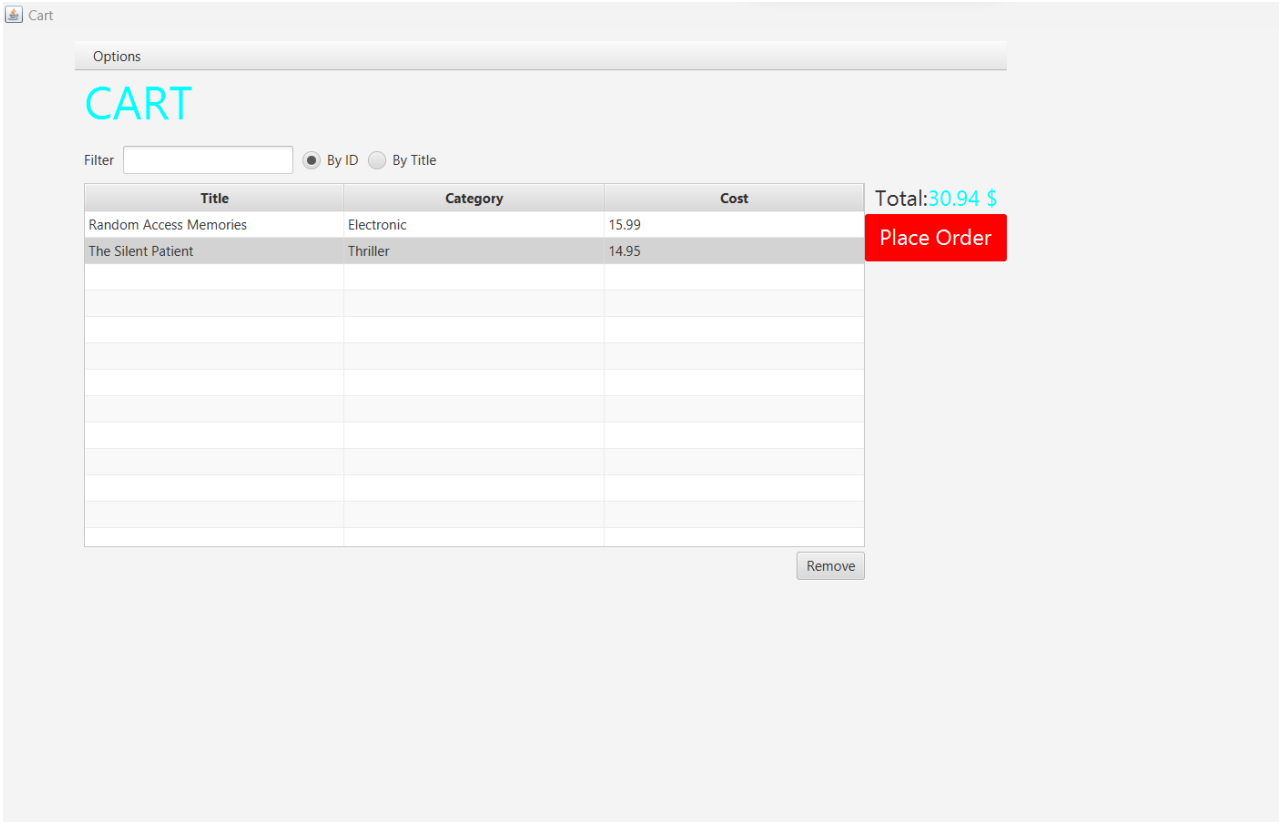


Figure 13: Demo Cart UI Remove Button _ Selected CD

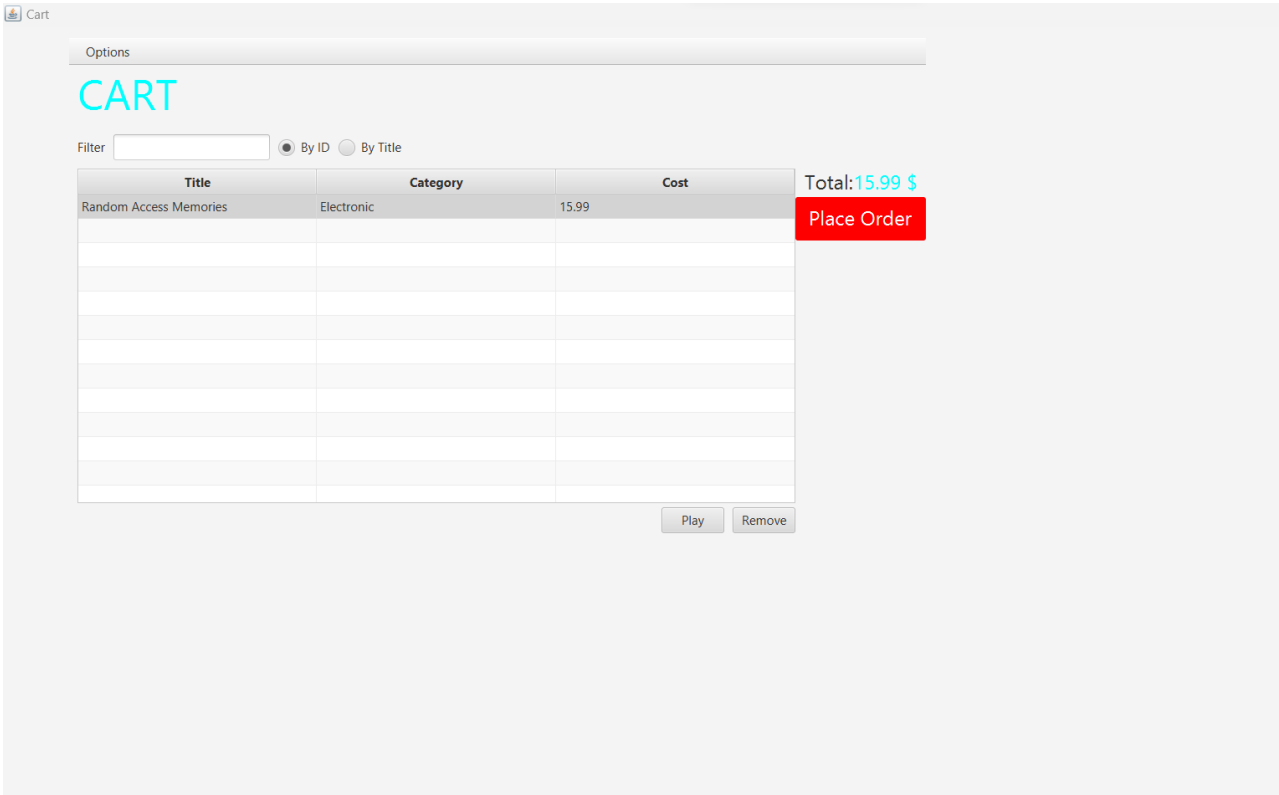


Figure 14: Demo Cart UI Remove Button _ Removed CD

