

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

Mục lục nội dung

1.	Swing components.....	4
1.1.	AWTAccumulator	4
1.2.	SwingAccumulator	6
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	12
3.1.	Create class StoreScreen.....	12
3.2.	Create class MediaStore	18
3.3.	Demo.....	21
4.	JavaFX API	23
4.1.	Create class Painter.....	23
4.2.	Create Painter.fxml	24
4.3.	Create class PainterController	26
4.4.	Demo.....	27
5.	View Cart Screen.....	29
5.1.	Create cart.fxml	29
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.....	42
8.	Complete the Aims GUI application.....	43
9.	Use case Diagram.....	47
10.	Class Diagram.....	48

Mục lục hình ảnh

Figure 1 Source code of AWTAccumulator	4
Figure 2 Source code of AWTAccumulator	5
Figure 3 Demo of AWTAccumulator	5
Figure 4 Source code of SwingAccumulator	6
Figure 5 Source code of SwingAccumulator	7
Figure 6 Demo of SwingAccumulator	7
Figure 7 Source code of NumberGrid	8
Figure 8 Source code of NumberGrid	9
Figure 9 Source code of NumberGrid	10
Figure 10 Source code of NumberGrid	10
Figure 11 Demo buttons 0-9	11
Figure 12 Demo button DEL.....	11
Figure 13 Demo button C.....	11
Figure 14 Source code of StoreScreen Class	12
Figure 15 Source code of StoreScreen Class.....	13
Figure 16 Source code of StoreScreen Class	14
Figure 17 Source code of StoreScreen Class	15
Figure 18 Source code of StoreScreen Class	16
Figure 19 Source code of StoreScreen Class	16
Figure 20 Source code of StoreScreen Class	17
Figure 21 Source code of MediaStore Class.....	18
Figure 22 Source code of MediaStore Class.....	19
Figure 23 Source code of MediaStore Class.....	20
Figure 24 Source code of MediaStore Class.....	21
Figure 25 Demo of Store Screen	21
Figure 26 Demo Add to cart button.....	22
Figure 27 Demo Play button	22
Figure 28 Demo View cart button.....	22
Figure 29 Source code of Painter Class	23
Figure 30 Source code of Painter.fxml	24
Figure 31 Source code of Painter.fxml.....	25
Figure 32 Source code of Painter.fxml.....	25
Figure 33 Source code of Painter.fxml.....	26
Figure 34 Source code of PainterController Class.....	26
Figure 35 Source code of PainterController Class.....	27
Figure 36 Demo Pen.....	27
Figure 37 Demo Eraser.....	28
Figure 38 Demo Clear button.....	28
Figure 39 Source code of cart.fxml	29
Figure 40 Source code of cart.fxml	30
Figure 41 Source code of cart.fxml	30

Figure 42 Source code of cart.fxml	31
Figure 43 Source code of cart.fxml	31
Figure 44 Source code of cart.fxml	32
Figure 45 Source code of CartScreen Class.....	33
Figure 46 Source code of CartScreen Class.....	34
Figure 47 Source code of CartScreenController Class	35
Figure 48 Source code of CartScreenController Class	36
Figure 49 Demo View Cart Screen	37
Figure 50 Source code of CartScreenController Class	38
Figure 51 Source code of CartScreenController	39
Figure 52 Source code of CartScreenController	40
Figure 53 Source code of CartScreenController	40
Figure 54 Demo media playable	41
Figure 55 Demo media unplayable	41
Figure 56 Source code of btnRemovePressed Method	42
Figure 57 Demo button Remove.....	42
Figure 58 Demo button remove	43
Figure 59 Demo Store before add book.....	43
Figure 60 Demo add book.....	44
Figure 61 Demo Store after add book.....	44
Figure 62 Demo add CD	45
Figure 63 Demo Store after add CD	45
Figure 64 Demo Store add DVD	46
Figure 65 Demo Store after add DVD	46
Figure 66 Demo Cart	47
Figure 67 Catch Exception.....	47
Figure 68 Use case Diagram	47
Figure 69 Class Diagram	48

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 TFIinputListener());
```

Figure 1 Source code of AWTAccumulator

```
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     1 usage  
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     }  
46 }
```

Figure 2 Source code of AWTAccumulator

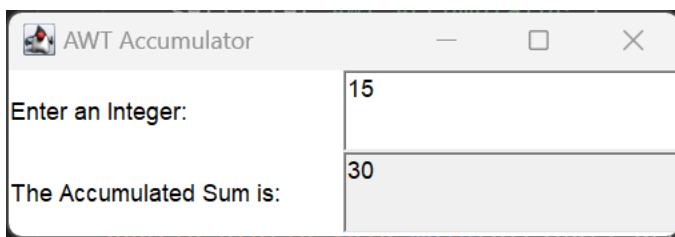


Figure 3 Demo of AWTAccumulator

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     private JTextField tfInput;
10    5 usages
11    private JTextField tfOutput;
12    4 usages
13    private int sum = 0;
14    2 usages
15    1 usage
16    // Constructor
17
18    public SwingAccumulator() {
19        Container cp = getContentPane();
20        cp.setLayout(new GridLayout( rows: 2, cols: 2));
21
22        cp.add(new JLabel( text: "Enter an Integer: "));
23
24        tfInput = new JTextField( columns: 10);
25        cp.add(tfInput);
```

Figure 4 Source code of SwingAccumulator

```

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
40 ⓘ
41
42
43
44
45

```

Figure 5 Source code of SwingAccumulator

1.3. Demo

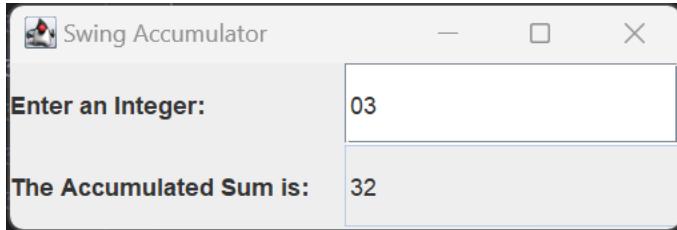


Figure 6 Demo of SwingAccumulator

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

Figure 7 Source code of NumberGrid

```
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 }
```

Figure 8 Source code of NumberGrid

```

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 }

```

Figure 9 Source code of NumberGrid

```

72  ▶  public static void main(String[] args) {
73      new NumberGrid();
74  }
75

