

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



BÁO CÁO THỰC HÀNH LAB 05
MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mã học phần: IT3103

Mã lớp: 744530

Giảng viên hướng dẫn:

Cô Lê Thị Hoa

Sinh viên thực hiện:

Nguyễn Mạnh Tùng

MSSV:

20225682

Hà Nội, tháng 12 năm 2024

Contents

1.	Swing components	4
1.1	AWTAccumulator	4
1.2	SwingAccumulator	5
2	Organizing Swing components with Layout Managers	6
2.1	Code	6
2.2	Demo	8
3	Create a graphical user interface for AIMS with Swing	9
3.1	Create class StoreScreen	9
3.2	Create class MediaStore	13
3.3	Demo	14
4	JavaFX API	16
4.1	Create class Painter	16
4.2	Create Painter.fxml	16
4.3	Create class PainterController	17
5	View Cart Screen	19
5.1	Create cart.fxml	19
5.2	Create class CartScreen	20
5.3	Create class CartScreenController	21
5.4	Demo	22
6	Updating buttons based on selected item in TableView – ChangeListener	22
6.1	Edit class CartScreenController	22
6.2	Demo	23
7	Deleting a media	24
7.1	Code	24
7.2	Demo	25
8	Complete the Aims GUI application	26
9	Use case Diagram	30
10	Class Diagram	31

Figure 1.1: Source code of AWTAccumulator.....	4
Figure 1.2: Demo of AWTAccumulator.....	5
Figure 1.3: Source code of SwingAccumulator	5
Figure 1.4: Demo of SwingAccumulator	6
Figure 2.1: Source code of NumberGrid 1.....	6
Figure 2.2: Source code of NumberGrid 2.....	7
Figure 2.3: Demo buttons 0-9.....	8
Figure 2.4: Demo DEL button	8
Figure 2.5: Demo C button	8
Figure 3.1: Class StoreScreen 1	9
Figure 3.2: Class StoreScreen 2	10
Figure 3.3: Class StoreScreen 3	10
Figure 3.4: Class StoreScreen 4	11
Figure 3.5: Class StoreScreen 5	11
Figure 3.6: Class StoreScreen 6	12
Figure 3.7: Class MediaStore 1.....	13
Figure 3.8: Class MediaStore 2.....	13
Figure 3.9: Class MediaStore 3.....	14
Figure 3.10: StoreScreen	14
Figure 3.11 Demo Add to cart button	15
Figure 3.12 Demo Play button.....	15
Figure 3.13 Demo View cart button	15
Figure 4.1: Class Painter	16
Figure 4.2: Painter.fxml 1	16
Figure 4.3: Painter.fxml 2	17
Figure 4.4: PainterController	17
Figure 4.5: Use Pen.....	18
Figure 4.6: Use Eraser.....	18
Figure 4.7: Clear button.....	18
Figure 5.1: Cart.fxml 1	19
Figure 5.2: Cart.fxml 2	19
Figure 5.3: Cart.fxml 3.....	20
Figure 5.4: CartScreen class.....	20
Figure 5.5: CartScreenController 1.....	21
Figure 5.6: CartScreenController 2.....	21
Figure 5.7: Demo CartScreen.....	22
Figure 6.1: CartScreenController 1.....	22
Figure 6.2: CartScreenController 2.....	23
Figure 6.3: Demo media playable.....	23
Figure 6.4: Demo media unplayable.....	24
Figure 7.1: btnRemovePressed Method.....	24
Figure 7.2: button Remove.....	25
Figure 7.3: button Remove.....	25
Figure 8.1: Store before add book.....	26

Figure 8.2: Add book	26
Figure 8.3: Store after add book.....	27
Figure 8.4: Add CD.....	27
Figure 8.5: Store after add CD	28
Figure 8.6 Add DVD	28
Figure 8.7: Store after add DVD.....	29
Figure 8.8: Cart	29
Figure 8.9: Exception.....	30

1. Swing components

1.1 AWTAccumulator

```

1  package hust.soict.dsai.aims.javafx;
2
3  import java.awt.*;
4  import java.awt.event.*;
5
6  public class AWTAccumulatorTungNM extends Frame{
7      private TextField tfInput;
8      private TextField tfOutput;
9      private int sum = 0;          // Tổng tích lũy, khởi tạo bằng 0
10
11     // Constructor để thiết lập GUI và các bộ xử lý sự kiện
12     public AWTAccumulatorTungNM() {
13         setLayout(new GridLayout(rows:2, cols:2));
14
15         // Nhận và ô nhập dữ liệu
16         add(new Label(text:"Enter an Integer: "));
17         tfInput = new TextField(columns:10);
18         add(tfInput);
19         tfInput.addActionListener(new TFInputListener());
20
21         add(new Label(text:"The Accumulated Sum is: "));
22         tfOutput = new TextField(columns:10);
23         tfOutput.setEditable(b:false);
24         add(tfOutput);
25
26         setTitle(title:"AWT Accumulator");
27         setSize(width:350, height:120);
28         setVisible(b:true);
29     }
30
31     Run | Debug
32     public static void main(String[] args) {
33         new AWTAccumulatorTungNM();
34     }
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 }

```

Figure 1.1: Source code of AWTAccumulator

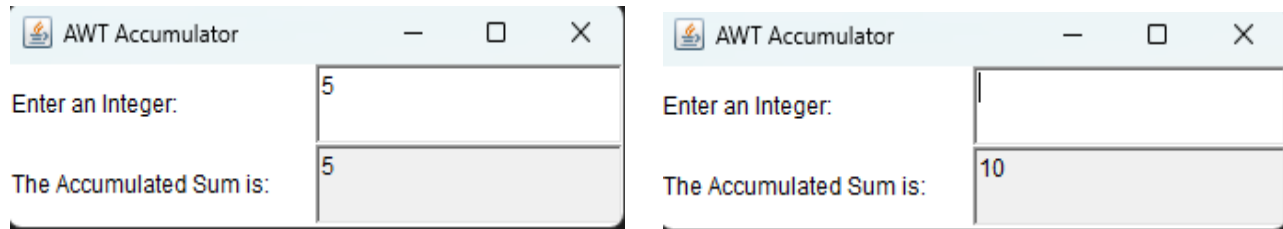


Figure 1.2: Demo of AWT Accumulator

1.2 SwingAccumulator

```

1  package hust.soict.dsai.aims.javafx;
2
3  import javax.swing.*;
4  import java.awt.*;
5  import java.awt.event.*;
6
7  public class SwingAccumulatorTungNM extends JFrame{
8      private JTextField tfInput;
9      private JTextField tfOutput;
10     private int sum = 0;      // Accumulated sum, init to 0
11
12     // Constructor to setup the GUI components and event handlers
13     public SwingAccumulatorTungNM() {
14         Container cp = getContentPane();
15         cp.setLayout(new GridLayout(rows:2, cols:2));
16
17         cp.add(new JLabel(text:"Enter an Integer: "));
18
19         tfInput = new JTextField(columns:10);
20         cp.add(tfInput);
21         tfInput.addActionListener(new TFInputListener());
22
23         cp.add(new JLabel(text:"The Accumulated Sum is: "));
24
25         tfOutput = new JTextField(columns:10);
26         tfOutput.setEditable(b:false);
27         cp.add(tfOutput);
28
29         setTitle(title:"Swing Accumulator");
30         setSize(width:350, height:120);
31         setVisible(b:true);
32     }
33
34     Run | Debug
35     public static void main(String[] args) {
36         new SwingAccumulatorTungNM();
37     }
38
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 1.3: Source code of SwingAccumulator

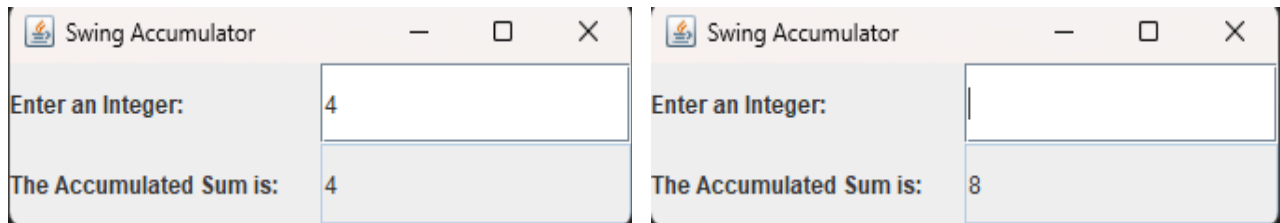


Figure 1.4: Demo of SwingAccumulator

2 Organizing Swing components with Layout Managers

2.1 Code

```

1  package hust.soict.dsai.aims.GUIproject;
2
3  import java.awt.*;
4  import java.awt.event.*;
5  import javax.swing.*;
6
7  public class NumberGridTungNM extends JFrame {
8      private JButton[] btnNumbers = new JButton[10];
9      private JButton btnDelete, btnReset;
10     private JTextField tfDisplay;
11
12     public NumberGridTungNM() {
13         tfDisplay = new JTextField();
14         tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
15         tfDisplay.setEditable(b:false);
16         tfDisplay.setFont(new Font(name:"Arial", Font.BOLD, size:20));
17
18         JPanel panelButtons = new JPanel(new GridLayout(rows:4, cols:3, hgap:5, vgap:5));
19         addButtons(panelButtons);
20
21         Container cp = getContentPane();
22         cp.setLayout(new BorderLayout(hgap:5, vgap:5));
23         cp.add(tfDisplay, BorderLayout.NORTH);
24         cp.add(panelButtons, BorderLayout.CENTER);
25
26         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27         setTitle(title:"Number Grid - Nguyen Manh Tung 20225682");
28         setSize(width:300, height:400);
29         setLocationRelativeTo(c:null);
30         setVisible(b:true);
31     }
32

```

Figure 2.1: Source code of NumberGrid 1

```

Run | Debug
33 public static void main(String[] args) {
34     new NumberGridTungNM();
35 }
36
37 void addButtons(JPanel panelButtons) {
38     ButtonListener btnListener = new ButtonListener();
39
40     for (int i = 1; i <= 9; i++) {
41         btnNumbers[i] = new JButton("" + i);
42         btnNumbers[i].setFont(new Font(name:"Arial", Font.BOLD, size:20));
43         panelButtons.add(btnNumbers[i]);
44         btnNumbers[i].addActionListener(btnListener);
45     }
46
47     btnDelete = new JButton(text:"DEL");
48     btnDelete.setFont(new Font(name:"Arial", Font.BOLD, size:20));
49     panelButtons.add(btnDelete);
50     btnDelete.addActionListener(btnListener);
51
52     btnNumbers[0] = new JButton(text:"0");
53     btnNumbers[0].setFont(new Font(name:"Arial", Font.BOLD, size:20));
54     panelButtons.add(btnNumbers[0]);
55     btnNumbers[0].addActionListener(btnListener);
56
57     btnReset = new JButton(text:"C");
58     btnReset.setFont(new Font(name:"Arial", Font.BOLD, size:20));
59     panelButtons.add(btnReset);
60     btnReset.addActionListener(btnListener);
61 }
62
63 private class ButtonListener implements ActionListener {
64     @Override
65     public void actionPerformed(ActionEvent e) {
66         String button = e.getActionCommand();
67         if (button.charAt(index:0) >= '0' && button.charAt(index:0) <= '9') {
68             tfDisplay.setText(tfDisplay.getText() + button);
69         } else if (button.equals(anObject:"DEL")) {
70             String displayStr = tfDisplay.getText();
71             if (displayStr.length() > 0) {
72                 tfDisplay.setText(displayStr.substring(beginIndex:0, displayStr.length() - 1));
73             }
74         } else if (button.equals(anObject:"C")) {
75             tfDisplay.setText(t:"");
76         }
77     }
78 }
79 }
80

```

Figure 2.2: Source code of NumberGrid 2

2.2 Demo

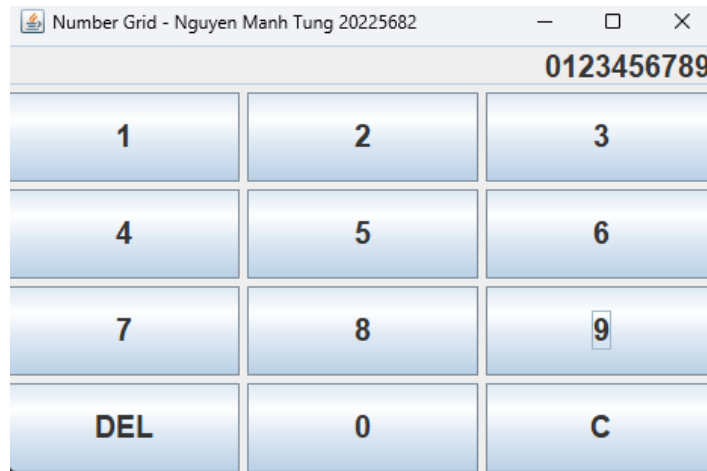


Figure 2.3: Demo buttons 0-9



Figure 2.4: Demo DEL button



Figure 2.5: Demo C button

3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