8. Complete the Aims GUI application

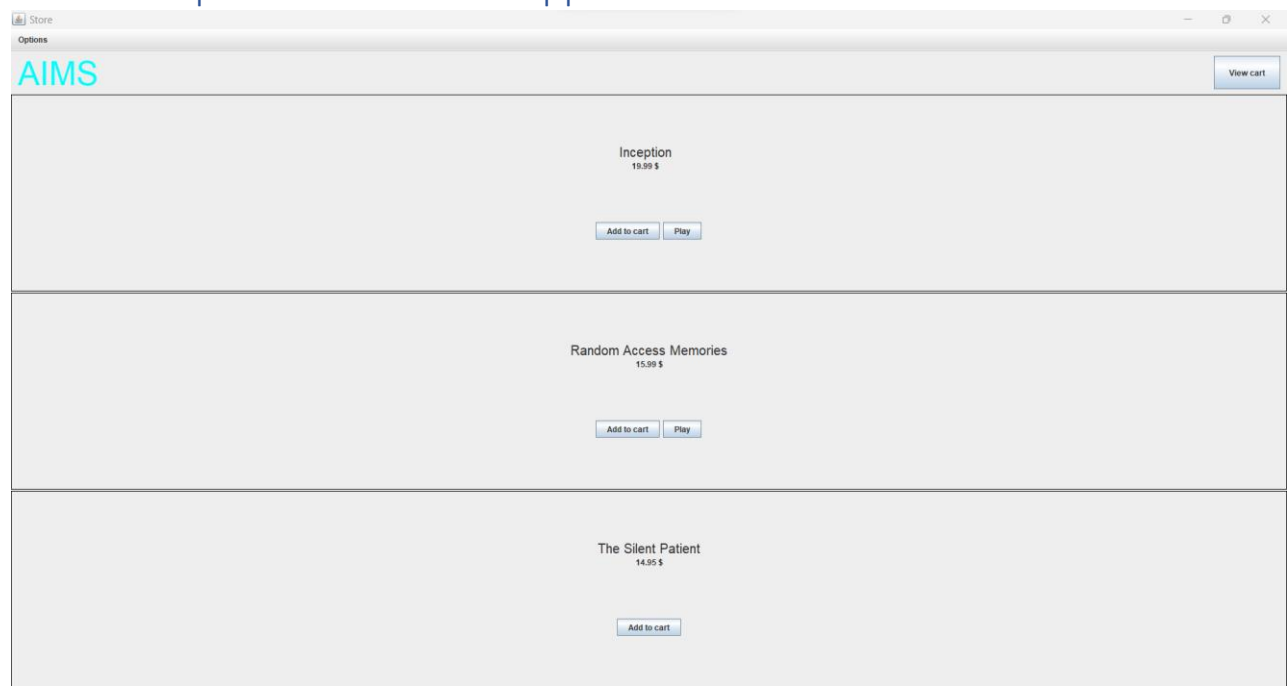


Figure 15: Demo Aims GUI_ Store

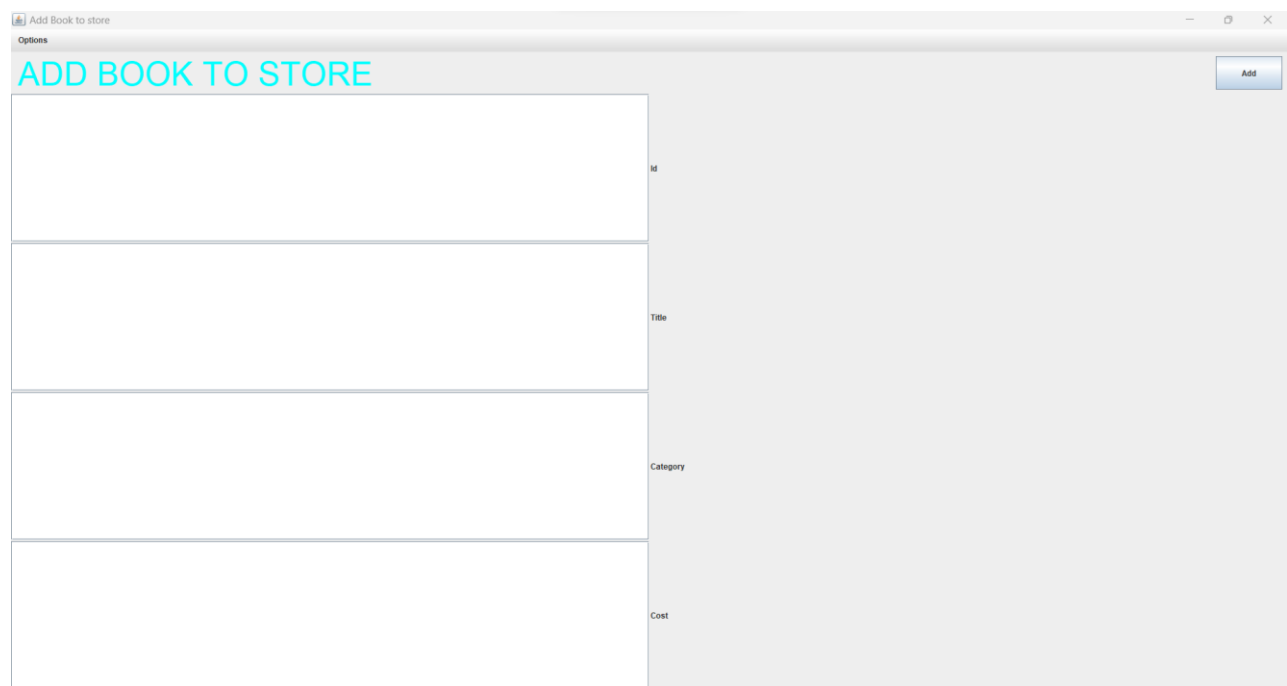


Figure 16: Demo Aims GUI_ Add Book To Store