```

Figure 10 Source code of NumberGrid

2.2. Demo

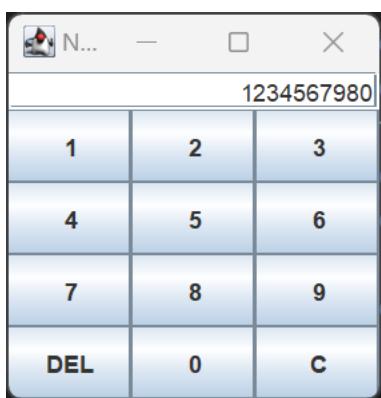


Figure 11 Demo buttons 0-9

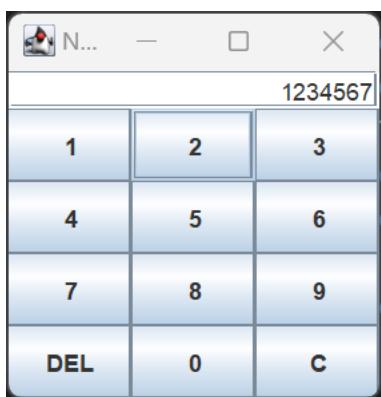


Figure 12 Demo button DEL

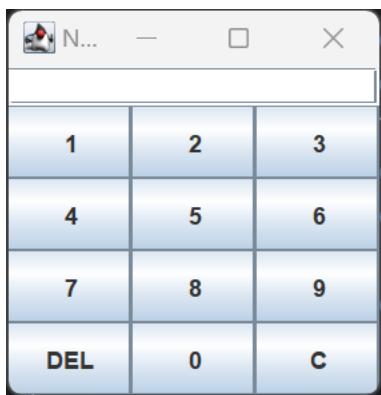


Figure 13 Demo button C

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());
```

Figure 14 Source code of *StoreScreen* Class

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

Figure 15 Source code of *StoreScreen Class*

```
47     smUpdateStore.add(new JMenuItem( text: "Add CD"));
48     smUpdateStore.add(new JMenuItem( text: "Add DVD"));

49     menu.add(smUpdateStore);
50     menu.add(new JMenuItem( text: "View store"));
51     menu.add(new JMenuItem( text: "View cart"));

52
53     JMenuBar menuBar = new JMenuBar();
54     menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
55     menuBar.add(menu);

56
57     return menuBar;
58 }
59
60 // Method to create Header
61
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);
```

Figure 16 Source code of StoreScreen Class

```
71     JButton cart = new JButton( text: "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);
```

Figure 17 Source code of StoreScreen Class

```

96
97         }
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" );

```

Figure 18 Source code of StoreScreen Class

```

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    }

```

Figure 19 Source code of StoreScreen Class

```
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 20 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 public class MediaStore extends JPanel {
12     // Attribute
13     private Media media;
14     private static Cart cart;
15
16     // Constructor
17     @v public MediaStore(Media media) {
18         this.media = media;
19         this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
20
21         JLabel title = new JLabel(media.getTitle());
```

Figure 21 Source code of MediaStore Class

```
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);
```

Figure 22 Source code of MediaStore Class

```
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 }
```

Figure 23 Source code of MediaStore Class

```

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 24 Source code of MediaStore Class

3.3. Demo

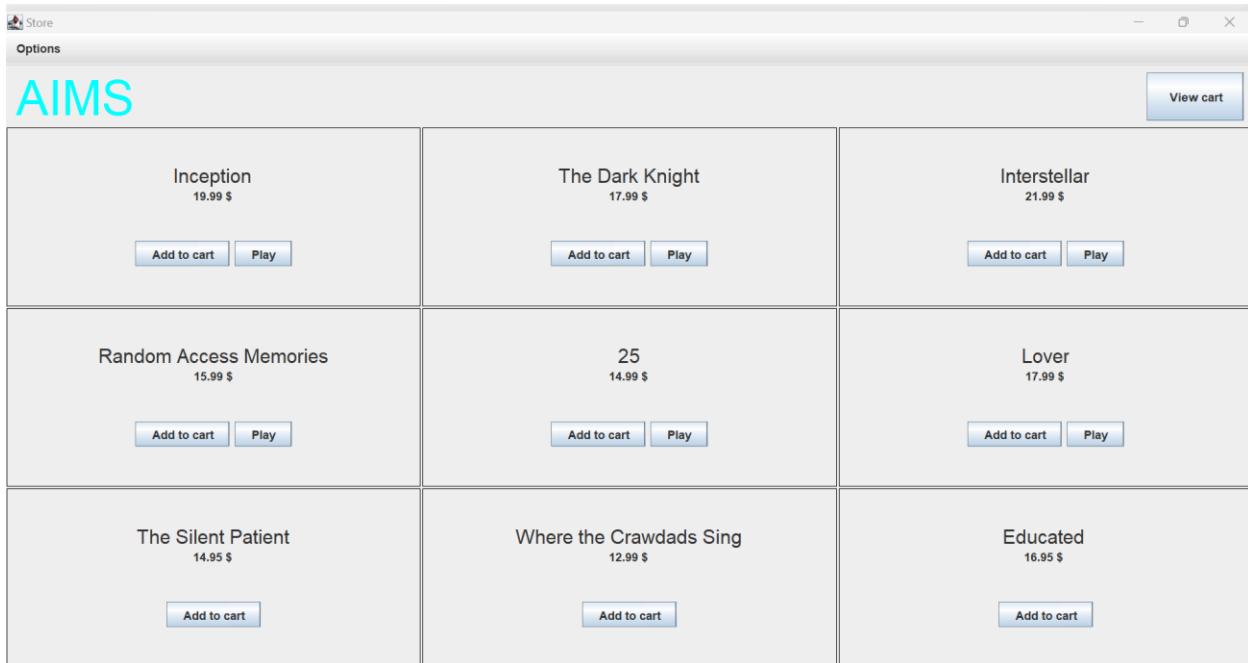


Figure 25 Demo of Store Screen

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

Figure 26 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 27 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 28 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 29 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" />
```

Figure 30 Source code of Painter.fxml

```

23
24         <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
25     </padding>
26     <left>
27         <VBox maxHeight="1.7976931348623157E308"
28             maxWidth="-Infinity" prefWidth="70.0"
29             spacing="8.0" BorderPane.alignment="CENTER">
30             <BorderPane.margin>
31                 <Insets right="8.0" />
32             </BorderPane.margin>
33             <children>
34                 <TitledPane animated="false" text="Tools">
35                     <content>
36                         <AnchorPane>
37                             <children>
38                                 <RadioButton mnemonicParsing="false"
39                                     onAction="#penButtonPressed" selected="true" text="Pen">
40                                     <toggleGroup>
41                                         <ToggleGroup fx:id="identical" />
42                                     </toggleGroup>
43                                     <padding>
44                                         <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
45                                     </padding>
46                                 </RadioButton>
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68

```

Figure 31 Source code of Painter.fxml

```

46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68

```

Figure 32 Source code of Painter.fxml

```

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 33 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
19     @FXML
20     private ToggleGroup identical;
21
22     @FXML
23     void clearButtonPressed(ActionEvent event) {
24         drawingAreaPane.getChildren().clear();
25     }

```

Figure 34 Source code of PainterController Class

```
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 35 Source code of PainterController Class

