

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

BÁO CÁO THỰC HÀNH
IT3103-744528-2024.1
BÀI THỰC HÀNH 05

Họ và tên sv: Chu Đình Hà
Lớp: **K67 – Việt Nhật 03**

GVHD: Lê Thị Hoa

TA: **Đặng Mạnh Cường**

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 5

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

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

```
package hust.soict.dsai.swing;

import java.awt.*;
import java.awt.event.*;

public class AWTAccumulator extends Frame{
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;          //Accumulated sum, init to 0

    //Constructor to setup the GUI components and event handlers
    public AWTAccumulator(){
        setLayout(new GridLayout(rows:2, cols:2));

        add(new Label(text:"Enter an Integer: "));

        tfInput = new TextField(columns:10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        add(new Label(text:"The Accumulated Sum is: "));

        tfOutput = new TextField(columns:10);
        tfOutput.setEditable(b:false);
        add(tfOutput);

        setTitle(title:"AWT Accumulator");
        setSize(width:350, height:120);
        setVisible(b:true);
    }
}
```

```
Run | Debug
public static void main(String[] args) {
    new AWTAccumulator();
}

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

Figure 1.1: Source code of AWTAccumulator

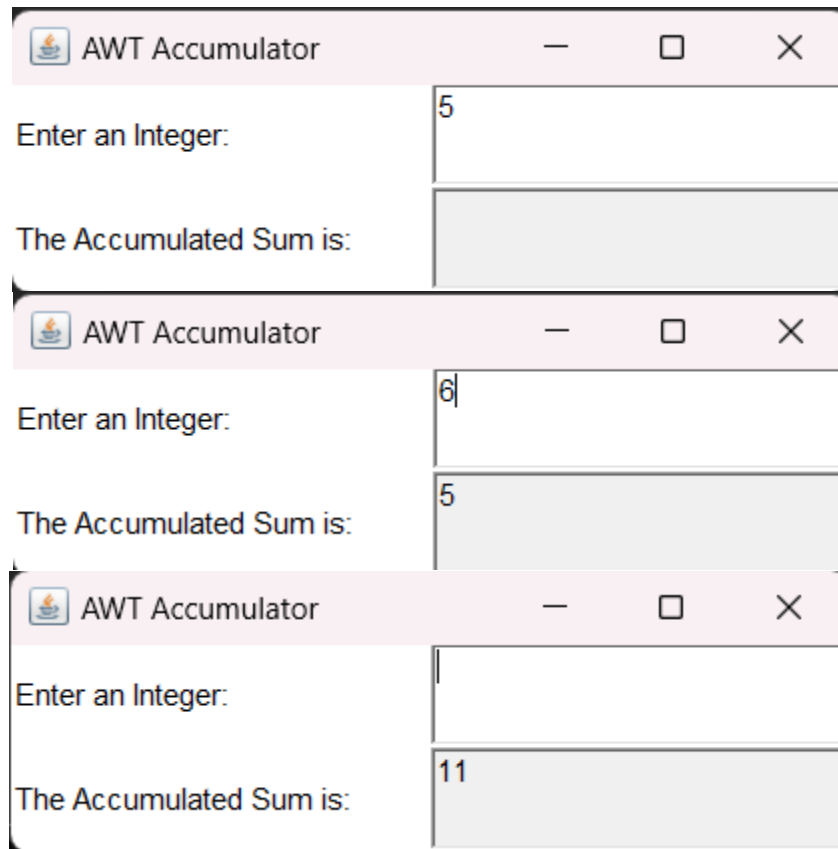


Figure 1.2: Demo of AWTAccumulator

1.2 SwingAccumulator

```
package hust.soict.dsai.swing;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*.*;

public class SwingAccumulator extends JFrame{
    private JTextField tfInput;
    private JTextField tfOutput;
    private int sum = 0;           //Accumulated sum, init to 0

    public SwingAccumulator(){
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(rows:2, cols:2));

        cp.add(new JLabel(text:"Enter an Integer"));

        tfInput = new JTextField(columns:10);
        cp.add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        cp.add(new JLabel(text:"The Accumulated Sum is: "));

        tfOutput = new JTextField(columns:10);
        tfOutput.setEditable(b:false);
        cp.add(tfOutput);

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



```
Run | Debug
public static void main(String[] args) {
    new SwingAccumulator();
}

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

Figure 1.3: Source code of SwingAccumulator

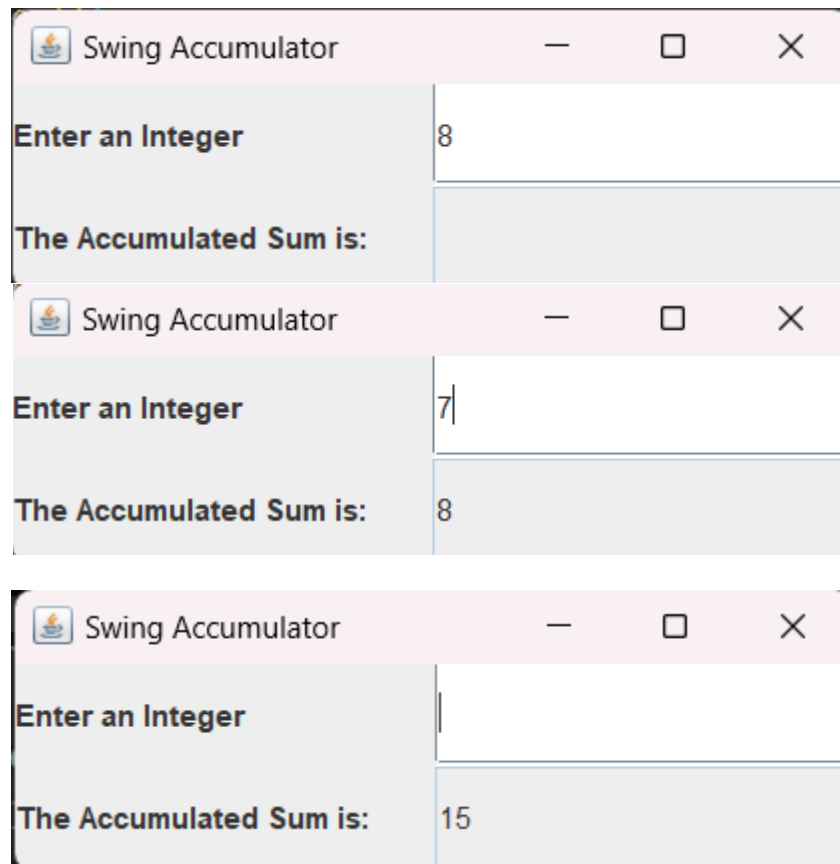


Figure 1.4: Demo of SwingAccumulator

2 Organizing Swing components with Layout Managers

2.1 Code

```
package hust.soict.dsai.swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class NumberGrid extends JFrame{
    private JButton[] btnNumbers = new JButton[10];
    private JButton btnDelete, btnReset;
    private JTextField tfDisplay;

    public NumberGrid(){
        tfDisplay = new JTextField();
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);