```

1  package hust.soict.dsai.aims.screen;
2
3  import java.awt.*;
4  import java.util.ArrayList;
5  import javax.swing.*;
6
7  import hust.soict.dsai.aims.media.MediaTungNM;
8  import hust.soict.dsai.aims.media.StoreTungNM;
9
10 public class StoreScreenTungNM extends JFrame {
11     private StoreTungNM store;
12
13     public StoreScreenTungNM(StoreTungNM store) {
14         this.store = store;
15
16         // Cài đặt Container chính của JFrame
17         Container cp = getContentPane();
18         cp.setLayout(new BorderLayout());
19         cp.add(createNorth(), BorderLayout.NORTH);
20         cp.add(createCenter(), BorderLayout.CENTER);
21
22         // Cài đặt các thuộc tính JFrame
23         setTitle(title:"Store Nguyen Manh Tung 20225682");
24         setSize(width:1024, height:768);
25         setVisible(b:true);
26         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27     }
28
29     // Tạo phần phía trên (MenuBar + Header)
30     JPanel createNorth() {
31         JPanel north = new JPanel();
32         north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
33         north.add(createMenuBar());
34         north.add(createHeader());
35         return north;
36     }

```

Figure 3.1: Class StoreScreen 1

```

38     // Tạo thanh menu
39     JMenuBar createMenuBar() {
40         JMenu menu = new JMenu(s:"Options");
41
42         JMenu smUpdateStore = new JMenu(s:"Update Store");
43         smUpdateStore.add(new JMenuItem(text:"Add Book"));
44         smUpdateStore.add(new JMenuItem(text:"Add CD"));
45         smUpdateStore.add(new JMenuItem(text:"Add DVD"));
46
47         menu.add(smUpdateStore);
48         menu.add(new JMenuItem(text:"View Store"));
49         menu.add(new JMenuItem(text:"View Cart"));
50
51         JMenuBar menuBar = new JMenuBar();
52         menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
53         menuBar.add(menu);
54
55         return menuBar;
56     }

```

Figure 3.2: Class StoreScreen 2

```

58     // Tạo tiêu đề và nút "View Cart" ở góc trên bên phải
59     JPanel createHeader() {
60         JPanel header = new JPanel();
61         header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));
62
63         // Tiêu đề của ứng dụng
64         JLabel title = new JLabel(text:"AIMS - Nguyen Manh Tung - 20225682");
65         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:50));
66         title.setForeground(Color.BLUE);
67
68         // Nút "View Cart"
69         JButton cartButton = new JButton(text:"View Cart");
70         cartButton.setPreferredSize(new Dimension(width:120, height:50));
71         cartButton.setMaximumSize(new Dimension(width:120, height:50));
72
73         // Xử lý sự kiện khi nhấn nút "View Cart"
74         cartButton.addActionListener(e -> {
75             JOptionPane.showMessageDialog(
76                 this,
77                 message:"Viewing cart...",
78                 title:"Cart",
79                 JOptionPane.INFORMATION_MESSAGE
80             );
81         });

```

Figure 3.3: Class StoreScreen 3

```

83         // Thêm tiêu đề và nút "View Cart" vào header
84         header.add(Box.createRigidArea(new Dimension(width:10, height:0)));
85         header.add(title);
86         header.add(Box.createHorizontalGlue());
87         header.add(cartButton);
88         header.add(Box.createRigidArea(new Dimension(width:10, height:0)));
89
90         return header;
91     }
92
93     JPanel createCenter() {
94         JPanel center = new JPanel();
95         center.setLayout(new GridLayout(rows:3, cols:3, hgap:2, vgap:2));
96
97         ArrayList<MediaTungNM> mediaInStore = store.getItemsInStore();
98         for (MediaTungNM media : mediaInStore) {
99             MediaStoreTungNM cell = new MediaStoreTungNM(media);
100             center.add(cell);
101         }
102
103         return center;
104     }
105 }
106

```

Figure 3.4: Class StoreScreen 4

```

1  package hust.soict.dsai.aims.test;
2
3  import javax.swing.*;
4  import hust.soict.dsai.aims.media.*;
5  import hust.soict.dsai.aims.screen.*;
6
7  public class TestStoreScreen {
8      Run | Debug
9      public static void main(String[] args) {
10         StoreTungNM store = new StoreTungNM();
11         CartTungNM cart = new CartTungNM();
12
13         MediaTungNM m1 = new DVDTungNM(title:"Fujio", category:"Doraemon", director:"Anime",length:15, price:15f);
14         MediaTungNM m2 = new BookTungNM(id:0, title:"Why we sleep", category:"Scientific", price:9f);
15         MediaTungNM m3 = new CompactDiscTungNM(id:1, title:"Gao ranger", category:"Super sentai", price:20f, length:2, director:"TungNM");
16         MediaTungNM m4 = new DiscTungNM(id:1, title:"Conan", category:"Anime", price:50f, length:10, director:"Khanh");
17
18         store.addMediaTungNM(m1);
19         store.addMediaTungNM(m2);
20         store.addMediaTungNM(m3);
21         store.addMediaTungNM(m4);
22
23         cart.addMediaTungNM(m1);
24         cart.addMediaTungNM(m2);
25         cart.addMediaTungNM(m3);
26         cart.addMediaTungNM(m4);
27
28         StoreScreenTungNM storeScreen = new StoreScreenTungNM(store);
29
30         storeScreen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31     }
32 }

```

Figure 3.5: Class TestStoreScreen 5

3.2 Create class MediaStore

```

1  package hust.soict.dsai.aims.screen;
2
3  import javax.swing.*;
4  import hust.soict.dsai.aims.media.MediaTungNM;
5  import hust.soict.dsai.aims.media.PlayableTungNM;
6  import java.awt.*;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9
10 public class MediaStoreTungNM extends JPanel {
11     private MediaTungNM media;
12
13     public MediaStoreTungNM(MediaTungNM media) {
14         this.media = media;
15         this.setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));
16
17         JLabel title = new JLabel(media.getTitle());
18         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:20));
19         title.setAlignmentX(CENTER_ALIGNMENT);
20
21         JLabel cost = new JLabel("" + media.getPrice() + " $");
22         cost.setAlignmentX(CENTER_ALIGNMENT);
23
24         JPanel container = new JPanel();
25         container.setLayout(new FlowLayout(FlowLayout.CENTER));

```

Figure 3.7: Class MediaStore 1

```

27     // Add "Add to Cart" Button
28     JButton addToCartButton = new JButton(text:"Add to cart");
29     addToCartButton.addActionListener(new ActionListener() {
30         @Override
31         public void actionPerformed(ActionEvent e) {
32             // Show success message
33             JOptionPane.showMessageDialog(
34                 MediaStoreTungNM.this,
35                 "Added " + media.getTitle() + " to the cart successfully!",
36                 title:"Add to Cart",
37                 JOptionPane.INFORMATION_MESSAGE
38             );
39         }
40     });
41     container.add(addToCartButton);

```

Figure 3.8: Class MediaStore 2

```

43 // Add "Play" Button if the media is playable
44 if (media instanceof PlayableTungNM) {
45     JButton playButton = new JButton(text:"Play");
46     playButton.addActionListener(new ActionListener() {
47         @Override
48         public void actionPerformed(ActionEvent e) {
49             // Show play message
50             JOptionPane.showMessageDialog(
51                 MediaStoreTungNM.this,
52                 "Playing " + media.getTitle(),
53                 title:"Play Media",
54                 JOptionPane.INFORMATION_MESSAGE
55             );
56         }
57     });
58     container.add(playButton);
59 }
60 this.add(Box.createVerticalGlue());
61 this.add(title);
62 this.add(cost);
63 this.add(Box.createVerticalGlue());
64 this.add(container);
65
66 this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
67 }
68 }
69

```

Figure 3.9: Class MediaStore 3

3.3 Demo



Figure 3.10: StoreScreen

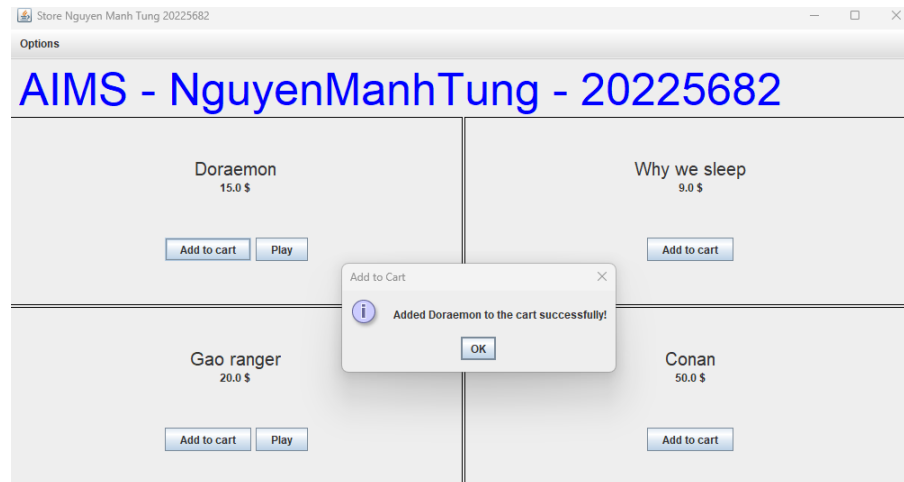


Figure 3.11 Demo Add to cart button

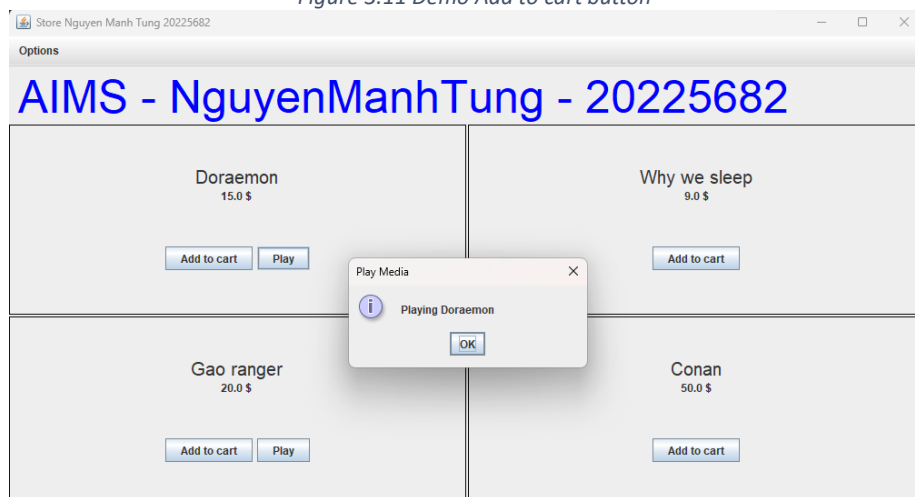


Figure 3.12 Demo Play button

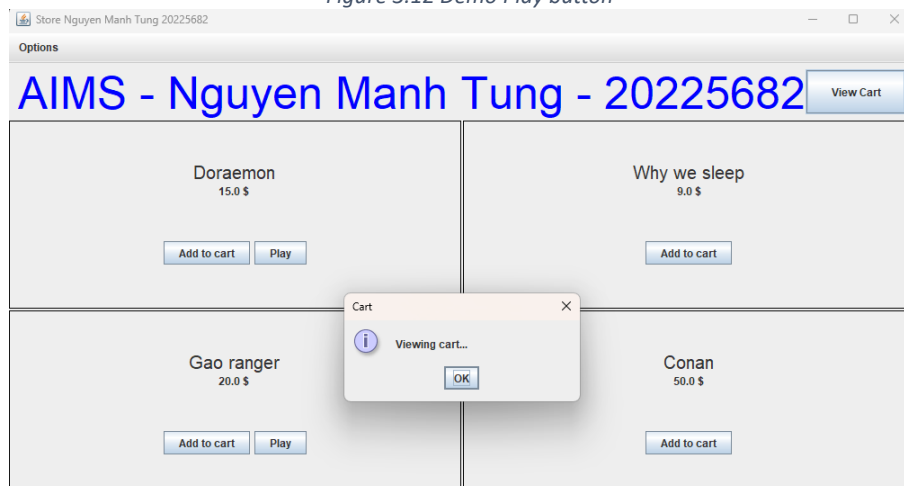


Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