4.4. Demo

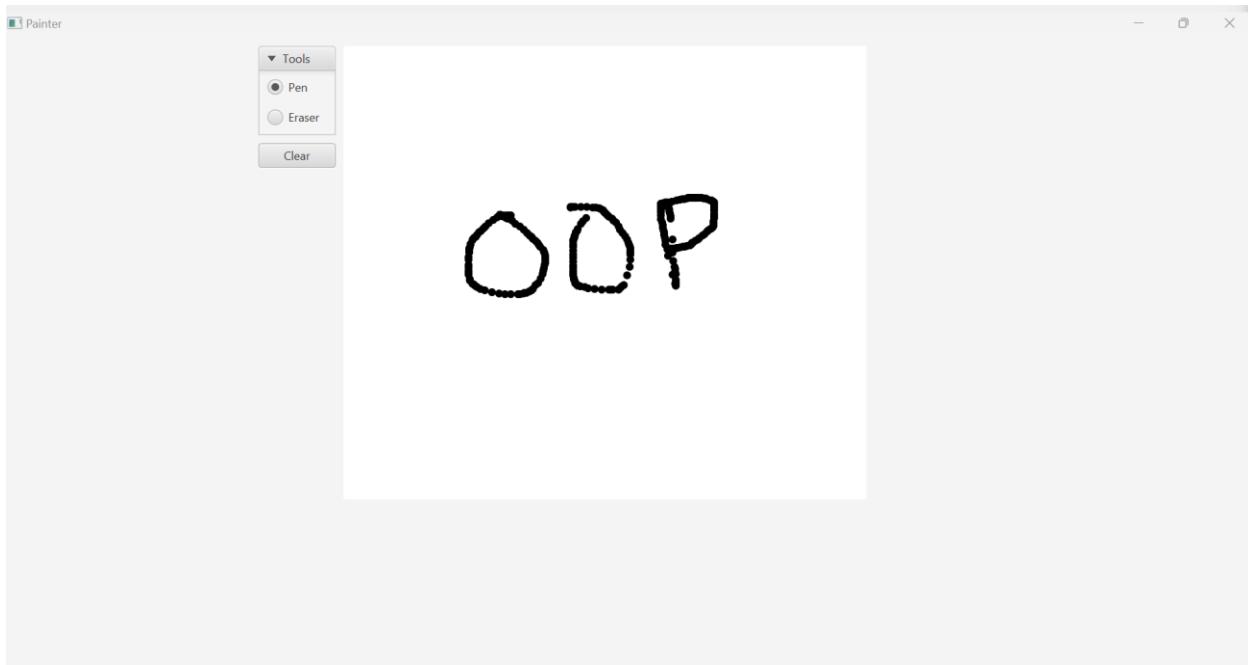


Figure 36 Demo Pen



Figure 37 Demo Eraser

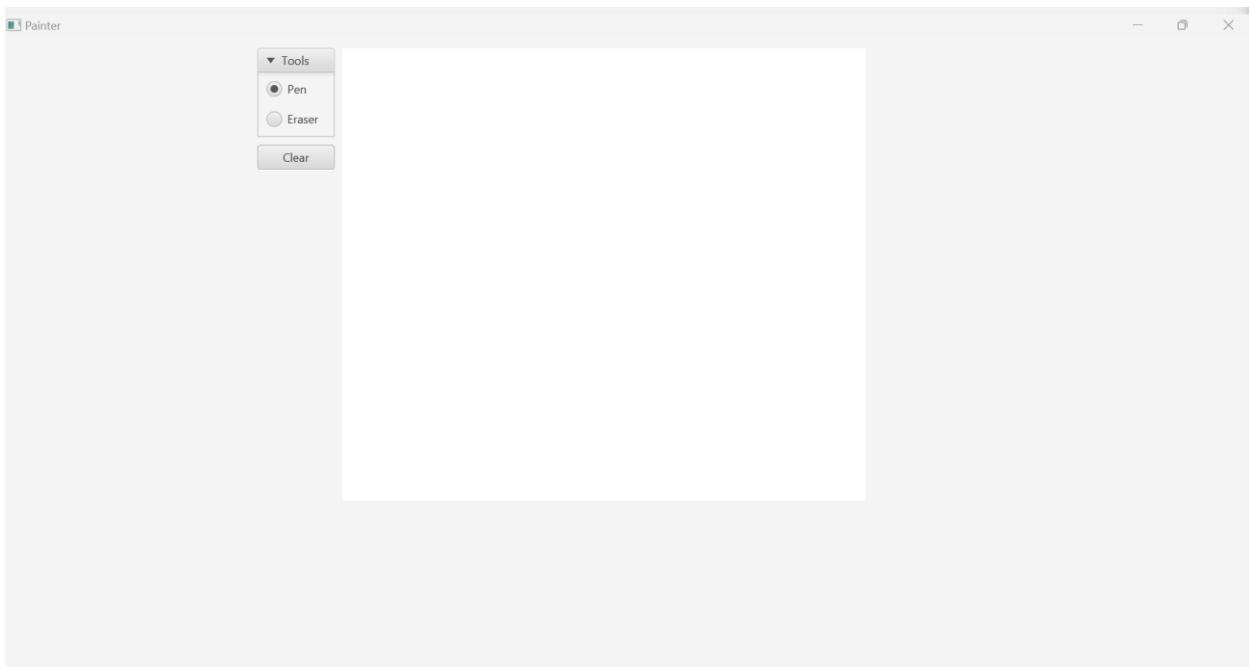


Figure 38 Demo Clear button

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.controlTableView?>
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>
```

Figure 39 Source code of cart.fxml

```

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                 </items>
40               </Menu>
41             </menus>
42           <Label text="CART" textFill="AQUA">
43             <font>
44               <Font size="50.0" />
45             </font>
46           </Label>
47         </children>
48       </VBox>
49     </top>
50     <center>
51       <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
52         <padding>
53           <Insets left="10.0" />
54         </padding>
55       </VBox>
56     </center>
57   </BorderPane>
58 
```

Figure 40 Source code of cart.fxml

```

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     </HBox>
71   </VBox>
72 </center>
73 
```

Figure 41 Source code of cart.fxml

```

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>

```

Figure 42 Source code of cart.fxml

```

93          <Button mnemonicParsing="false" text="Play" />
94          <Button mnemonicParsing="false" text="Remove" />
95      </buttons>
96  </ButtonBar>
97  </children>
98 </VBox>
99 </center>
100 <right>
101     <VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
102         <padding>
103             <Insets top="50.0" />
104         </padding>
105         <children>
106             <HBox alignment="CENTER">
107                 <children>
108                     <Label lineSpacing="10.0" text="Total:>
109                         <font>
110                             <Font size="24.0" />
111                         </font>
112                     </Label>
113                     <Label text="0 $" textFill="AQUA">
114                         <font>
115                             <Font size="24.0" />

```

Figure 43 Source code of cart.fxml

```
116           </font>
117         </Label>
118       </children>
119     </HBox>
120   ■ <Button mnemonicParsing="false" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">
121   ■   <font>
122     <Font size="24.0" />
123   </font>
124   </Button>
125   </children>
126   </VBox>
127 </right>
128 </BorderPane>
129 </children>
130 </AnchorPane>
131
```

Figure 44 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         this.cart = cart;
23
24         JFXPanel fxPanel = new JFXPanel();
```

Figure 45 Source code of CartScreen Class

```
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 46 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
11 public class CartScreenController {
12     ⚡ 2 usages
13     private Cart cart;
14
15     no usages
16     private FXMLLoader loader;
17
18     1 usage
19     @FXML
20     private TableView<Media> tblMedia;
21
22     1 usage
23     @FXML
24     private TableColumn<Media, String> colMediaTitle;
```