        JPanel panelButtons = new JPanel(new GridLayout(4, 3));
        addButtons(panelButtons);

        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(tfDisplay, BorderLayout.NORTH);
        cp.add(panelButtons, BorderLayout.CENTER);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle(title:"Number Grid");
        setSize(width:200, height:200);
        setVisible(b:true);
    }
}
```

Figure 2.1: Source code of NumberGrid 1

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

    btnDelete = new JButton(text:"DEL");
    panelButtons.add(btnDelete);
    btnDelete.addActionListener(btnListener);

    btnNumbers[0] = new JButton(text:"0");
    panelButtons.add(btnNumbers[0]);
    btnNumbers[0].addActionListener(btnListener);

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

```
private class ButtonListener implements ActionListener{
    @Override
    public void actionPerformed(ActionEvent e){
        String button = e.getActionCommand();
        if(button.charAt(index:0) >= '0' && button.charAt(index:0) <= '9'){
            tfDisplay.setText(tfDisplay.getText() + button);
        }
        else if(button.equals(anObject:"DEL")){
            String deleteString = tfDisplay.getText();
            if(deleteString.length() > 0){
                deleteString = deleteString.substring(beginIndex:0, deleteString.length() - 1);
            }
            tfDisplay.setText(deleteString);
        }
        else{
            tfDisplay.setText(t:"");
        }
    }
}

public static String delLastCharacter(String str){
    if((str != null) && (str.length() > 0)){
        return str.substring(beginIndex:0, str.length() - 1);
    }
    return null;
}

Run | Debug
public static void main(String[] args) {
    new NumberGrid();
}
```

Figure 2.2: Source code of NumberGrid 2

2.2 Demo

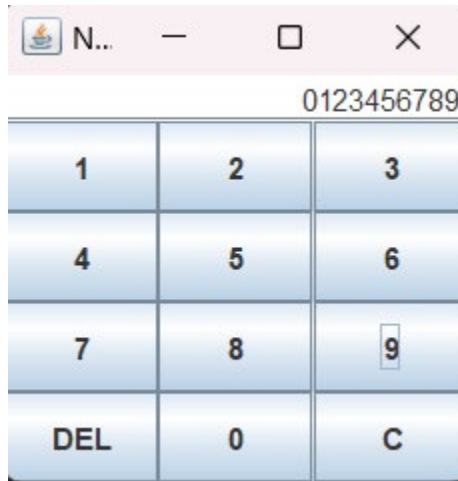


Figure 2.3: Demo buttons 0-9

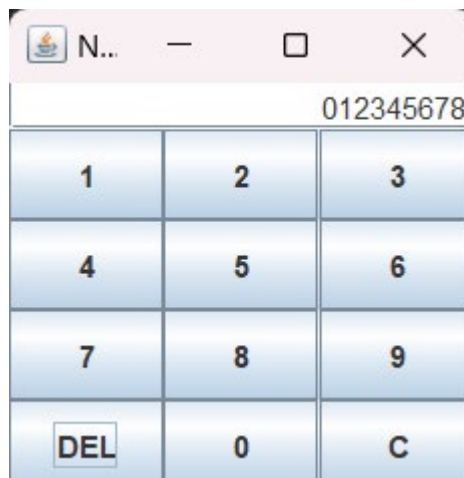


Figure 2.4: Demo DEL button

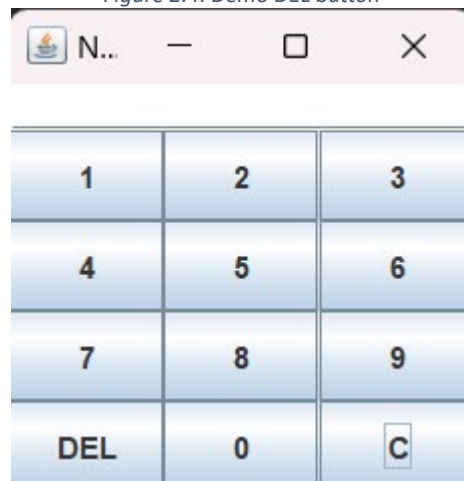


Figure 2.5: Demo

3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

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

import javax.swing.*;

import hust.soict.dsai.aims.media.Media;
import hust.soict.dsai.aims.store.Store;

import java.awt.*;
import java.util.ArrayList;

public class StoreScreen extends JFrame{
    private Store store;

    JPanel createNorth(){
        JPanel north = new JPanel();
        north.setLayout(new BorderLayout(north, BorderLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }

    JMenuBar createMenuBar(){
        JMenu menu = new JMenu(s:"Options");

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

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

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

Figure 3.1: Class StoreScreen 1

```
JPanel createHeader(){
    JPanel header = new JPanel();
    header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

    JLabel title = new JLabel(text:"AIMS");
    title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:50));
    title.setForeground(Color.CYAN);

    JButton cart = new JButton(text:"View cart");
    cart.setPreferredSize(new Dimension(width:100, height:50));
    cart.setMaximumSize(new Dimension(width:100, height:50));

    header.add(Box.createRigidArea(new Dimension(width:10, height:10)));
    header.add(title);
    header.add(Box.createHorizontalGlue());
    header.add(cart);
    header.add(Box.createRigidArea(new Dimension(width:10, height:10)));

    return header;
}

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

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

Figure 3.2: Class StoreScreen 2

```
public StoreScreen(Store store){
    this.store = store;
    Container cp = getContentPane();
    cp.setLayout(new BorderLayout());

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

    setVisible(b:true);
    setTitle(title:"Store");
    setSize(width:1024, height:768);
}
}
```

Figure 3.3: Class StoreScreen 3

```
public static void main(String[] args) {
    DigitalVideoDisc dvd1 = new DigitalVideoDisc(id:1, title:"The Lion King", category:"Animation", director:"Roger Aller", length:87, cost
    DigitalVideoDisc dvd2 = new DigitalVideoDisc(id:2, title:"Star War", category:"Science Fiction", director:"George Lucas", length:87, cc
    DigitalVideoDisc dvd3 = new DigitalVideoDisc(id:3, title:"Alladin", category:"Animation", director:null,length:0, cost:18.99f);

    Book book1 = new Book(id:4, title:"Sherlock Holmes", category:"Thriller", cost:21.89f);
    Book book2 = new Book(id:5, title:"One Thousand and One Nights", category:"Adventure", cost:25.99f);
    Book book3 = new Book(id:6, title:"Robin Hood", category:"Novel", cost:18.89f);

    CompactDisc cd1 = new CompactDisc(id:7, title:"Tobu Best Songs Collection", category:"Music", cost:45.99f, director:null, length:15, ar
    Track track1CD1 = new Track(title:"Color", length:5);
    Track track2CD1 = new Track(title:"Life", length:5);
    Track track3CD1 = new Track(title:"High", length:4);

    CompactDisc cd2 = new CompactDisc(id:8, title:"Lé Miserablé", category:"Music", cost:35.99f, director:null, length:10, arti_null);
    Track track1CD2 = new Track(title:"I Dreamed a Dream", length:5);
    Track track2CD2 = new Track(title:"Look Down", length:2);
    Track track3CD2 = new Track(title:"Stars", length:3);

    CompactDisc cd3 = new CompactDisc(id:9, title:"Katie Perrie", category:"Music", cost:63.99f, director:null, length:17, arti..."Katy Perry
    Track track1CD3 = new Track(title:"Roar", length:6);
    Track track2CD3 = new Track(title:"Firework", length:5);
    Track track3CD3 = new Track(title:"Dark Horse", length:5);
}
```