```
1 package hust.soict.dsai.aims.javafx;  
2  
3 import javafx.application.Application;  
4 import javafx.fxml.FXML;  
5 import javafx.fxml.FXMLLoader;  
6 import javafx.scene.Parent;  
7 import javafx.scene.Scene;  
8 import javafx.stage.Stage;  
9  
10 public class PainterTungNM extends Application{  
11     @Override  
12     public void start(Stage stage) throws Exception {  
13         Parent root = FXMLLoader.load(getClass().getResource("/hust/soict/dsai/aims/javafx/Painter.fxml"));  
14  
15         Scene scene = new Scene(root);  
16         stage.setTitle("Painter");  
17         stage.setScene(scene);  
18         stage.show();  
19     }  
20  
21     Run | Debug  
22     public static void main(String[] args) {  
23         launch(args);  
24     }  
25 }  
26
```

Figure 4.1: Class Painter

4.2 Create Painter.fxml

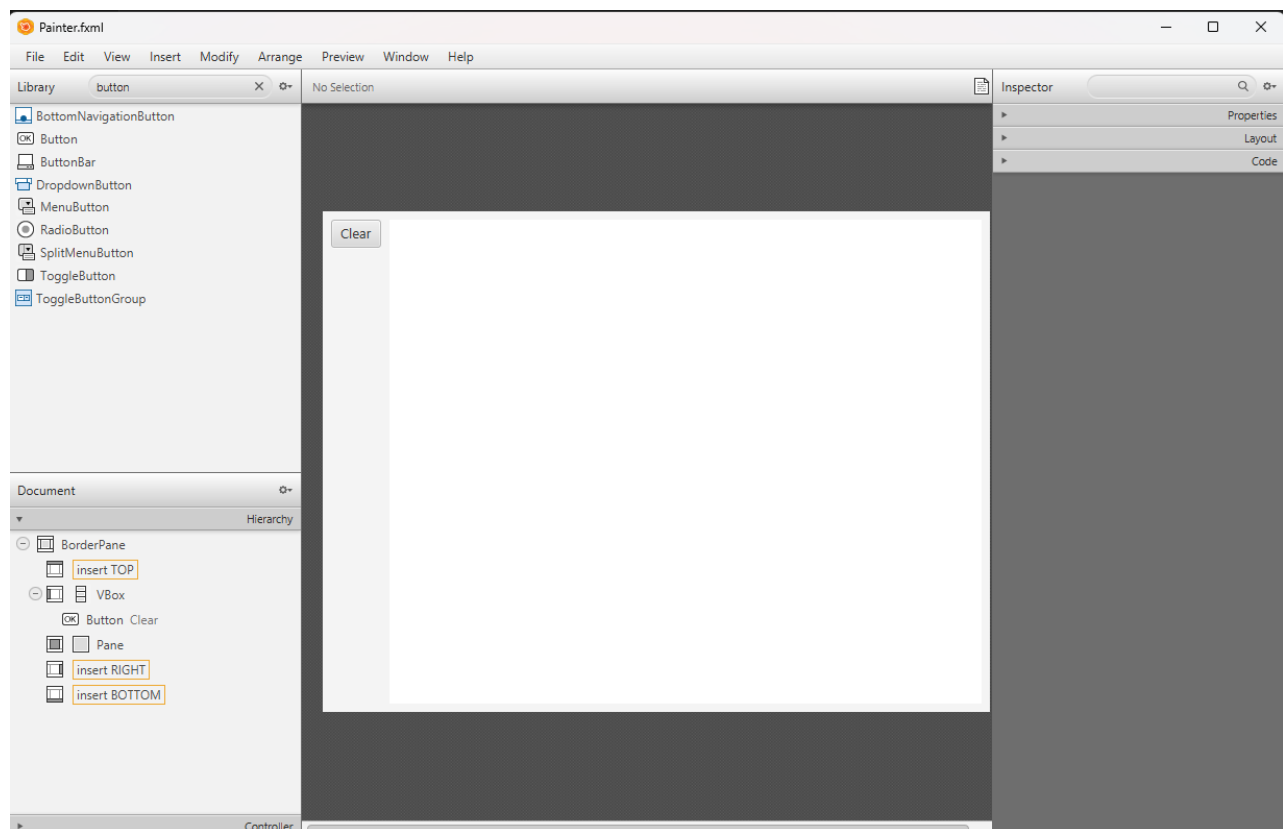


Figure 4.2: Painter.fxml 1

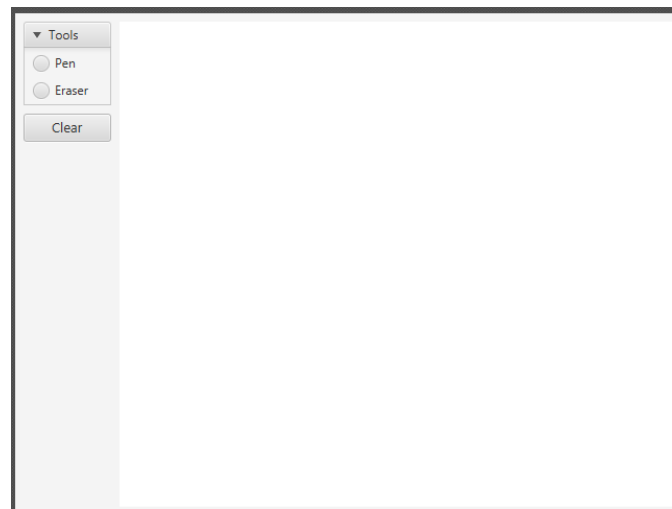


Figure 4.3: Painter.fxml 2

4.3 Create class PainterController