Figure 47 Source code of CartScreenController Class

```
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 48 Source code of CartScreenController Class

5.4. Demo

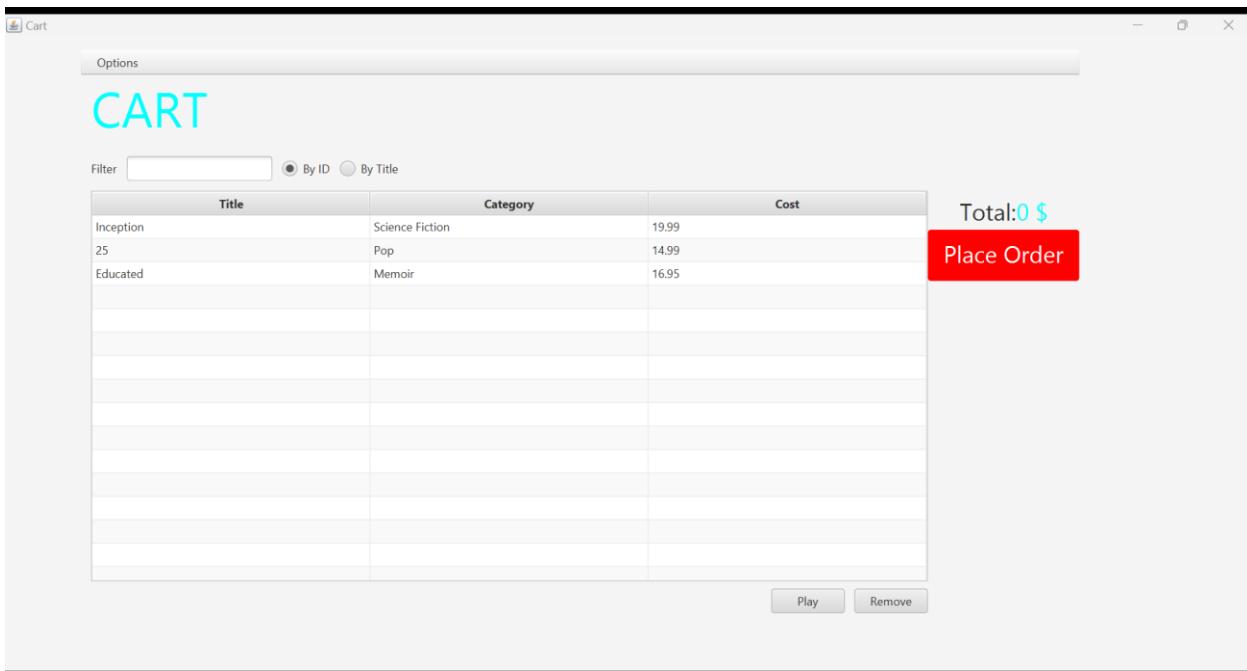


Figure 49 Demo View Cart Screen

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;
```

Figure 50 Source code of CartScreenController Class

```
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();  
    }
```

Figure 51 Source code of CartScreenController

```

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     }

```

Figure 52 Source code of CartScreenController

```

68     void updateButtonBar(Media media){
69         btnRemove.setVisible(true);
70         btnPlay.setVisible(media instanceof Playable);
71     }
72 }