Figure 3.4: Class StoreScreen 4


```
cd1.addTrack_hacd(track1CD1);  
cd1.addTrack_hacd(track2CD1);  
cd1.addTrack_hacd(track3CD1);  
  
cd2.addTrack_hacd(track1CD2);  
cd2.addTrack_hacd(track2CD2);  
cd2.addTrack_hacd(track3CD2);  
  
cd2.addTrack_hacd(track1CD3);  
cd2.addTrack_hacd(track2CD3);  
cd2.addTrack_hacd(track3CD3);  
  
store.addMedia_hacd(dvd1);  
store.addMedia_hacd(dvd2);  
store.addMedia_hacd(dvd3);  
store.addMedia_hacd(book1);  
store.addMedia_hacd(book2);  
store.addMedia_hacd(book3);  
store.addMedia_hacd(cd1);  
store.addMedia_hacd(cd2);  
store.addMedia_hacd(cd3);  
  
new StoreScreen(store);  
}  
}
```

Figure 3.5: Class StoreScreen 5

Figure 3.6: Class StoreScreen 6

3.2 Create class MediaStore

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

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.naming.LimitExceededException;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;

import hust.soict.dsai.aims.media.*;
import hust.soict.dsai.aims.cart.*;

public class MediaStore extends JPanel{
    private Media media;
    private Cart cart;
    public MediaStore(Media media){
        this.media = media;
        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

        JLabel title = new JLabel(media.getTitle_hacd());
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:20));
        title.setAlignmentX(CENTER_ALIGNMENT);
    }
}
```

Figure 3.7: Class MediaStore 1

```

JLabel cost = new JLabel("" + media.getCost_hacd() + "$");
cost.setAlignmentX(CENTER_ALIGNMENT);

JPanel container = new JPanel();
container.setLayout(new FlowLayout(FlowLayout.CENTER));

if(media instanceof Playable){
    JButton playButton = new JButton(text:"Play");
    playButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e){
            JDialog dialog = new JDialog();
            dialog.setTitle(media.getTitle_hacd());
            dialog.setSize(width:400, height:300);

            String mediaInfo = "";
            try{
                mediaInfo = "<html>" + media.play_hacd().replace(target:"\n", replacement:"<br/>") + "</html>";
                JLabel mediLabel = new JLabel(mediaInfo);
                mediLabel.setHorizontalAlignment(JLabel.CENTER);
                mediLabel.setVerticalAlignment(JLabel.CENTER);
                JScrollPane scrollPane = new JScrollPane(mediLabel);
                scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

                dialog.add(scrollPane);
                dialog.setVisible(b:true);
            }catch (Exception ex){
                JOptionPane.showMessageDialog(parentComponent:null, ex.getMessage(), title:"ERROR", JOptionPane.ERROR_MESSAGE);
            }
        }
    });
}
container.add(new JButton(text:"Play"));

JButton addtoCartButton = new JButton(text:"Add to Cart");
addtoCartButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e){
        try{
            String mes = cart.addMedia_hacd(media);
            JOptionPane.showMessageDialog(parentComponent:null, mes);
        }catch (Exception ex){
            JOptionPane.showMessageDialog(parentComponent:null, ex.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
container.add(addtoCartButton);

this.add(Box.createVerticalGlue());
this.add(title);
this.add(cost);
this.add(Box.createVerticalGlue());
this.add(container);