```
1  package hust.soict.dsai.aims.javafx;
2
3  import javafx.event.ActionEvent;
4  import javafx.fxml.FXML;
5  import javafx.scene.control.RadioButton;
6  import javafx.scene.input.MouseEvent;
7  import javafx.scene.layout.Pane;
8  import javafx.scene.paint.Color;
9  import javafx.scene.shape.Circle;
10
11  public class PainterControllerTungNM {
12
13      @FXML
14      private Pane drawingAreaPane;
15
16      @FXML
17      private RadioButton eraser;
18
19      @FXML
20      private RadioButton pen;
21
22      private Color currentColor = Color.BLACK;
23
24      @FXML
25      void toolSelection(ActionEvent event) {
26          if (pen.isSelected()) {
27              currentColor = Color.BLACK;
28          } else if (eraser.isSelected()) {
29              currentColor = Color.WHITE;
30          }
31      }
32
33      @FXML
34      void clearButtonPressed(ActionEvent event) {
35          drawingAreaPane.getChildren().clear();
36      }
37
38      @FXML
39      void initialize() {
40          drawingAreaPane.setOnMouseDragged(this::drawingAreaMouseDragged);
41      }
42
43      void drawingAreaMouseDragged(MouseEvent event) {
44          Circle newCircle = new Circle(event.getX(), event.getY(), 4, currentColor);
45          drawingAreaPane.getChildren().add(newCircle);
46      }
47  }
48
```

Figure 4.4: PainterController

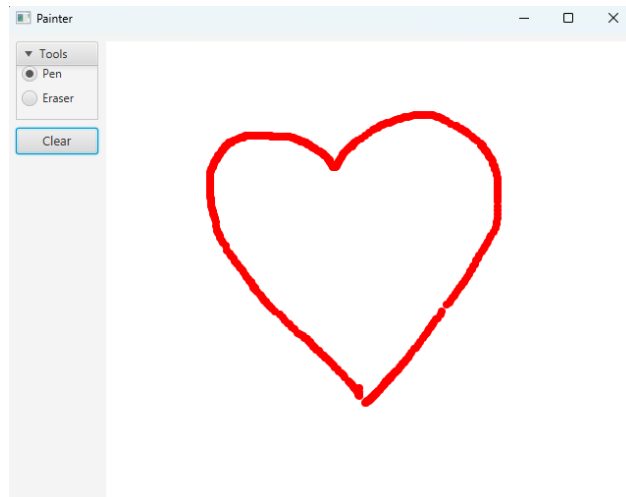


Figure 4.5: Use Pen

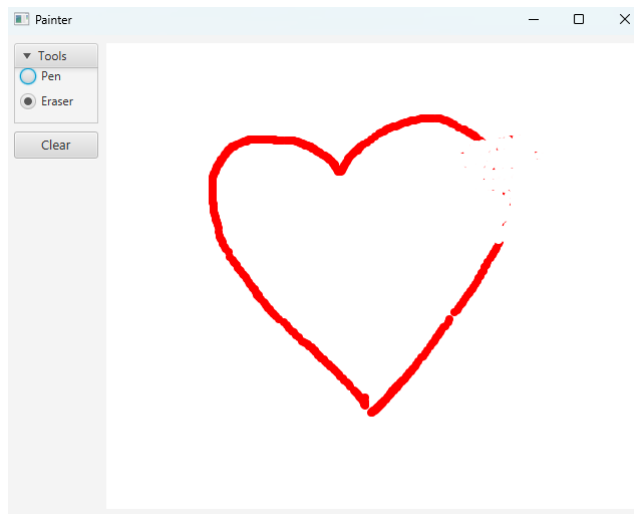


Figure 4.6: Use Eraser

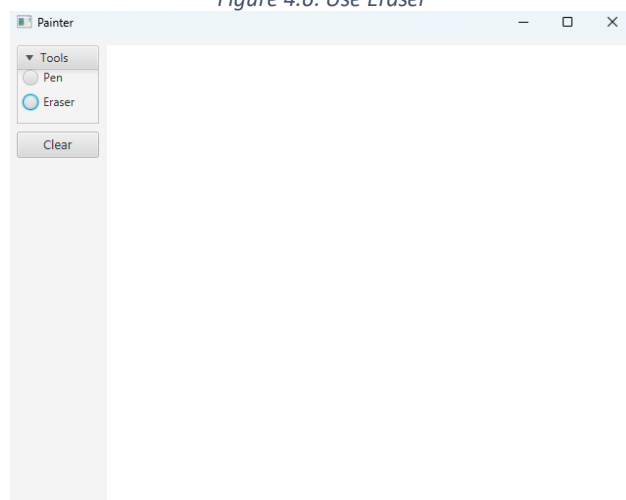


Figure 4.7: Clear button

5 View Cart Screen

5.1 Create cart.fxml

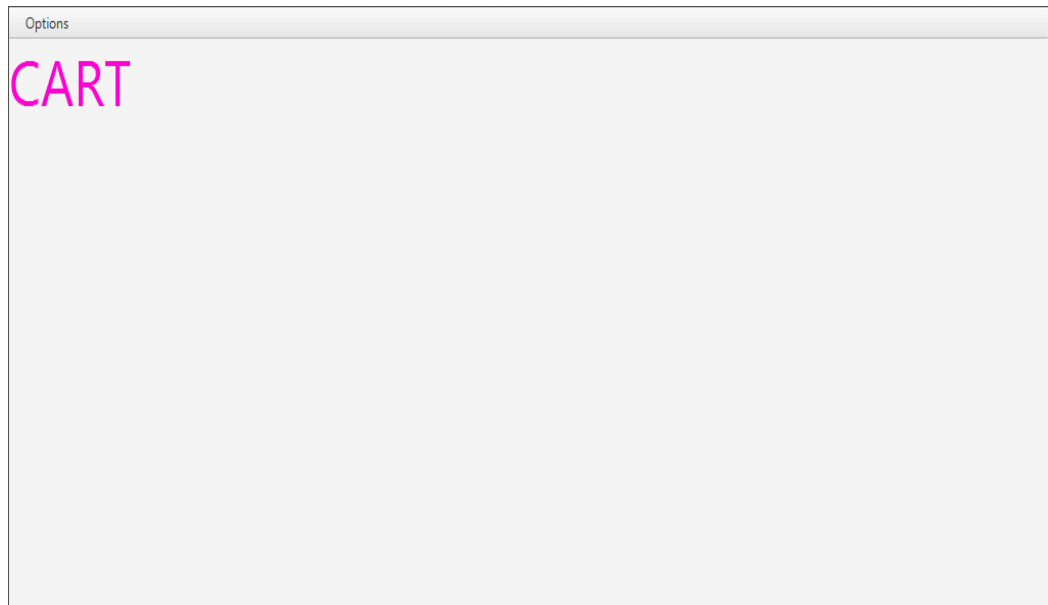


Figure 5.1: Cart.fxml 1

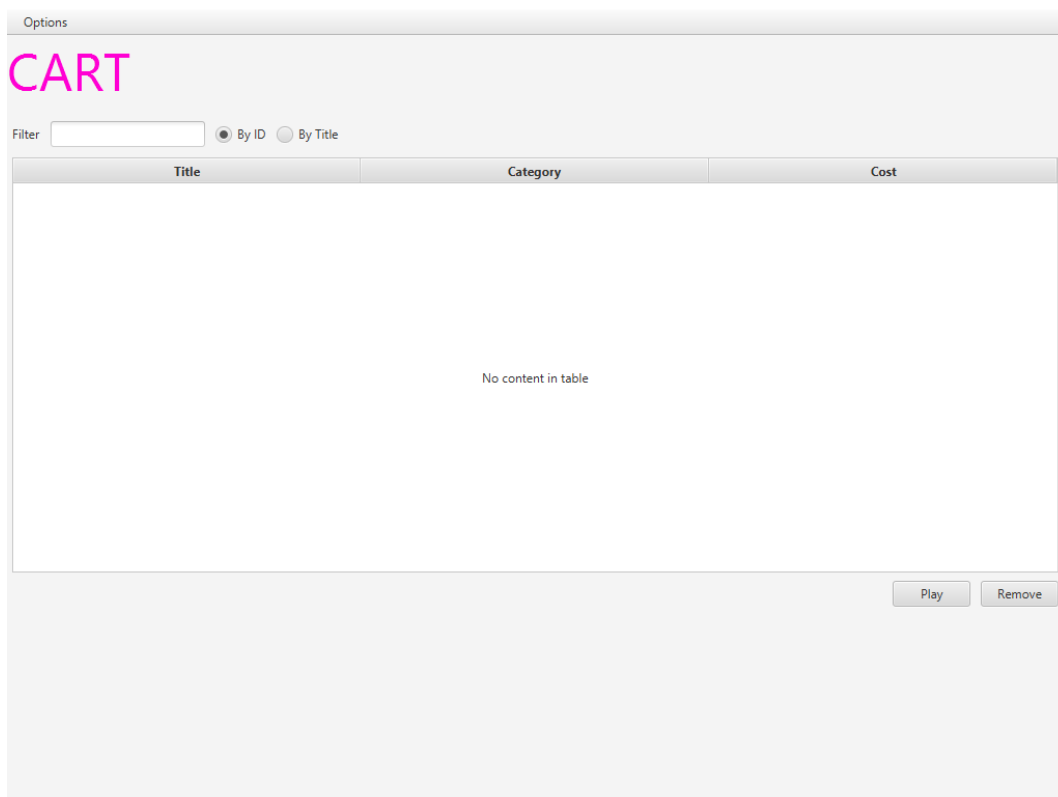


Figure 5.2: Cart.fxml 2

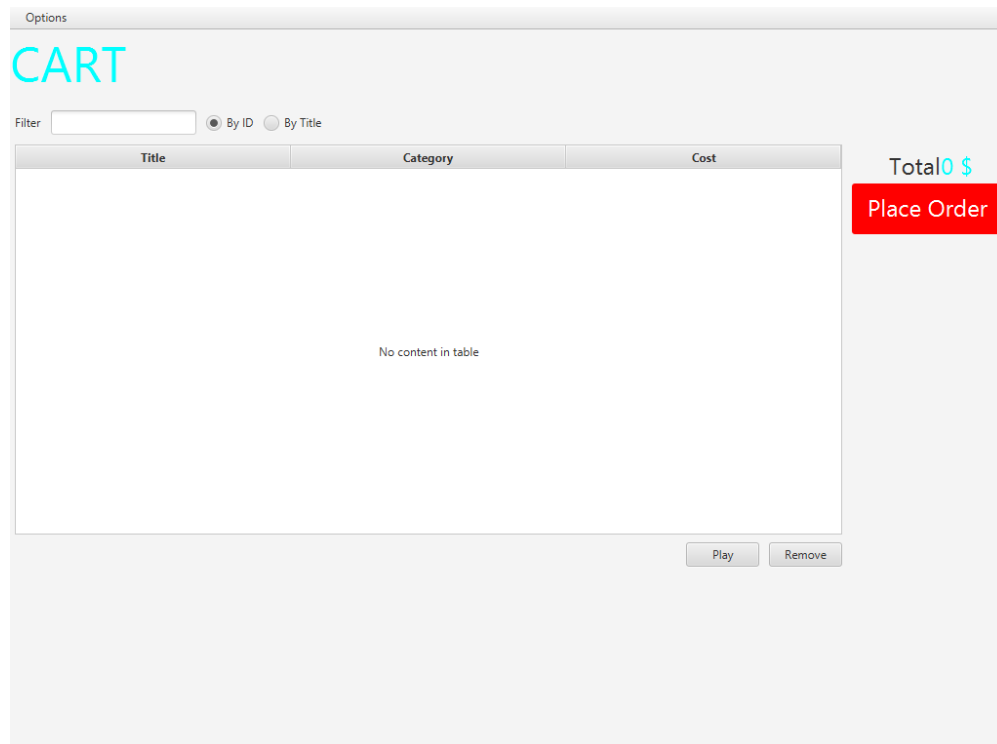


Figure 5.3: Cart.fxml 3

5.2 Create class CartScreen

```

1  package hust.soict.dsai.aims.cart;
2
3  import java.io.IOException;
4
5  import javax.swing.JFrame;
6
7  import hust.soict.dsai.aims.media.CartTungNM;
8  import javafx.application.Platform;
9  import javafx.embed.swing.JFXPanel;
10 import javafx.fxml.FXMLLoader;
11 import javafx.scene.Parent;
12 import javafx.scene.Scene;
13
14 public class CartScreenTungNM extends JFrame{
15     private CartTungNM cart;
16     public CartScreenTungNM(CartTungNM cart) {
17         super();
18         this.cart = cart;
19
20         JFXPanel fxPanel = new JFXPanel();
21         this.add(fxPanel);
22         this.setTitle(title:"Cart");
23         this.setVisible(b:true);
24
25         Platform.runLater(new Runnable() {
26             @Override
27             public void run() {
28                 try {
29                     FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"/hust/soict/dsai/aims/cart/cart.fxml"));
30                     CartScreenController controller = new CartScreenController(cart);
31                     loader.setController(controller);
32                     Parent root = loader.load();
33                     fxPanel.setScene(new Scene(root));
34                 } catch (IOException e) {
35                     e.printStackTrace();
36                 }
37             }
38         });
39     }
40 }

```

Figure 5.4: CartScreen class

5.3 Create class CartScreenController

```

1  package hust.soict.dsai.aims.cart;
2
3  import hust.soict.dsai.aims.media.CartTungNM;
4  import hust.soict.dsai.aims.media.MediaTungNM;
5  import javafx.fxml.FXML;
6  import javafx.scene.control.TableColumn;
7  import javafx.scene.control.TableView;
8  import javafx.scene.control.ToggleGroup;
9  import javafx.scene.control.cell.PropertyValueFactory;
10
11 public class CartScreenController {
12     private CartTungNM cart;
13
14     @FXML
15     private TableColumn<MediaTungNM, String> colMediaCategory;
16
17     @FXML
18     private TableColumn<MediaTungNM, Float> colMediaCost;
19
20     @FXML
21     private TableColumn<MediaTungNM, String> colMediaTitle;
22
23     @FXML
24     private ToggleGroup filterCategory;
25
26     @FXML
27     private TableView<MediaTungNM> tblMedia;
28
29     public CartScreenController (CartTungNM cart) {
30         super();
31         this.cart = cart;
32     }

```

Figure 5.5: CartScreenController 1