```

Figure 53 Source code of CartScreenController

6.2. Demo

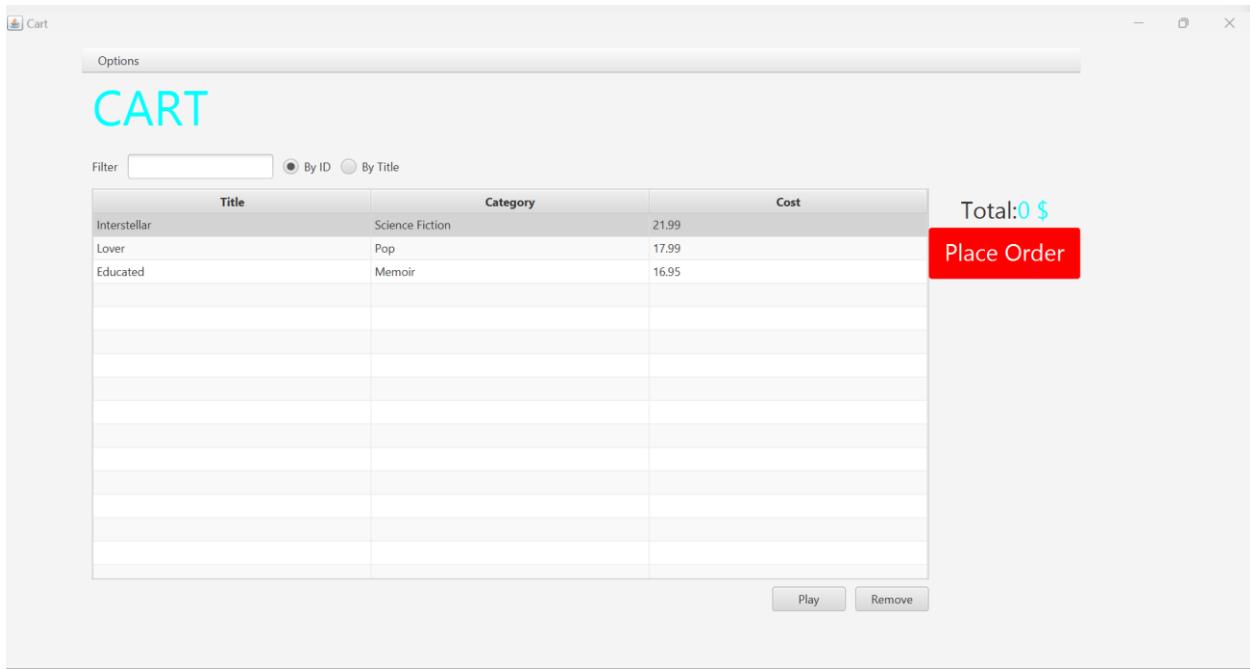


Figure 54 Demo media playable

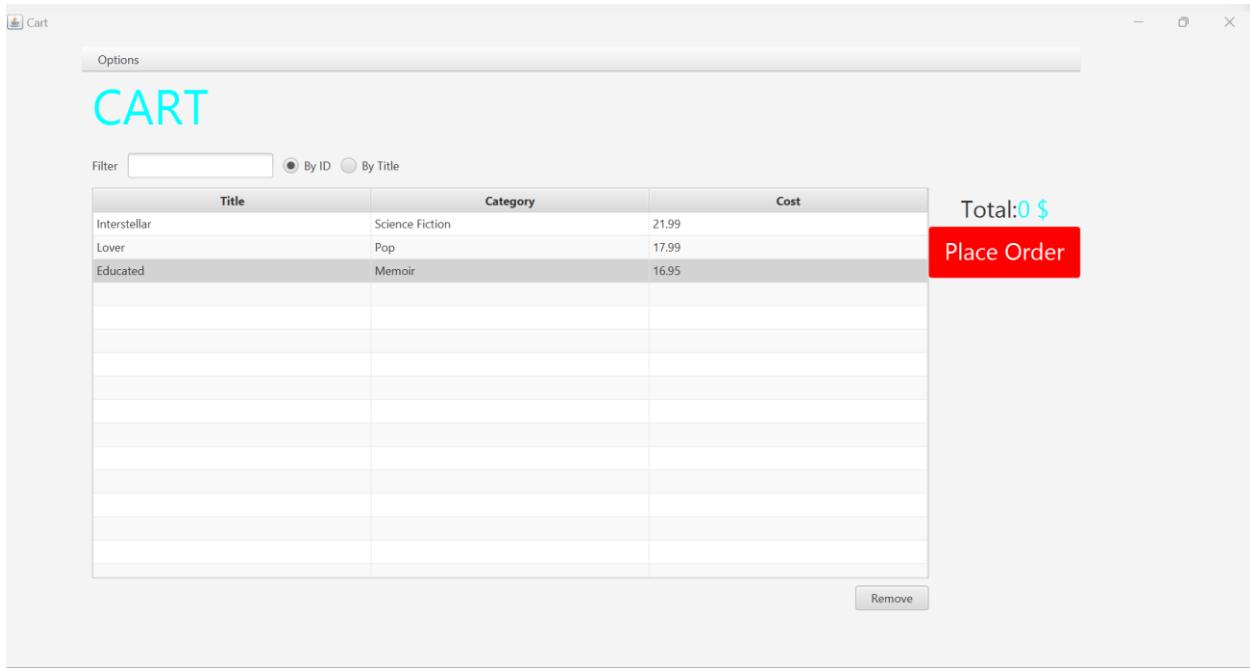


Figure 55 Demo media unplayable

7. Deleting a media

7.1. Code

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

Figure 56 Source code of btnRemovePressed Method

7.2. Demo

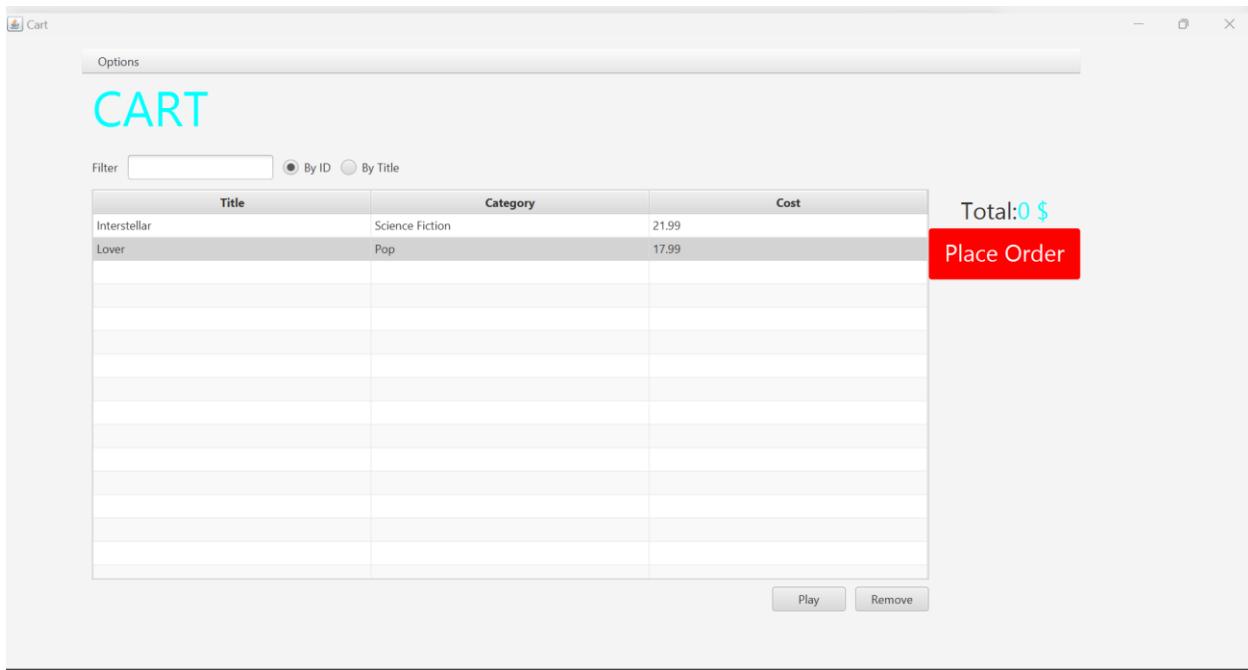


Figure 57 Demo button Remove

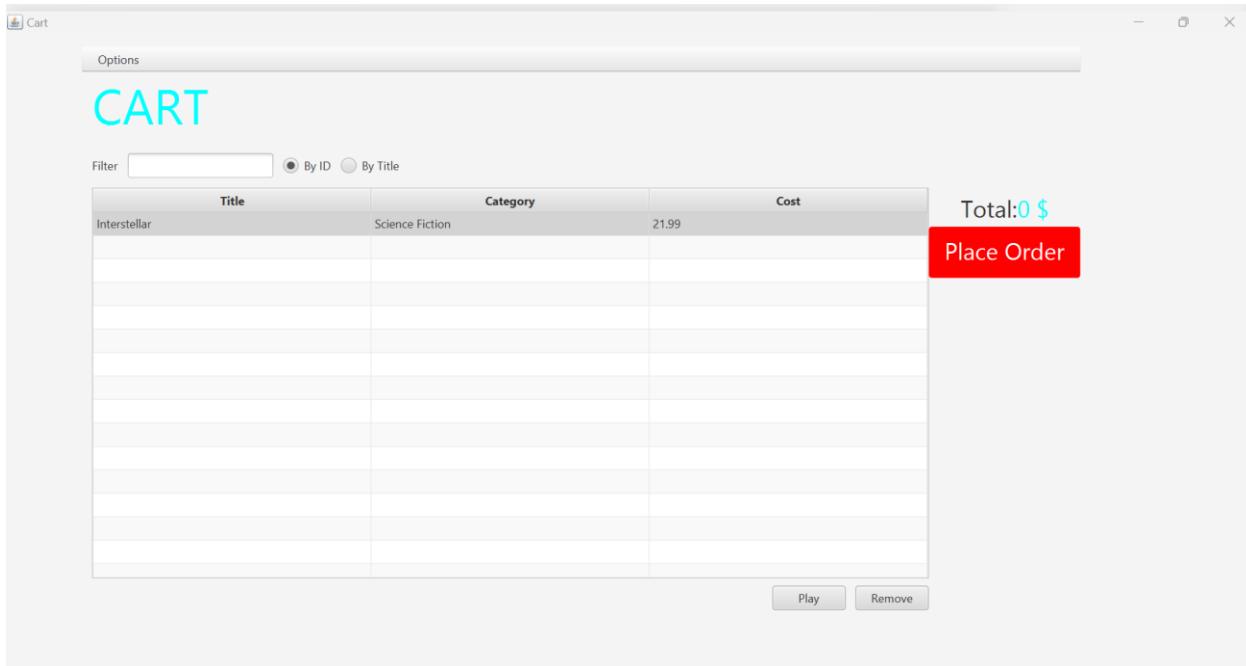