this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
}

```

Figure 3.8: Class MediaStore 2

Figure 3.9: Class MediaStore 3

3.3 Demo

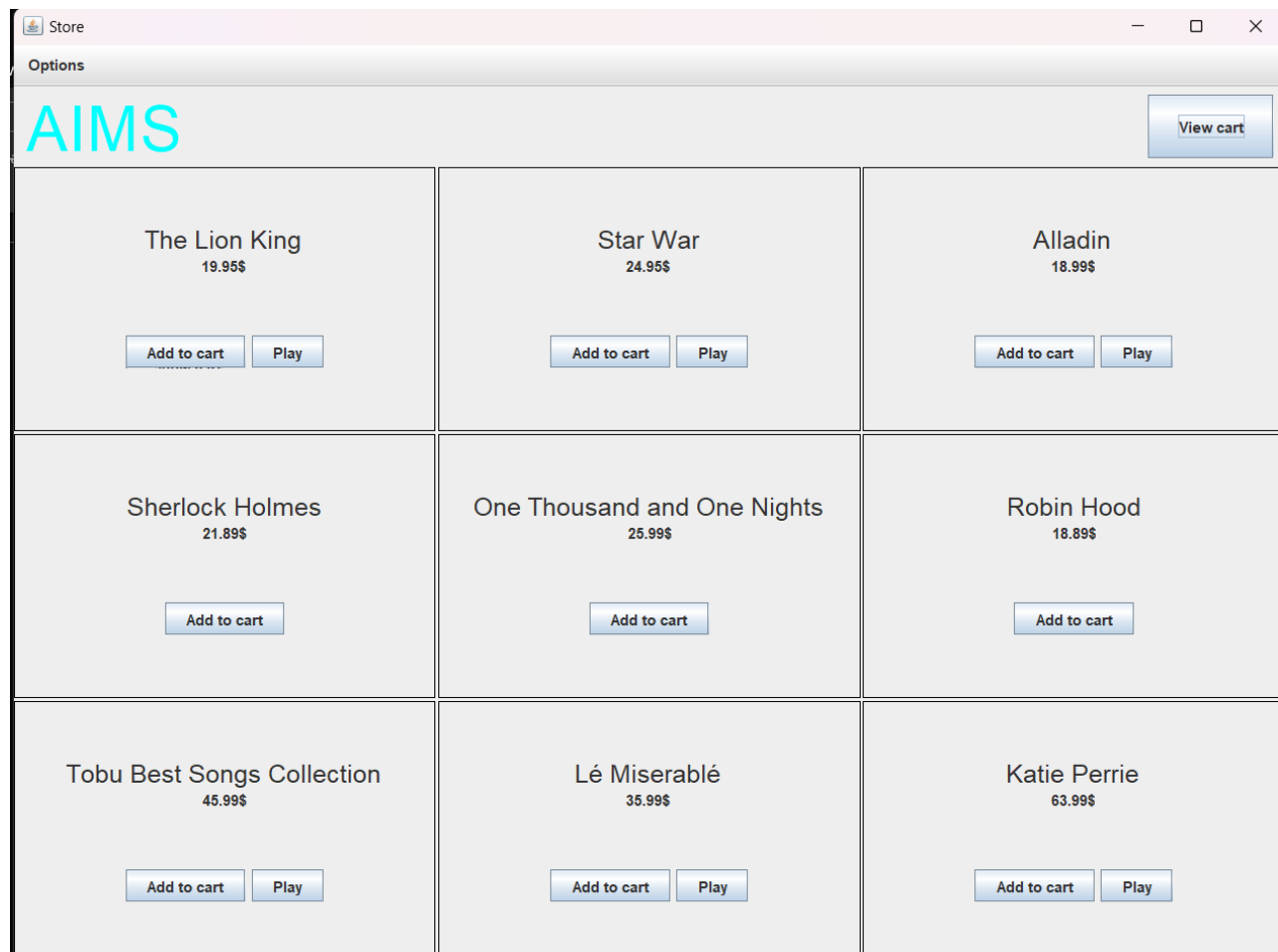


Figure 3.10: StoreScreen

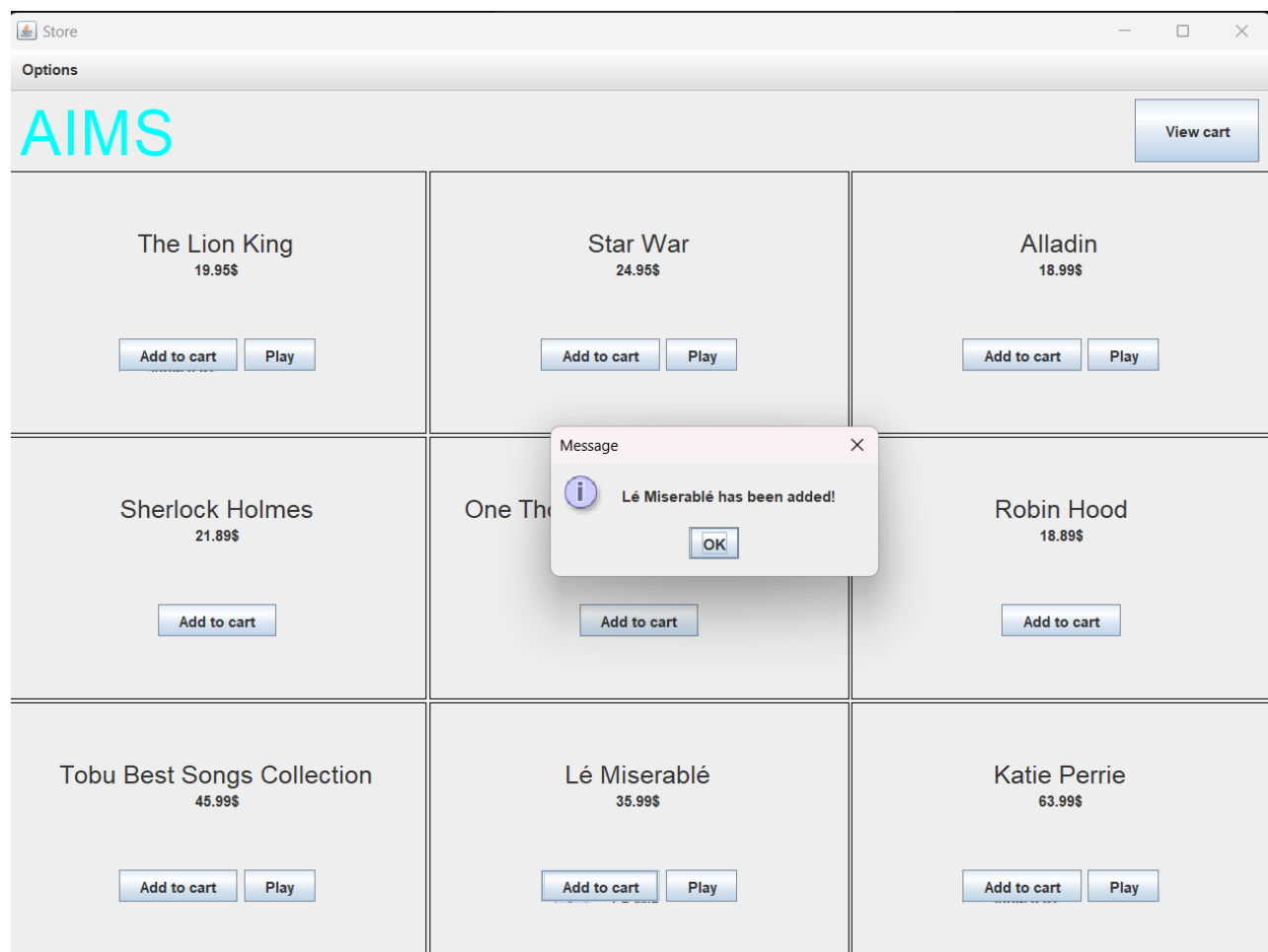


Figure 3.11 Demo Add to cart button

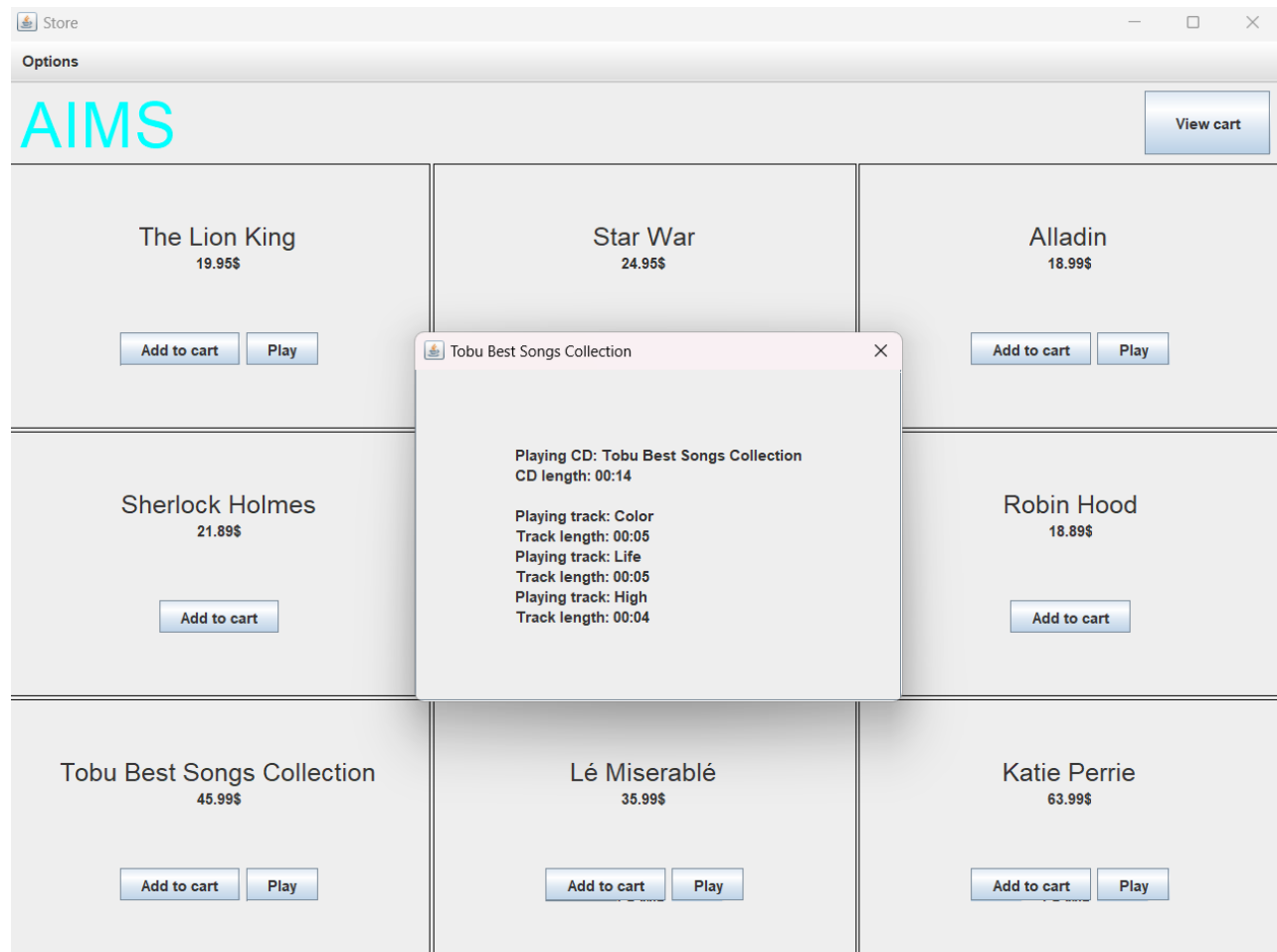


Figure 3.12 Demo Play button

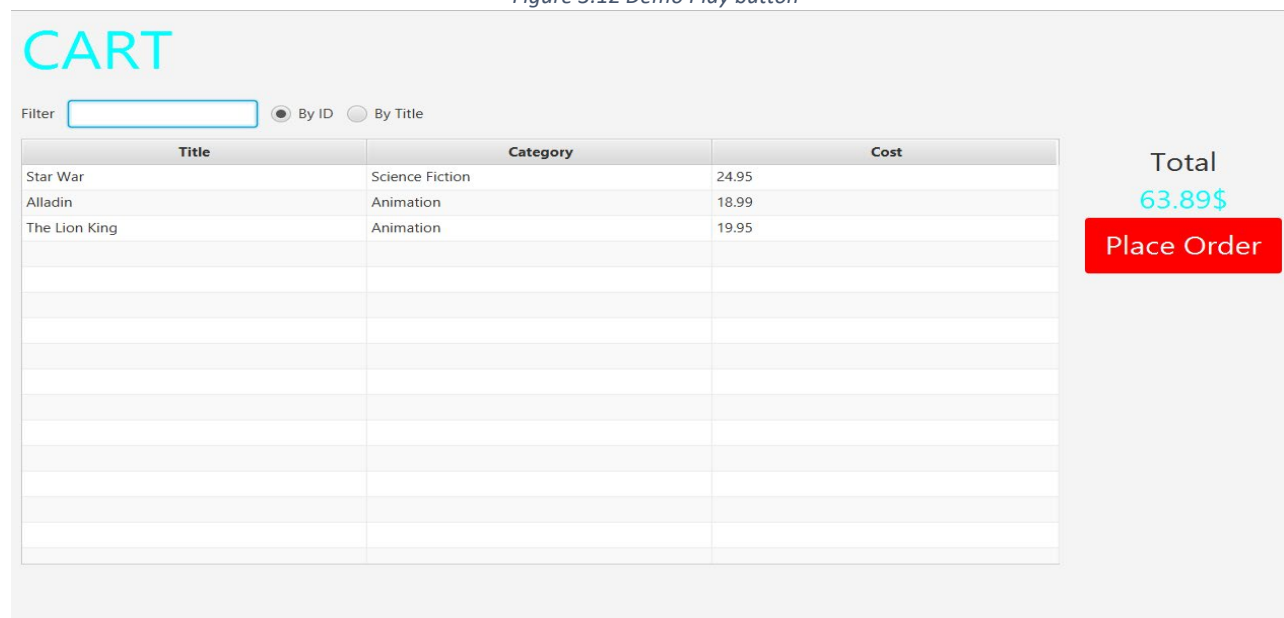


Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