```

34     @FXML
35     private void initialize() {
36         colMediaTitle.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Title"));
37         colMediaCategory.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Category"));
38         colMediaCost.setCellValueFactory(new PropertyValueFactory<MediaTungNM, Float>("Cost"));
39         tblMedia.setItems(this.cart.getItemsOrdered());
40     }
41
42 }
43

```

Figure 5.6: CartScreenController 2

5.4 Demo

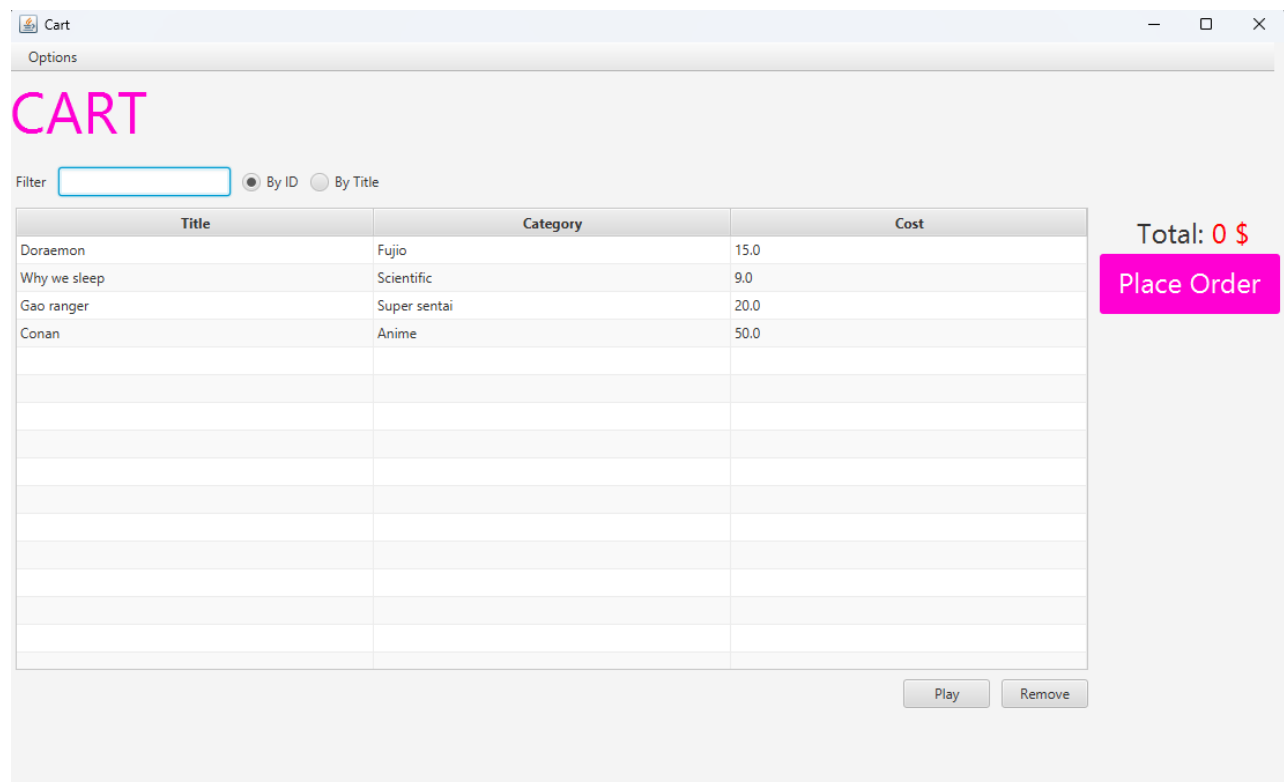


Figure 5.7: Demo CartScreen

6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController

```

1 package hust.soict.dsai.aims.cart;
2
3 import hust.soict.dsai.aims.media.CartTungNM;
4 import hust.soict.dsai.aims.media.MediaTungNM;
5 import hust.soict.dsai.aims.media.PlayableTungNM;
6 import javafx.beans.Observable;
7 import javafx.beans.value.ChangeListener;
8 import javafx.beans.value.ObservableValue;
9 import javafx.fxml.FXML;
10 import javafx.scene.control.Button;
11 import javafx.scene.control.TableColumn;
12 import javafx.scene.control.TableView;
13 import javafx.scene.control.ToggleGroup;
14 import javafx.scene.control.cell.PropertyValueFactory;
15
16 public class CartScreenController {
17     private CartTungNM cart;
18
19     @FXML
20     private Button btnPlay;
21
22     @FXML
23     private Button btnRemove;

```

Figure 6.1: CartScreenController 1

```
25  @FXML
26  private TableColumn<MediaTungNM, String> colMediaCategory;
27
28  @FXML
29  private TableColumn<MediaTungNM, Float> colMediaCost;
30
31  @FXML
32  private TableColumn<MediaTungNM, String> colMediaTitle;
33
34  @FXML
35  private ToggleGroup filterCategory;
36
37  @FXML
38  private TableView<MediaTungNM> tblMedia;
39
40  public CartScreenController (CartTungNM cart) {
41      super();
42      this.cart = cart;
43  }
44
45  @FXML
46  private void initialize() {
47      colMediaTitle.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Title"));
48      colMediaCategory.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Category"));
49      colMediaCost.setCellValueFactory(new PropertyValueFactory<MediaTungNM, Float>("Cost"));
50      tblMedia.setItems(this.cart.getItemsOrdered());
51
52      btnPlay.setVisible(false);
53      btnRemove.setVisible(false);
54
55      tblMedia.getSelectionModel().selectedItemProperty().addListener(
56          new ChangeListener<MediaTungNM>() {
57              @Override
58              public void changed(ObservableValue<? extends MediaTungNM> observable, MediaTungNM oldValue, MediaTungNM newValue) {
59                  if (newValue != null) {
60                      updateButtonBar(newValue);
61                  }
62              }
63          });
64
65      void updateButtonBar(MediaTungNM media) {
66          btnRemove.setVisible(true);
67          if(media instanceof PlayableTungNM) btnPlay.setVisible(true);
68          else btnPlay.setVisible(false);
69      }
70  }
71 }
```

Figure 6.2: CartScreenController 2

6.2 Demo

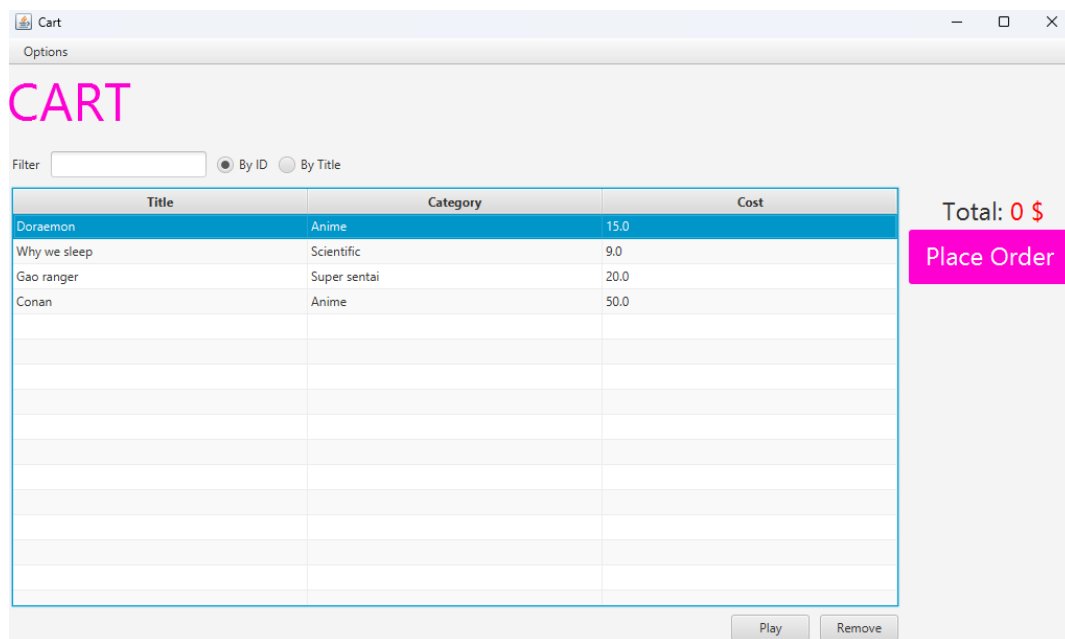


Figure 6.3: Demo media playable

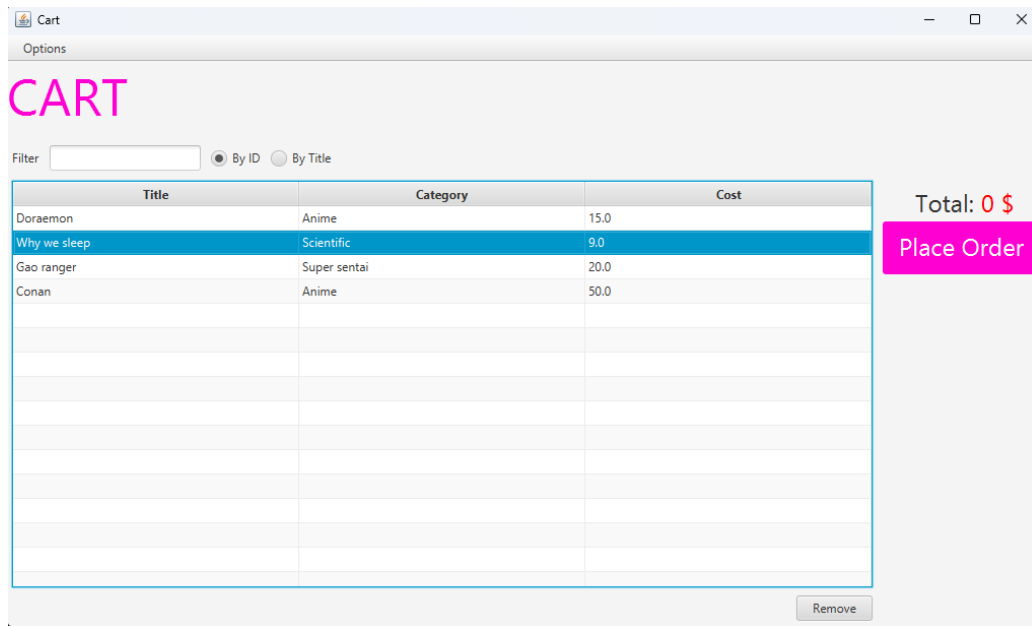


Figure 6.4: Demo media unplayable

7 Deleting a media

7.1 Code