Add CD to store

Options

ADD CD TO STORE

Id

Title

Category

Cost

Director

Length

Artist

Add

Figure 17: Demo Aims GUI_Add CD To Store

Add DVD to store

Options

ADD DVD TO STORE

Id

Title

Category

Cost

Director

Length

Add

Figure 18: Demo Aims GUI_Add DVD To Store

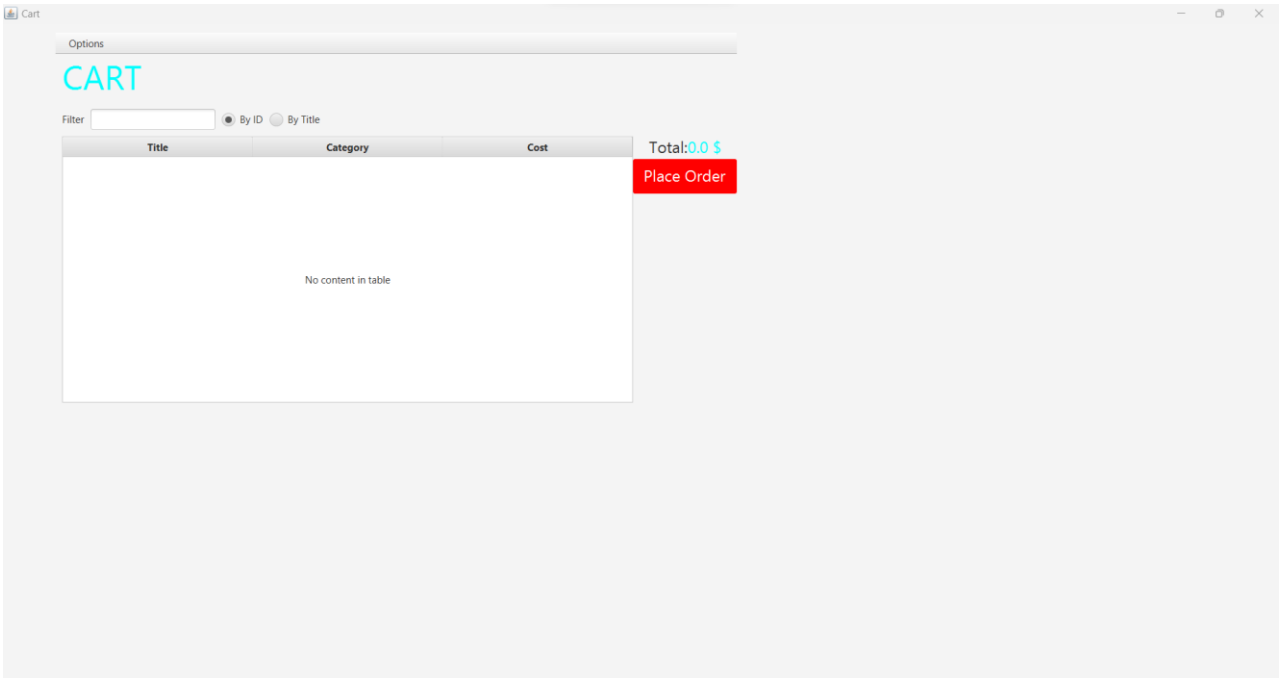


Figure 19: Demo Aims GUI_ Cart