```
package hust.soict.dsai.javafx;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Painter extends Application{
    public void start(Stage stage) throws Exception{
        Parent root = FXMLLoader.load(getClass().getResource(name: "/hust/soict/dsai/javafx/Painter.fxml"));

        Scene scene = new Scene(root);
        stage.setTitle("Painter");
        stage.setScene(scene);
        stage.show();
    }

    Run | Debug
    public static void main(String[] args) {
        launch(args);
    }
}
```

Figure 4.1: Class Painter

4.2 Create Painter.fxml

```
35 <?import javafx.geometry.Insets?>
36 <?import javafx.scene.control.Button?>
37 <?import javafx.scene.control.ButtonBar?>
38 <?import javafx.scene.control.RadioButton?>
39 <?import javafx.scene.control.TitledPane?>
40 <?import javafx.scene.control.ToggleGroup?>
41 <?import javafx.scene.layout.AnchorPane?>
42 <?import javafx.scene.layout.BorderPane?>
43 <?import javafx.scene.layout.Pane?>
44 <?import javafx.scene.layout.VBox?>
45 <?import javafx.scene.text.Font?>
46
47 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0" xmlns="http://javafx.com/javafx/8.0.60" xmlns:fx="http://javafx.com/fxml/1">
48     <center>
49         <Pane maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" onMouseDragged="#drawingAreaMouseDragged" prefHeight="200.0" prefWidth="400.0">
50             <BorderPane.margin>
51                 <Insets right="8.0" />
52             </BorderPane.margin>
53         </Pane>
54     </center>
55     <left>
56         <VBox fx:id="drawingAreaPane" maxHeight="1.7976931348623157E308" prefHeight="200.0" prefWidth="100.0" spacing="8.0" BorderPane.alignment="top">
57             <children>
58                 <ButtonBar prefHeight="40.0" prefWidth="200.0">
59                     <TitledPane animated="false" maxWidth="1.7976931348623157E308" text="Tools">
60                         <content>
61                             <AnchorPane maxHeight="-Infinity" maxWidth="1.7976931348623157E308" minHeight="0.0" minWidth="0.0" prefWidth="60.0">
62                                 <children>
63                                     <RadioButton layoutY="8.0" mnemonicParsing="false" text="Pen">
64                                         <padding>
65                                             <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
66                                         </padding>
67                                     </RadioButton>
68                                 </children>
69                             </AnchorPane>
70                         </content>
71                     </TitledPane>
72                 </ButtonBar>
73             </children>
74         </VBox>
75     </left>
76 </BorderPane>
```