```
72  
73     @FXML  
74     void btnRemovePressed(ActionEvent event) {  
75         MediaTungNM media = tblMedia.getSelectionModel().getSelectedItem();  
76         cart.removeMediaTungNM(media);  
77     }  
78 }  
79
```

Figure 7.1: btnRemovePressed Method

7.2 Demo

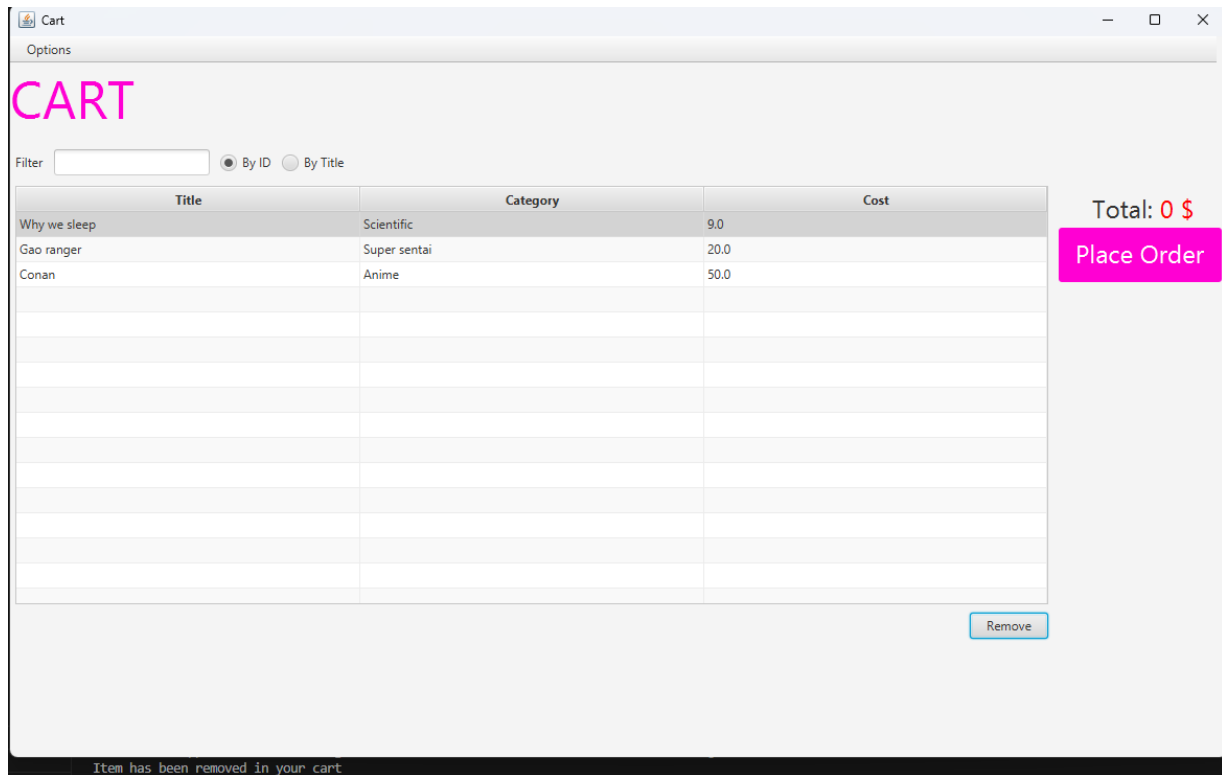


Figure 7.2: button Remove

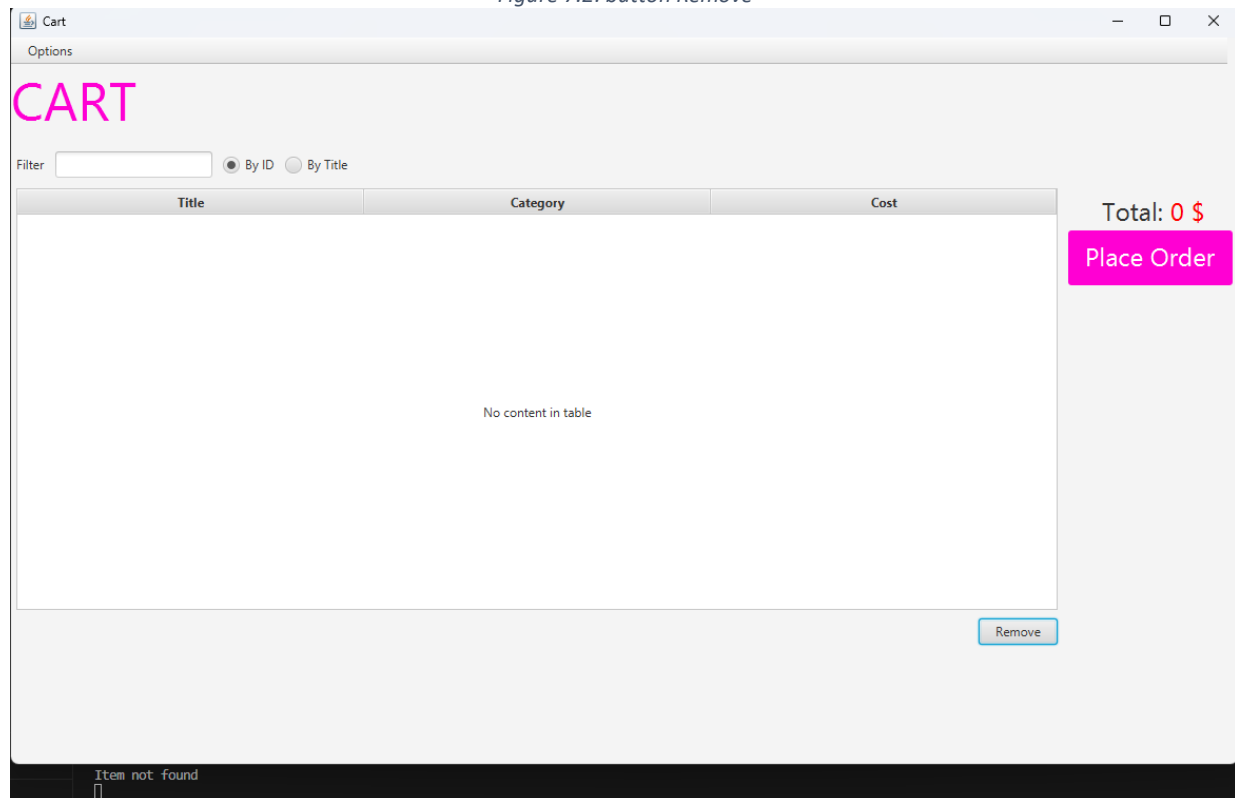


Figure 7.3: button Remove

8 Complete the Aims GUI application

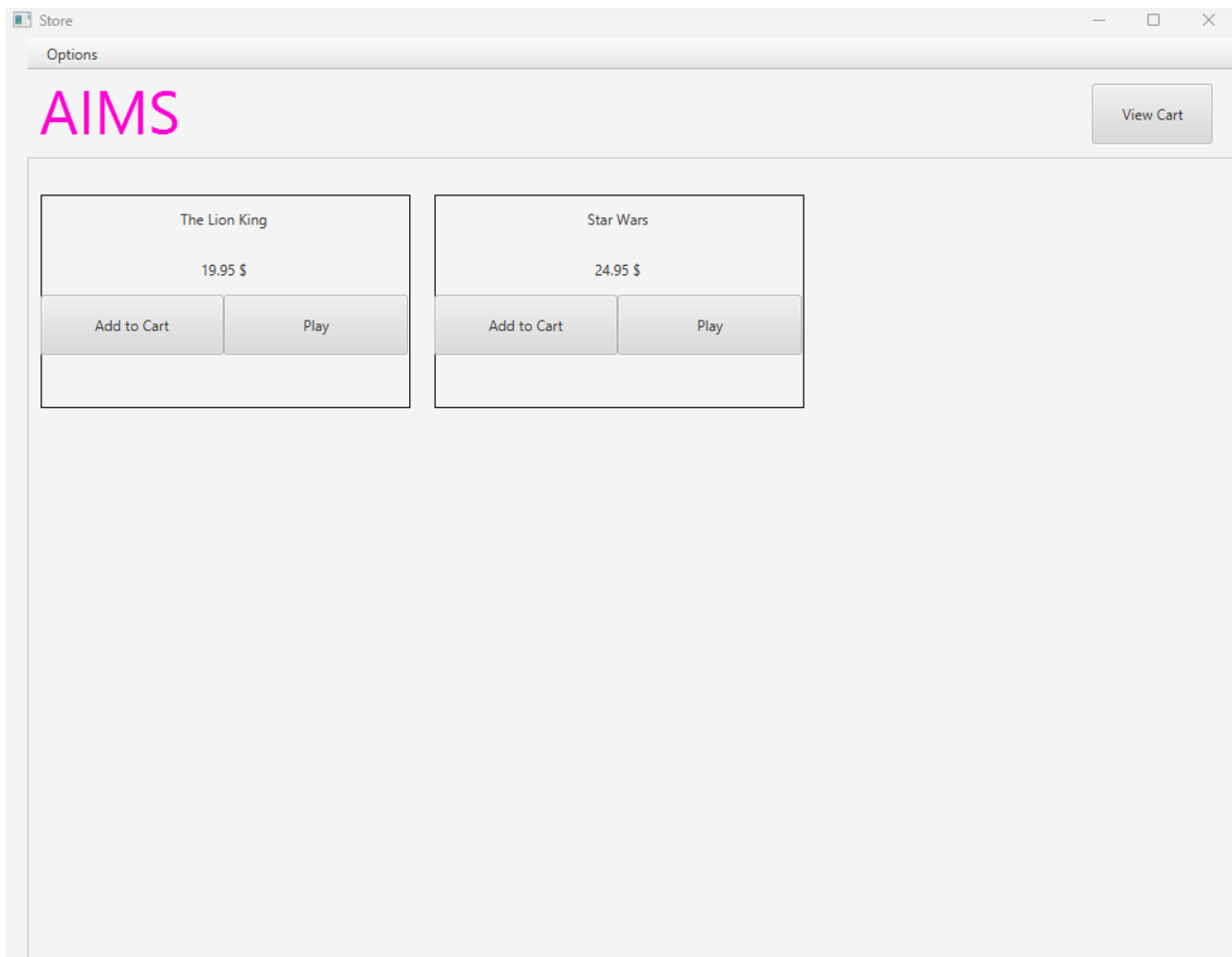


Figure 8.1: Store before add book