Figure 58 Demo button remove

8. Complete the Aims GUI application

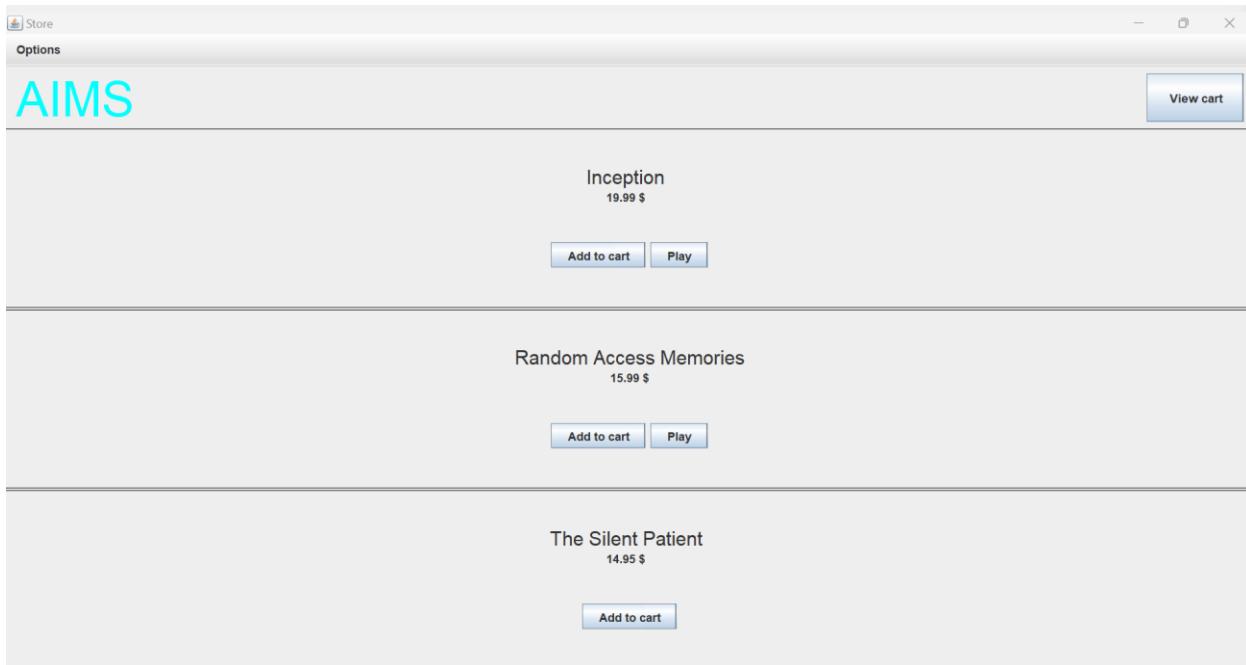


Figure 59 Demo Store before add book

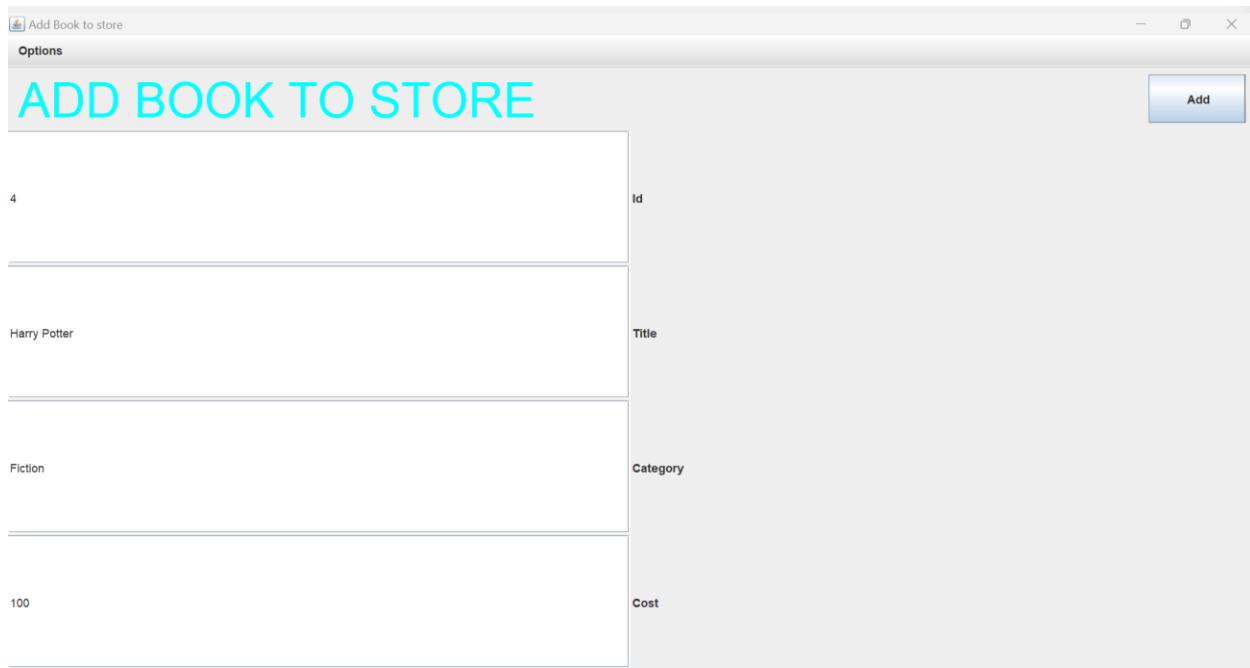


Figure 60 Demo add book

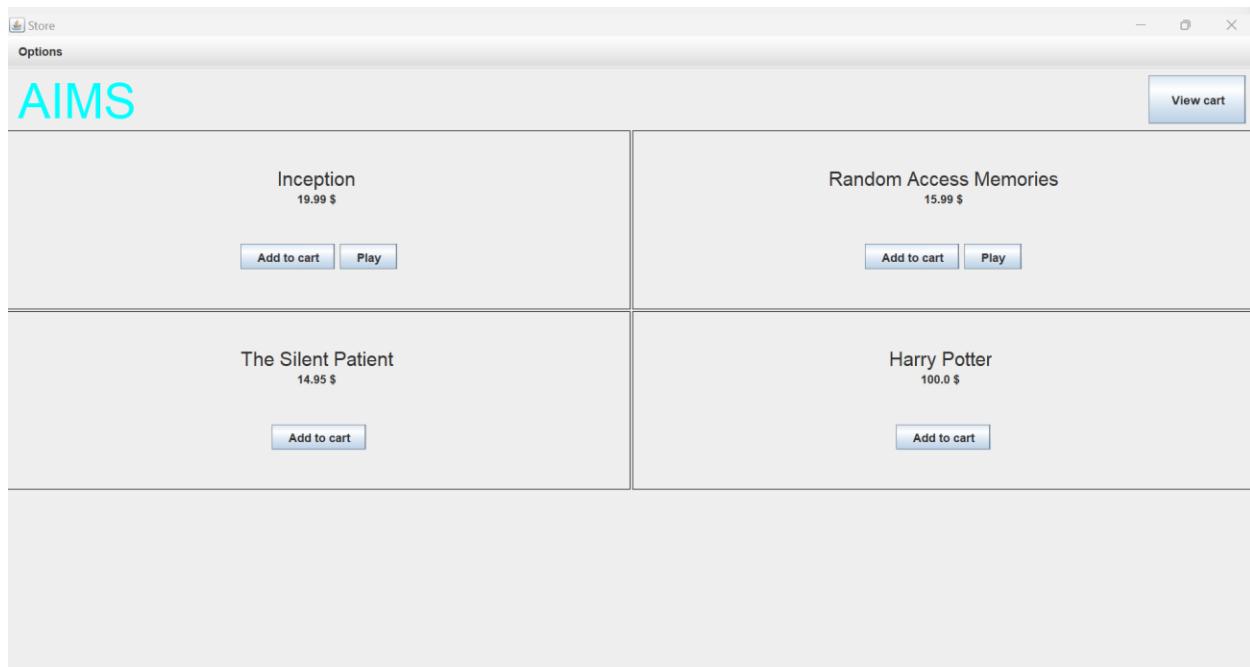


Figure 61 Demo Store after add book

ADD CD TO STORE

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

Figure 62 Demo add CD

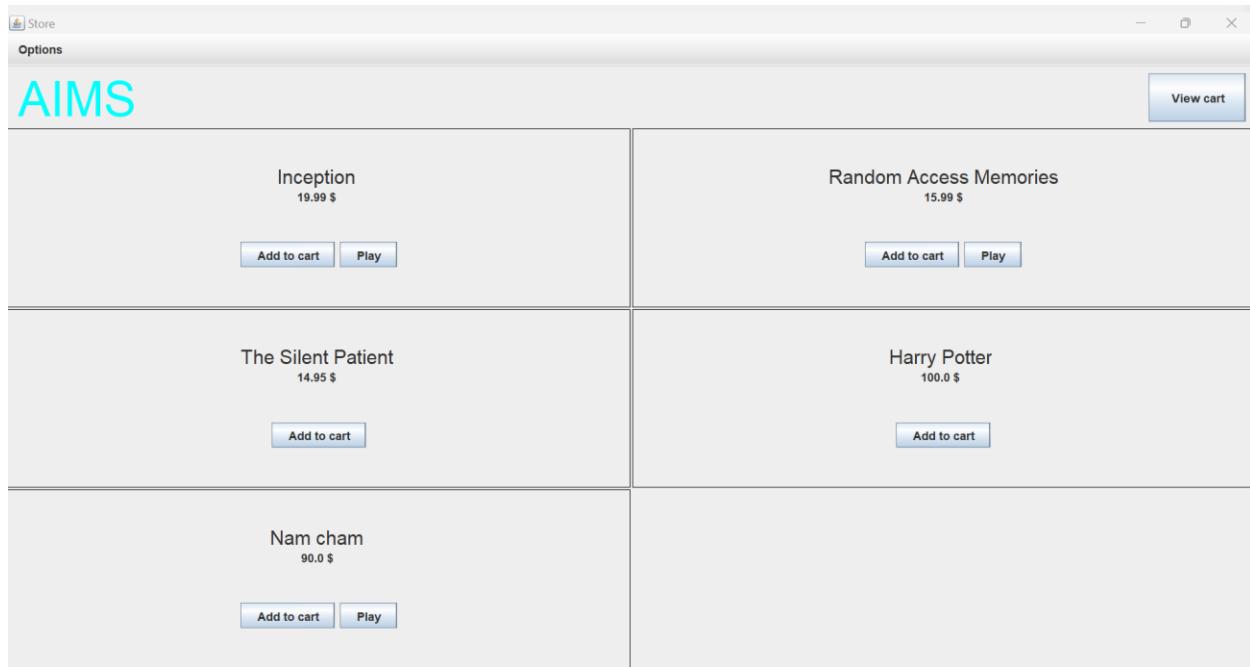


Figure 63 Demo Store after add CD

ADD DVD TO STORE