```
        </padding>
        <toggleGroup>
            <ToggleGroup fx:id="penOption" />
        </toggleGroup>
    </RadioButton>
    <RadioButton layoutY="42.0" mnemonicParsing="false" text="Eraser" toggleGroup="$penOption">
        <padding>
            <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
        </padding>
    </RadioButton>
</children>
</AnchorPane>
</content>
</TitledPane>
<Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear">
    <font>
        <Font size="13.0" />
    </font>
</Button>
</children>
<BorderPane.margin>
    <Insets right="8.0" />
</BorderPane.margin>
</VBox>
</left>
<padding>
    <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
</padding>
</BorderPane>
```

Figure 4.2: Painter.fxml 1

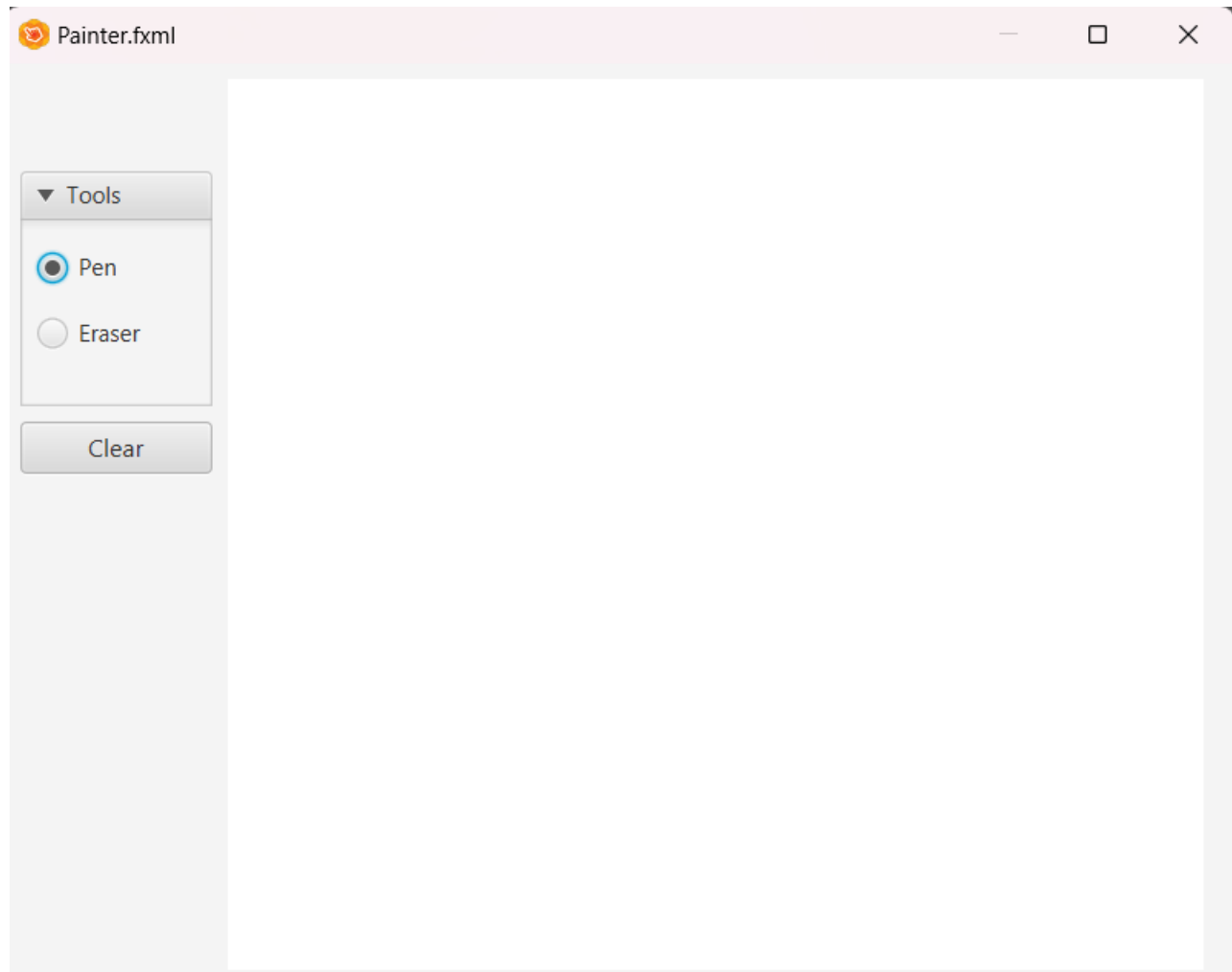


Figure 4.3: Painter.fxml 2

4.3 Create class PainterController

```
package hust.soict.dsai.javafx;

import javafx.event.*;
import javafx.fxml.FXML;
import javafx.scene.input.*;
import javafx.scene.layout.*;
import javafx.scene.paint.*;
import javafx.scene.shape.*;

public class PainterController {

    @FXML
    private VBox drawingAreaPane;

    @FXML
    void clearButtonPressed(ActionEvent event) {
        drawingAreaPane.getChildren().clear();
    }

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Circle newCircle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
        drawingAreaPane.getChildren().add(newCircle);
    }
}
```

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

```
<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" >
  <top>
    <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
      <children>
        <MenuBar>
          <menus>
            <Menu mnemonicParsing="false" text="Options">
              <items>
                <Menu mnemonicParsing="false" text="Update Store">
                  <items>
                    <MenuItem mnemonicParsing="false" text="Add Book" />
                    <MenuItem mnemonicParsing="false" text="Add CD" />
                    <MenuItem mnemonicParsing="false" text="Add DVD" />
                  </items>
                </Menu>
                <Menu mnemonicParsing="false" text="View Store">
                  <items>
                    <MenuItem mnemonicParsing="false" text="About" />
                  </items>
                </Menu>
                <Menu mnemonicParsing="false" text="View Cart">
                  <items>
                    <MenuItem mnemonicParsing="false" text="Action 1" />
                  </items>
                </Menu>
              </items>
            </Menu>
          </menus>
        </MenuBar>
        <Label text="CART" textFill="AQUA">
          <font>
            <Font size="50.0" />
          </font>
          <padding>
            <Insets left="10.0" />
          </padding>
        </Label>
      </children>
    </VBox>
  </top>
  <center>
    <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
      <padding>
        <Insets left="10.0" />
      </padding>
      <children>
        <HBox alignment="CENTER_LEFT" prefWidth="200.0" spacing="10.0">
          <padding>
            <Insets bottom="10.0" top="10.0" />
          </padding>
          <children>
            <RadioButton mnemonicParsing="false" text="By Title">
              <toggleGroup>
                <ToggleGroup fx:id="filterCategory" />
              </toggleGroup>
            </RadioButton>
            <RadioButton mnemonicParsing="false" selected="true" text="By ID" toggleGroup="$filterCategory" />
            <Label text="Filter" />
            <TextField />
          </children>
        </HBox>
        <TableView>
          <columns>
            <TableColumn prefWidth="75.0" text="Title" />
            <TableColumn prefWidth="75.0" text="Category" />
            <TableColumn prefWidth="75.0" text="Cost" />
          </columns>
        </TableView>
      </children>
    </VBox>
  </center>
</BorderPane>
```

Figure 5.1: Cart.fxml 1