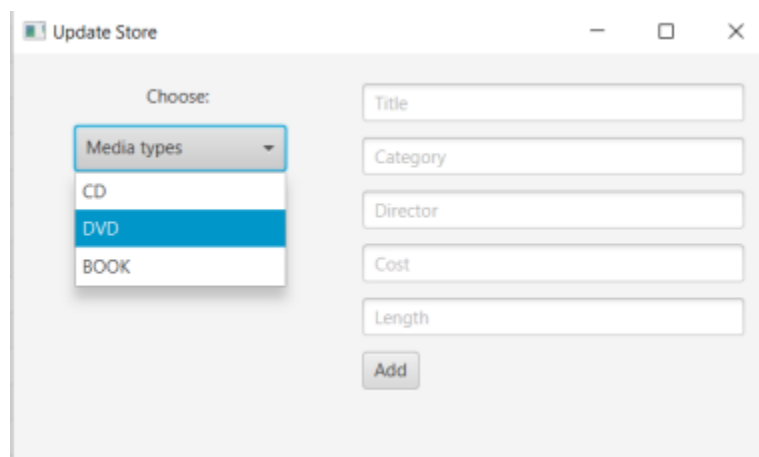


Figure 8.2: Add book

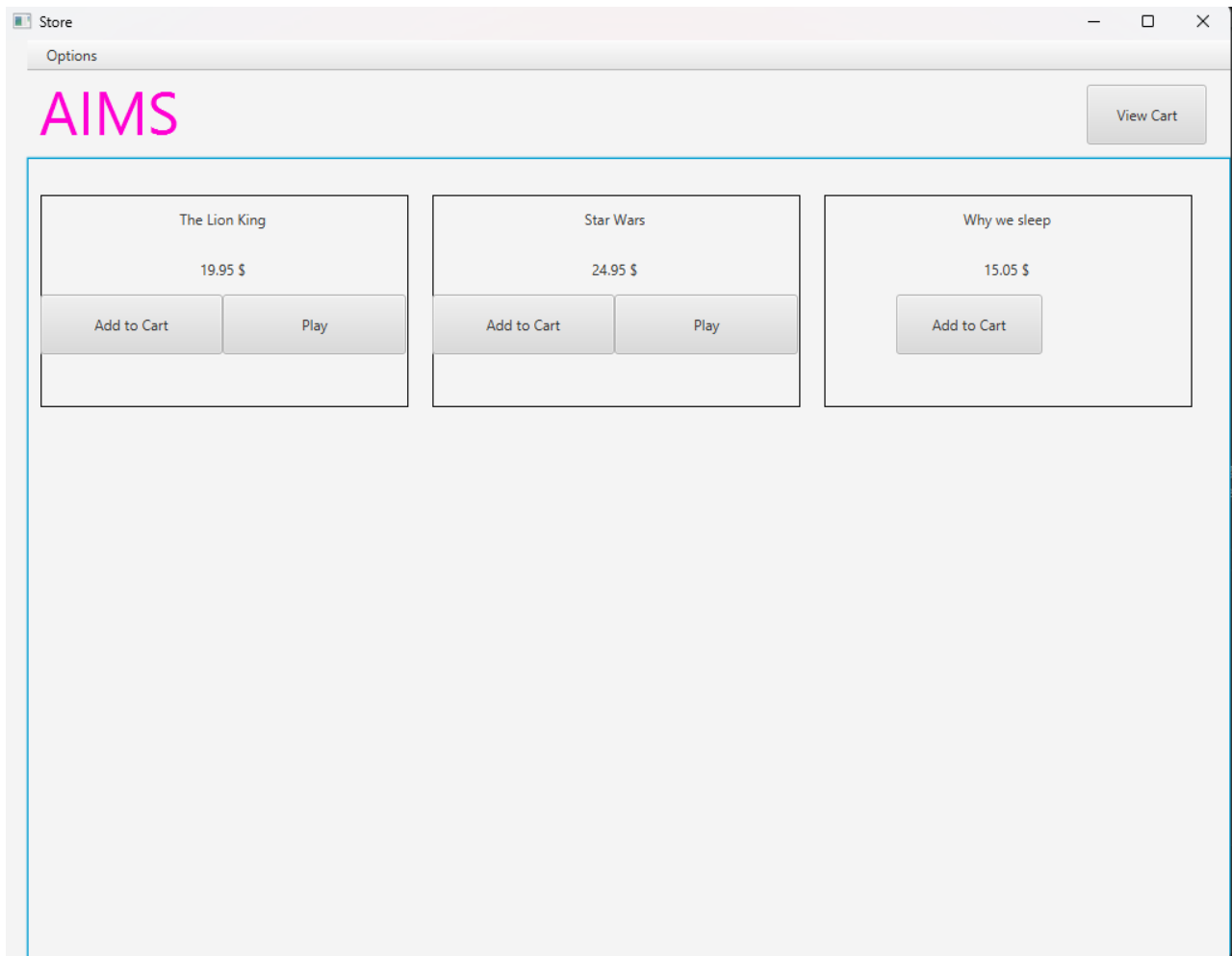


Figure 8.3: Store after add book

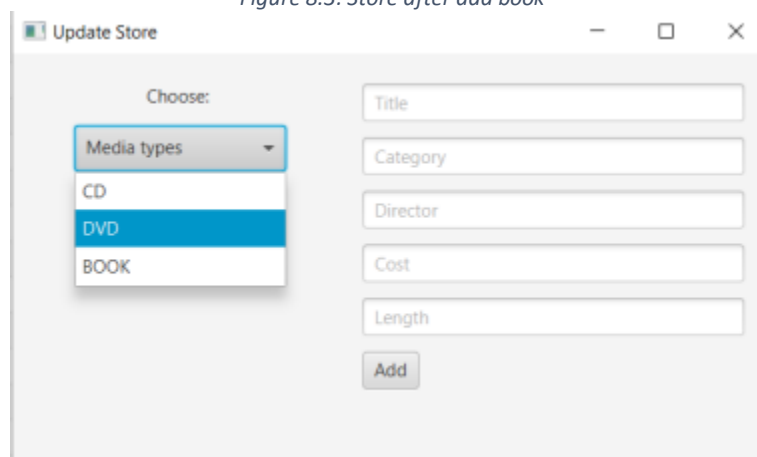


Figure 8.4: Add CD

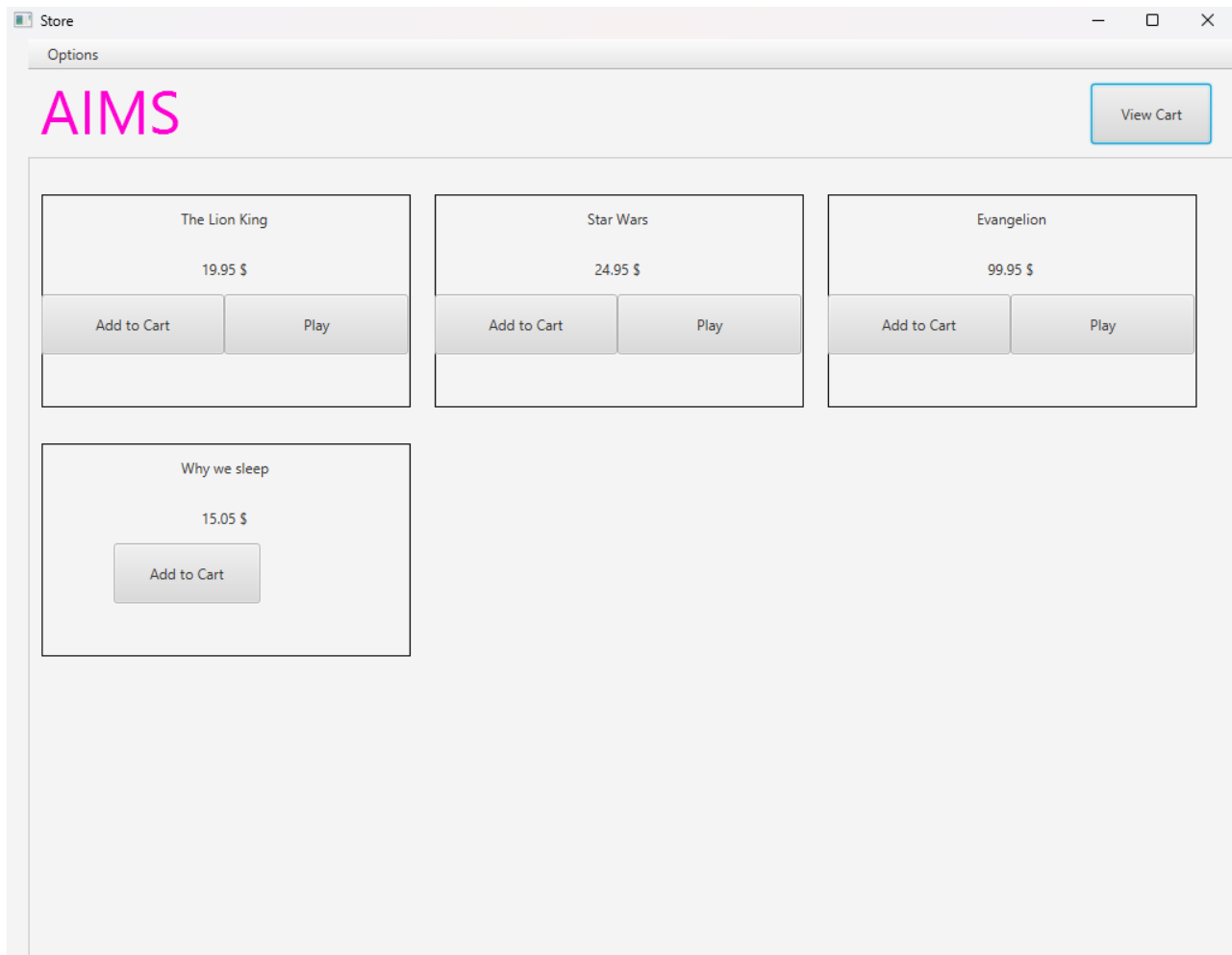


Figure 8.5: Store after add CD

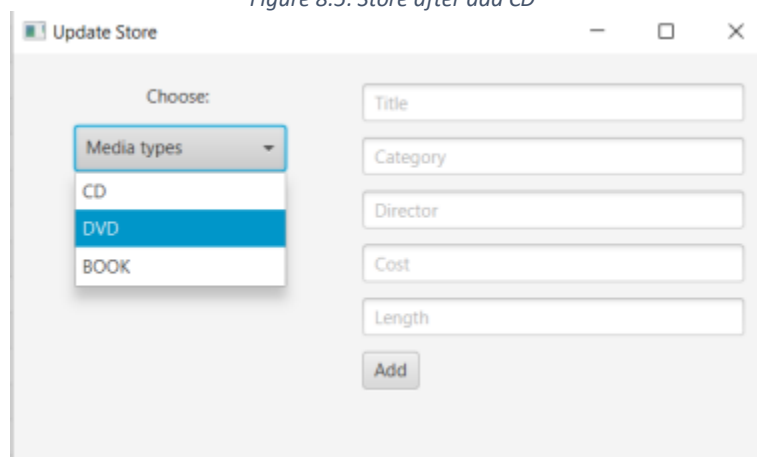


Figure 8.6 Add DVD



Figure 8.7: Store after add DVD

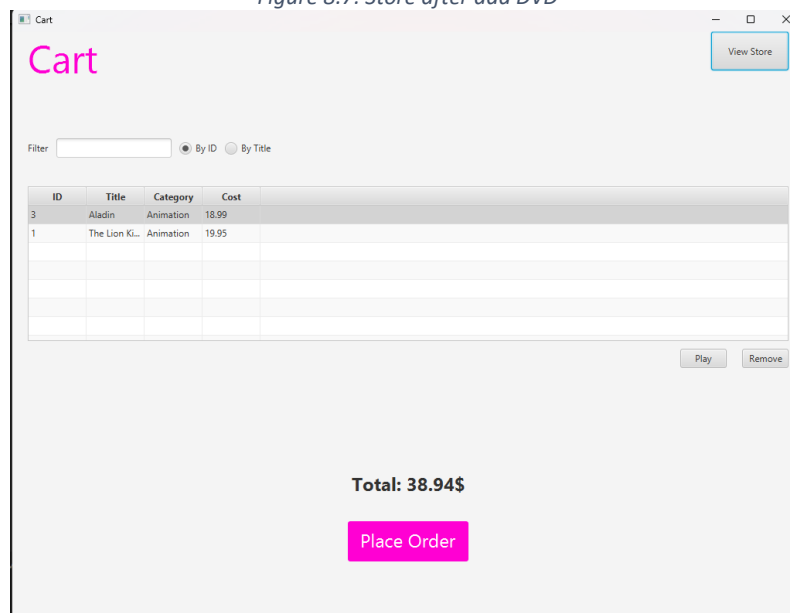


Figure 8.8: Cart

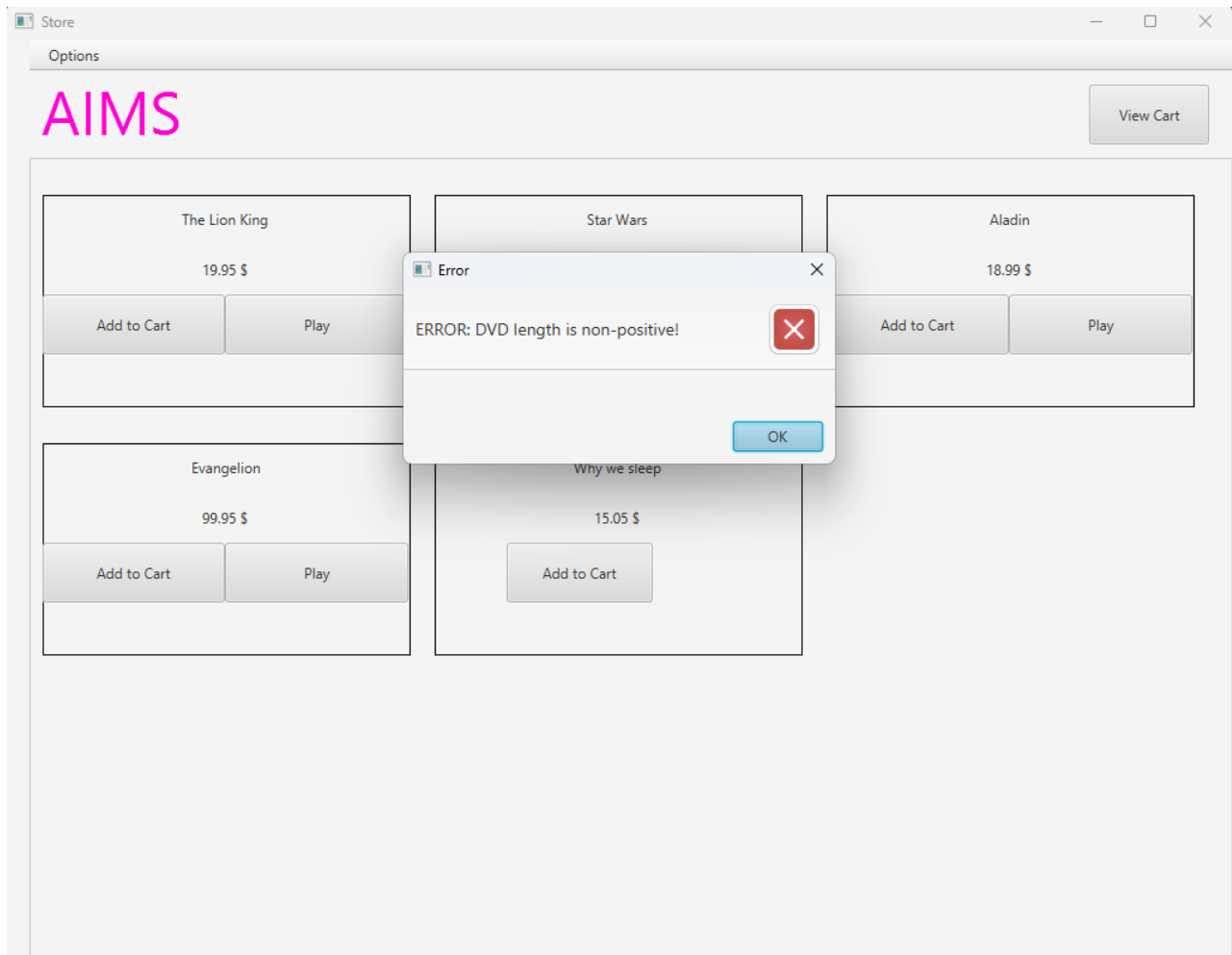
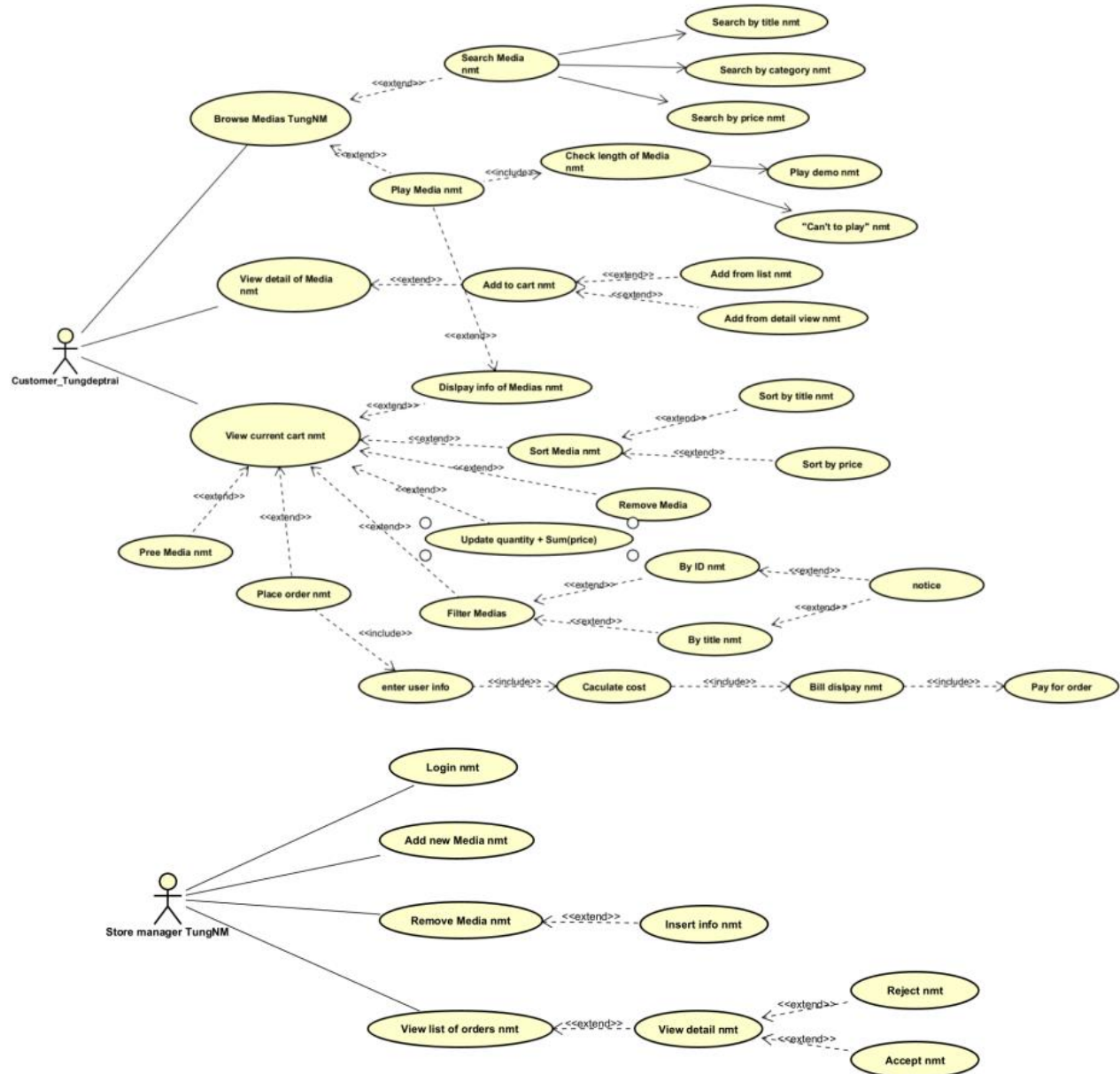


Figure 8.9: Exception

9 Use case Diagram