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

Add

Figure 64 Demo Store add DVD

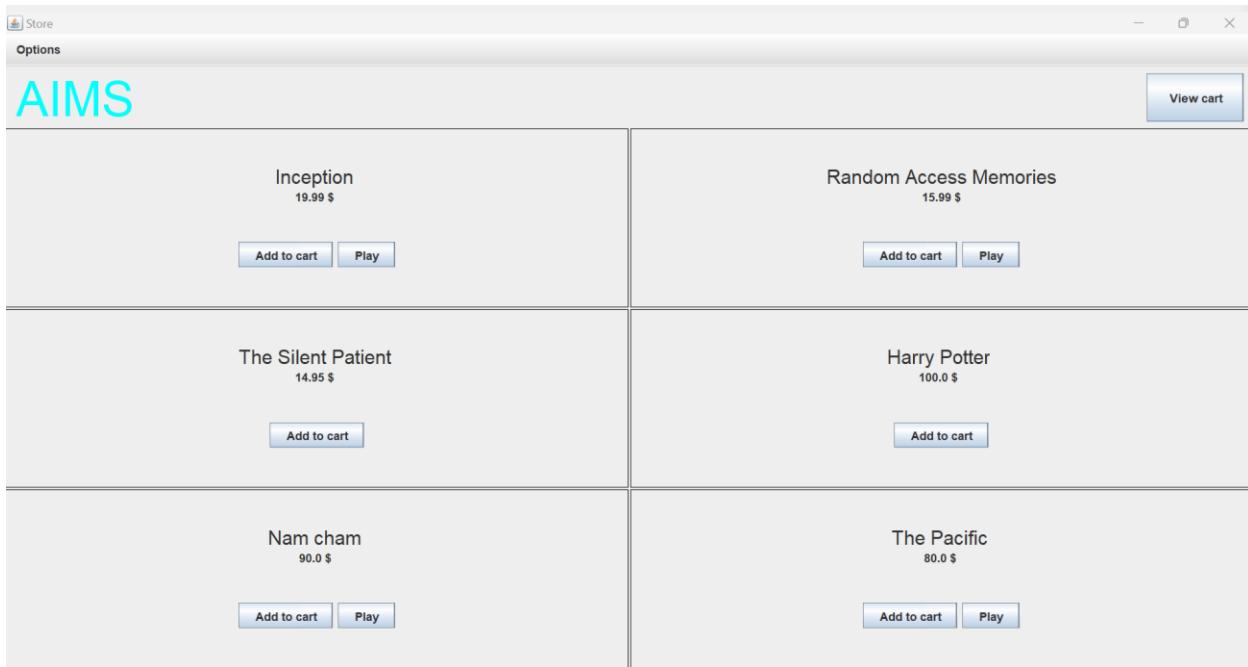


Figure 65 Demo Store after add DVD

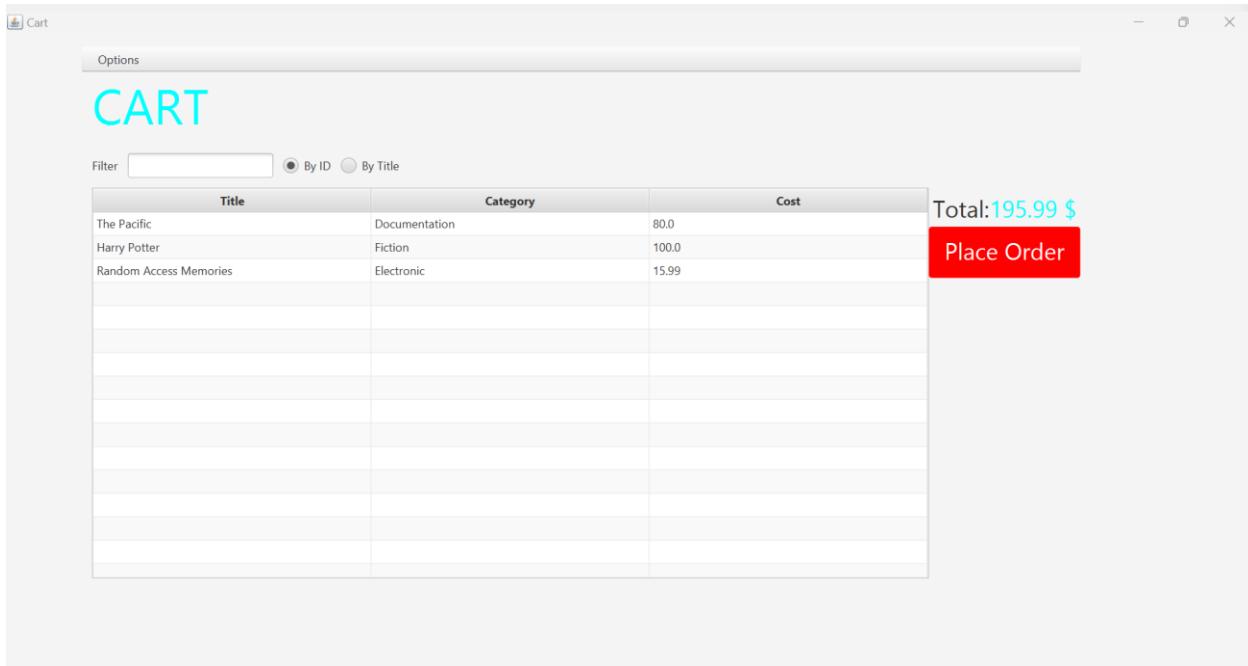


Figure 66 Demo Cart

```
! "AWT-EventQueue-0" java.lang.RuntimeException Create breakpoint : lab05.AimsProject.exception.PlayerException: ERROR: CD length is non-positive!
object.screen.MediaStore$ButtonListener.actionPerformed(MediaStore.java:87) <4 internal lines>