```
<columnResizePolicy>
  <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
</columnResizePolicy>
</TableView>
<ButtonBar prefHeight="40.0" prefWidth="200.0">
  <buttons>
    <Button mnemonicParsing="false" text="Play" />
    <Button mnemonicParsing="false" text="Remove" />
  </buttons>
</ButtonBar>
</children>
</VBox>
</center>
<right>
  <VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
    <padding>
      <Insets top="50.0" />
    </padding>
    <children>
      <HBox alignment="CENTER">
        <children>
          <Label lineSpacing="10.0" text="Total">
            <font>
              <Font size="24.0" />
            </font>
          </Label>
          <Label text="0 $" textFill="AQUA">
            <font>
              <Font size="24.0" />
            </font>
          </Label>
        </children>
      </HBox>
      <Button mnemonicParsing="false" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">
        <font>
          <Font size="24.0" />
        </font>
      </Button>
    </children>
  </VBox>
</right>
</BorderPane>
```

Figure 5.2: Cart.fxml 2

CART

Filter ☒ By ID ☐ By Title

Title	Category	Cost
Star War	Science Fiction	24.95
Alladin	Animation	18.99
The Lion King	Animation	19.95

Total

63.89\$

Place Order

Figure 5.3: Cart.fxml 3

5.2 Create class CartScreen

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

import javax.swing.*;
import java.io.IOException;

import javafx.application.Platform;
import javafx.embed.swing.JFXPanel;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import hust.soict.dsai.aims.cart.Cart;

public class CartScreen extends JFrame{
    private Cart cart;
    public CartScreen(){
        super();
        this.cart = cart;
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle(title:"Cart");
        this.setVisible(b:true);
        Platform.runLater(new Runnable() {
            @Override
            public void run(){
                try{
                    FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"/screen/cart.fxml"));
                    CartScreenController controller = new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                }catch(IOException e){
                    e.printStackTrace();
                }
            }
        });
    }
}
```

Figure 5.4: CartScreen class

5.3 Create class CartScreenController

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

import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.*;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.transformation.FilteredList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;

public class CartScreenController {
    private Cart cart;

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;

    @FXML
    private TableColumn<Media, Float> colMediaCost;

    @FXML
    private TableColumn<Media, String> colMediaTitle;

    @FXML
    private TableColumn<Media, String> colMediacategory;

    @FXML
    private TableView<Media> tblMedia;
```

Figure 5.5: CartScreenController 1

```
@FXML
private ToggleGroup filterCategory;

@FXML
private RadioButton radioBtnFilterId;

@FXML
private RadioButton radioBtnFilterTitle;

@FXML
private Label costLabel;

@FXML
private TextField tfFilter;

@FXML
private Button placeOrder;

@FXML
void placeOrderPressed(ActionEvent event) {
    Alert alert = new Alert(Alert.AlertType.INFORMATION, cart.placeOrder_hacd());
    alert.setTitle(title:"Order created");
    alert.setHeaderText(headerText:null);
    alert.showAndWait();
}
```

```
@FXML
void btnPlayPressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    Alert alert;
    try {
        alert = new Alert(Alert.AlertType.NONE, media.playGUI());
        alert.setTitle(title:"Playing");
        alert.setHeaderText(headerText:null);
        alert.getDialogPane().getButtonTypes().add(ButtonType.OK);
        alert.showAndWait();
    } catch (Exception e) {
        alert = new Alert(Alert.AlertType.ERROR, e.getMessage());
        alert.setTitle(title:"ERROR");
        alert.setHeaderText(headerText:null);
        alert.showAndWait();
    }
}

@FXML
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia_hacd(media);
    costLabel.setText(cart.totalCost_hacd() + " $");
}

public CartScreenController(Cart cart) {
    super();
    this.cart = cart;
}
```

```
@FXML
void initialize() {
    colMediaTitle.setCellValueFactory(
        new PropertyValueFactory<Media, String>(property:"title")
    );
    colMediacategory.setCellValueFactory(
        new PropertyValueFactory<Media, String>(property:"category")
    );
    colMediaCost.setCellValueFactory(
        new PropertyValueFactory<Media, Float>(property:"cost")
    );
    tblMedia.setItems(this.cart.getItemsOrdered_hacd());

    costLabel.setText(cart.totalCost_hacd() + "$");

    btnPlay.setVisible(false);
    btnRemove.setVisible(false);
}
```

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

```
tblMedia.getSelectionModel().selectedItemProperty().addListener(  
    new ChangeListener<Media>() {  
  
        @Override  
        public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {  
            if (newValue != null) {  
                updateButtonBar(newValue);  
            }  
        }  
  
        private void updateButtonBar(Media media) {  
            btnRemove.setVisible(true);  
            if (media instanceof Playable) {  
                btnPlay.setVisible(true);  
            } else {  
                btnPlay.setVisible(false);  
            }  
        }  
    }  
);
```

Figure 6.1: CartScreenController 1

```
tfFilter.textProperty().addListener(  
    new ChangeListener<String>() {  
  
        @Override  
        public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {  
            showFilteredMedia(newValue);  
        }  
  
        private void showFilteredMedia(String keyword) {  
            FilteredList<Media> filteredList = new FilteredList<>(cart.getItemsOrdered_hacd());  
  