9. Use case Diagram

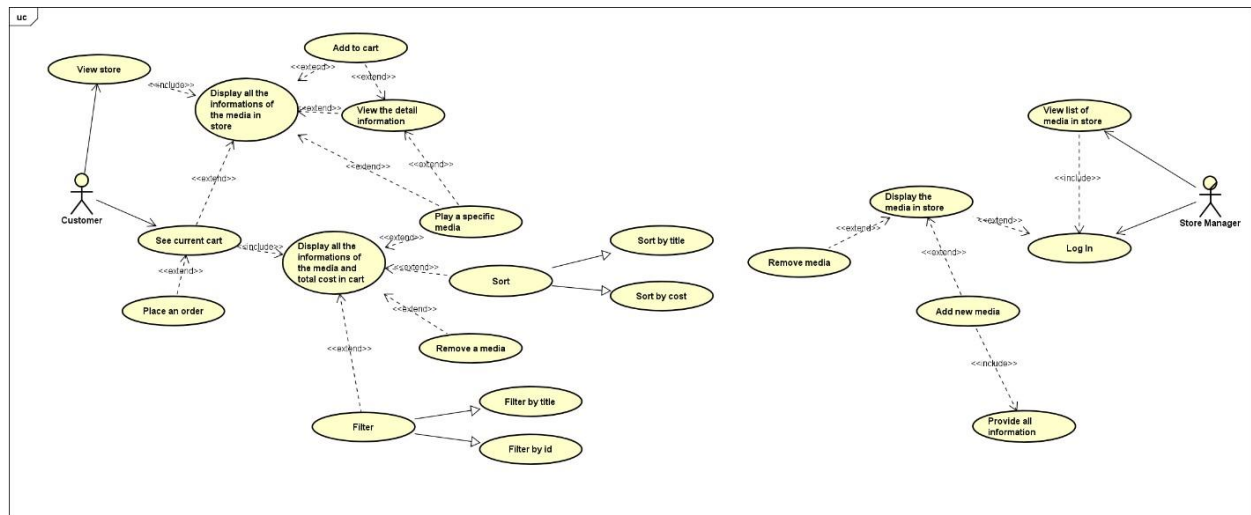


Figure 20: Use Case Diagram

10. Class Diagram

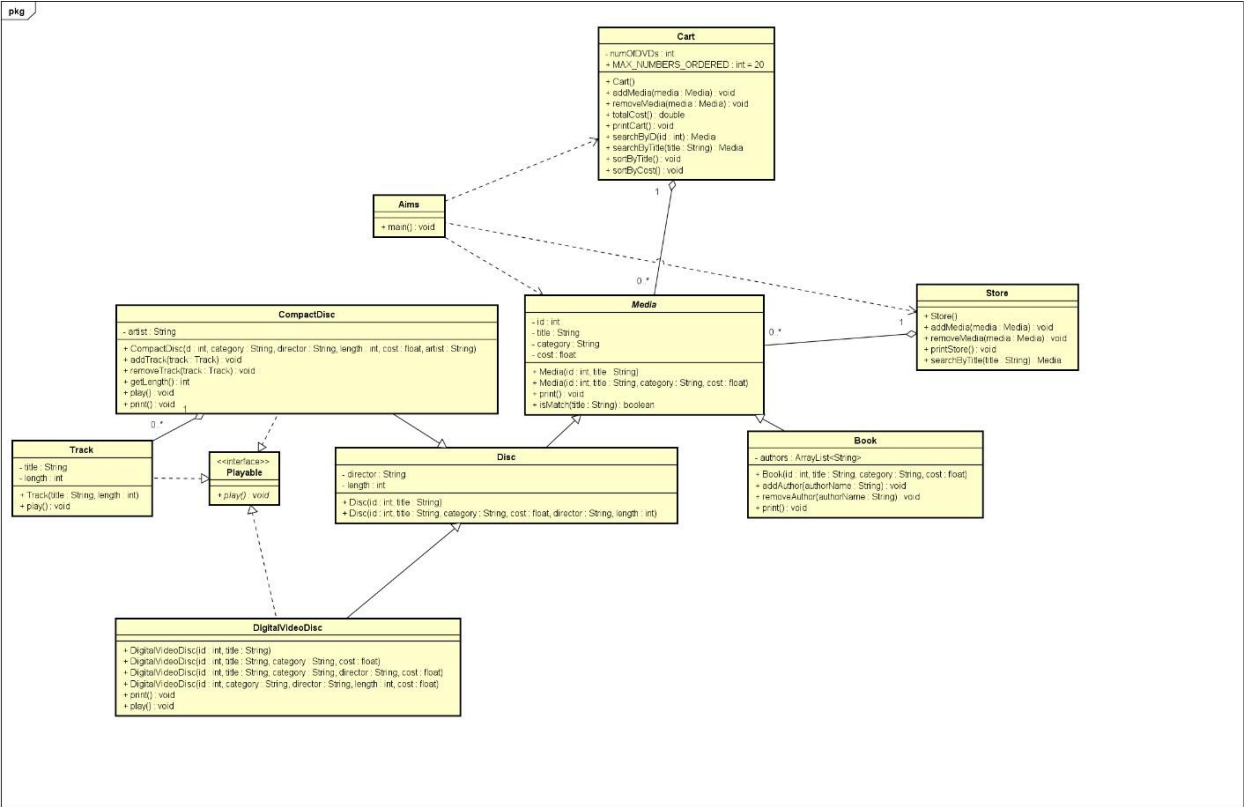


Figure 21: Class Diagram