!/java.awt.event.WindowAdapter.windowClosed(WindowAdapter.java:300) <30 internal lines>
AimsProject.exception.PlayerException Create breakpoint : ERROR: CD length is non-positive!
object.media.CompactDisc.play(CompactDisc.java:73)
object.screen.MediaStore$ButtonListener.actionPerformed(MediaStore.java:85)
```

Figure 67 Catch Exception

9. Use case Diagram

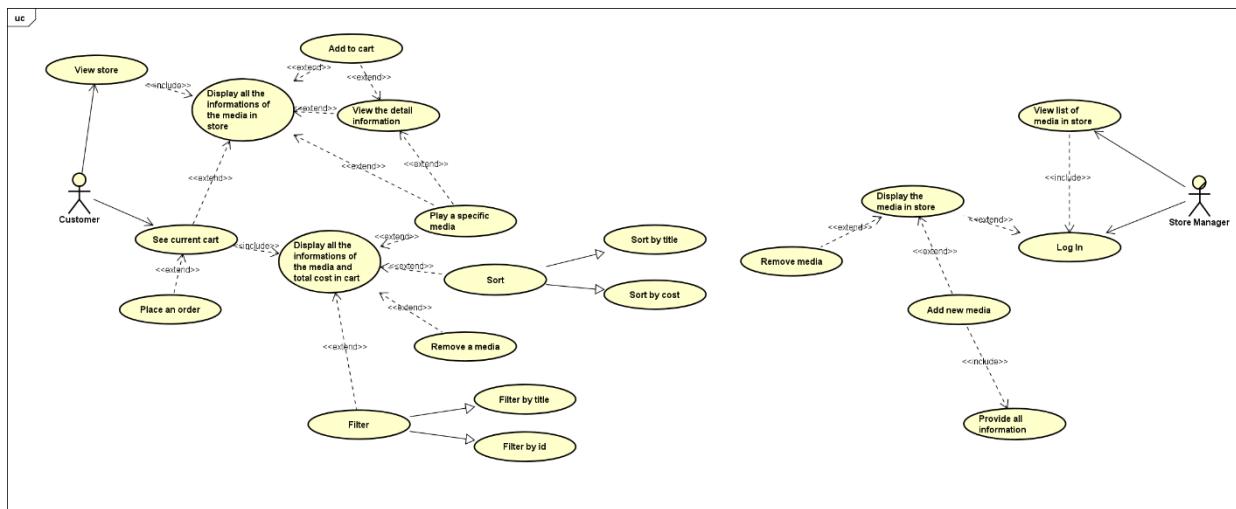


Figure 68 Use case Diagram

10. Class Diagram

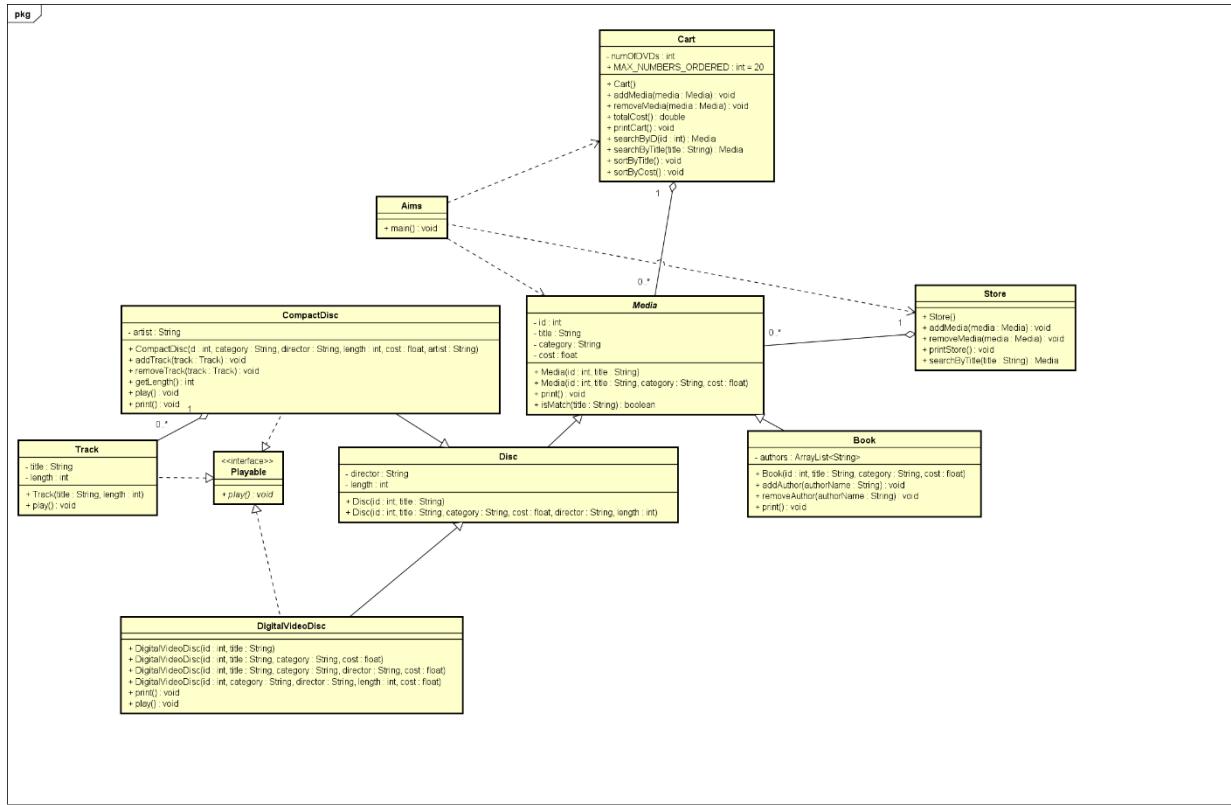


Figure 69 Class Diagram