```

classDiagram
    class DiscTungNM {
        -length: int
        -director: String
        +DiscTungNM()
        +getLength(): int
        +setLength(length: int): void
        +getDirector(): String
        +setDirector(director: String): void
        +DiscTungNM(title: String, price: float)
        +DiscTungNM(id: int, title: String, category: String, price: float, length: int, director: String)
    }
    class DVDTungNM {
        -nbDVDTungNM: int = 0
        +DVDTungNM(title: String)
        +DVDTungNM(title: String, price: float)
        +DVDTungNM(title: String, category: String, price: float)
        +DVDTungNM(director: String, title: String, category: String, price: float)
        +DVDTungNM(title: String, category: String, director: String, length: int, price: float)
        +play(): void
    }
    class CartTungNM {
        +MAX_NUMBERS_ORDERED: int = 20
        +addMediaTungNM(media: MediaTungNM): boolean
        +removeMediaTungNM(media: MediaTungNM): boolean
        +totalPrice(): float
        +printMainCartNM(): void
        +searchByItemEnd(id: int): MediaTungNM
        +searchByTitleNM(title: String): MediaTungNM
        +sortByCost(): void
        +sortByTitle(): void
        +emptyCart(): void
    }
    class StoreTungNM {
        +MAX_MEDIA_IN_STORE: int = 50
        +addMediaTungNM(media: MediaTungNM): boolean
        +removeMediaTungNM(media: MediaTungNM): boolean
        +removeMediaTungNM(title: String): boolean
        +printItemsInStore(): void
        +searchByTitle(title: String): MediaTungNM
    }
    class MediaTungNM {
        -id: int
        -title: String
        -category: String
        -price: float
        +get(id): int
        +set(id: int): void
        +getTitle(): String
        +setTitle(title: String): void
        +getCategory(): String
        +setCategory(category: String): void
        +getPrice(): float
        +setPrice(price: float): void
        +MediaTungNM()
        +MediaTungNM(title: String, price: float)
        +MediaTungNM(title: String, category: String, price: float)
        +MediaTungNM(id: int, title: String, category: String, price: float)
        +isMatchNM(title: String): boolean
        +equals(o: Object): boolean
        +toString(): String
    }
    class BookTungNM {
        -authors: List<String> = new ArrayList<String>()
        +BookTungNM()
        +BookTungNM(id: int, title: String, category: String, price: float)
        +BookTungNM(title: String, price: float)
        +addAuthor(name: String): boolean
        +removeAuthor(name: String): boolean
    }
    class CompactDiscTungNM {
        -artist: String
        +CompactDiscTungNM(title: String, price: float)
        +CompactDiscTungNM(id: int, title: String, category: String, price: float, length: int, director: String)
        +CompactDiscTungNM(id: int, title: String, category: String, price: float, length: int, director: String, artist: String, tracks: List<TrackTungNM>)
        +addTrack(track: TrackTungNM): boolean
        +removeTrack(track: TrackTungNM): boolean
        +getTotalLength(): int
        +play(): void
    }
    class TrackTungNM {
        -title: String
        -length: int
        +getTitle(): String
        +getLength(): int
        +TrackTungNM()
        +TrackTungNM(title: String)
        +TrackTungNM(title: String, length: int)
        +play(): void
        +equals(o: Object): boolean
    }
    class List {
        <<interface>>
    }
    class Comparator {
        <<interface>>
    }
    class ComparatorMediaTungNM {
        <<interface>>
    }
    class MediaComparatorByCostTitle {
        +compare(o1: MediaTungNM, o2: MediaTungNM): int
    }
    class PlayableTungNM {
        <<interface>>
    }
    class MediaComparatorByTitleCost {
        +compare(o1: MediaTungNM, o2: MediaTungNM): int
    }

    DiscTungNM <|-- DVDTungNM
    MediaTungNM <|-- BookTungNM
    MediaTungNM <|-- CompactDiscTungNM
    MediaTungNM <|-- TrackTungNM
    MediaTungNM <|-- CartTungNM
    MediaTungNM <|-- StoreTungNM

    MediaTungNM "0" -- "*" CartTungNM : itemsOrdered
    MediaTungNM "0" -- "*" StoreTungNM : itemsInStore
    CartTungNM --> StoreTungNM : store
    MediaTungNM <..> Comparator
    MediaTungNM <..> ComparatorMediaTungNM
    MediaTungNM <..> MediaComparatorByCostTitle
    MediaTungNM <..> PlayableTungNM
    MediaTungNM <..> MediaComparatorByTitleCost
    
```