            if (!keyword.isEmpty() && radioBtnFilterId.isSelected()) {  
                filteredList.setPredicate(media -> {  
                    String idString = String.valueOf(media.getId_hacd());  
                    return idString.equals(keyword);  
                });  
            } else if (!keyword.isEmpty() && radioBtnFilterTitle.isSelected()) {  
                filteredList.setPredicate(media -> {  
                    String title = media.getTitle_hacd().toLowerCase();  
                    return title.contains(keyword.toLowerCase());  
                });  
            } else {  
                filteredList.setPredicate(null);  
            }  
            tblMedia.setItems(filteredList);  
        }  
    }  
);
```

Figure 6.2: CartScreenController 2

6.2 Demo

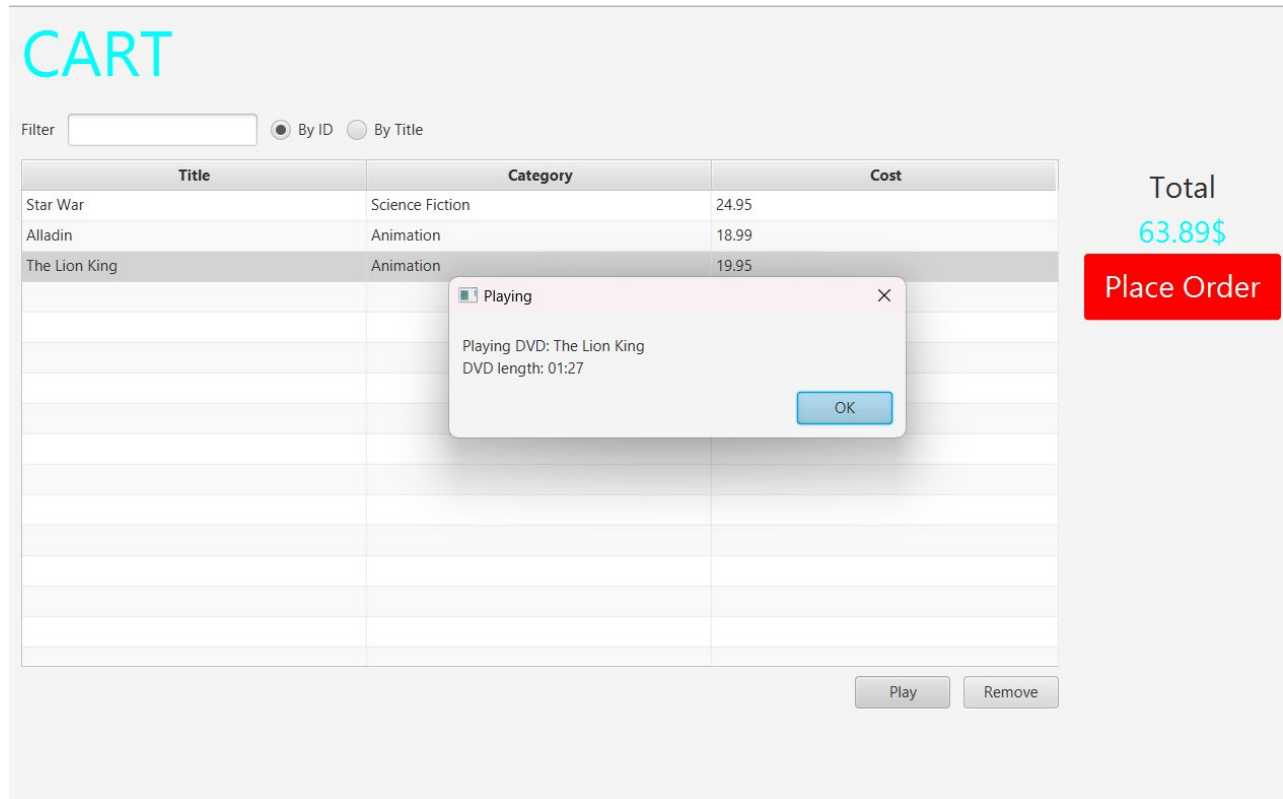


Figure 6.3: Demo media playable

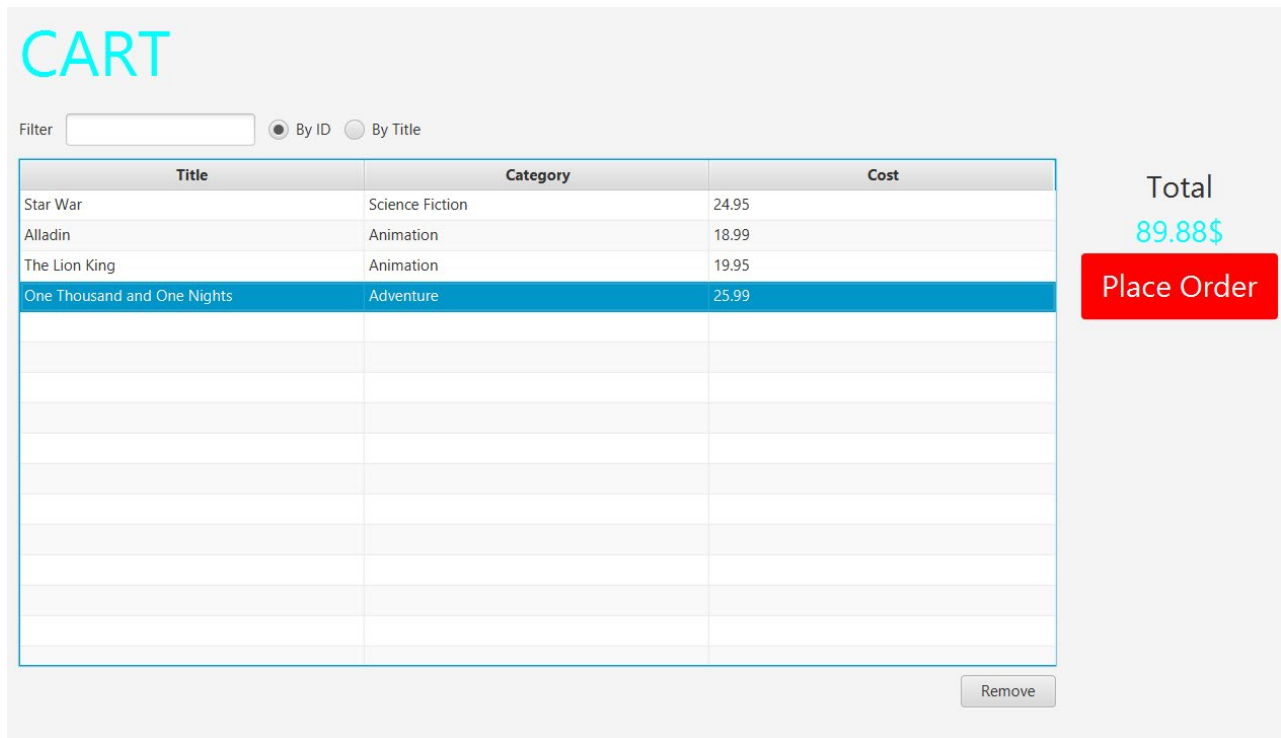


Figure 6.4: Demo media unplayable

7 Deleting a media

7.1 Code

```
@FXML
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia_hacd(media);
    costLabel.setText(cart.totalCost_hacd() + " $");
}
```

Figure 7.1: btnRemovePressed Method

7.2 Demo

CART

Filter ☒ By ID ☐ By Title

Title	Category	Cost
Star War	Science Fiction	24.95
Alladin	Animation	18.99
The Lion King	Animation	19.95
One Thousand and One Nights	Adventure	25.99

Play Remove

Total
89.88\$

Place Order

Figure 7.2: button Remove

CART

Filter ☒ By ID ☐ By Title

Title	Category	Cost
Star War	Science Fiction	24.95
Alladin	Animation	18.99
One Thousand and One Nights	Adventure	25.99

Play Remove

Total
69.93 \$

Place Order

Figure 7.3: button Remove

8 Complete the Aims GUI application

Options

AIMS

[View cart](#)

<p>The Lion King 19.95\$</p> <p>Add to cart Play</p>	<p>Star War 24.95\$</p> <p>Add to cart Play</p>	<p>Alladin 18.99\$</p> <p>Add to cart Play</p>
<p>Sherlock Holmes 21.89\$</p> <p>Add to cart</p>	<p>One Thousand and One Nights 25.99\$</p> <p>Add to cart</p>	<p>Robin Hood 18.89\$</p> <p>Add to cart</p>
<p>Tobu Best Songs Collection 45.99\$</p> <p>Add to cart Play</p>	<p>Lé Miserablé 35.99\$</p> <p>Add to cart Play</p>	<p>Katie Perrie 63.99\$</p> <p>Add to cart Play</p>

Figure 8.1: Store before add book

ADD BOOK

Title

Category

Cost

[Save](#)

Figure 8.2: Add book

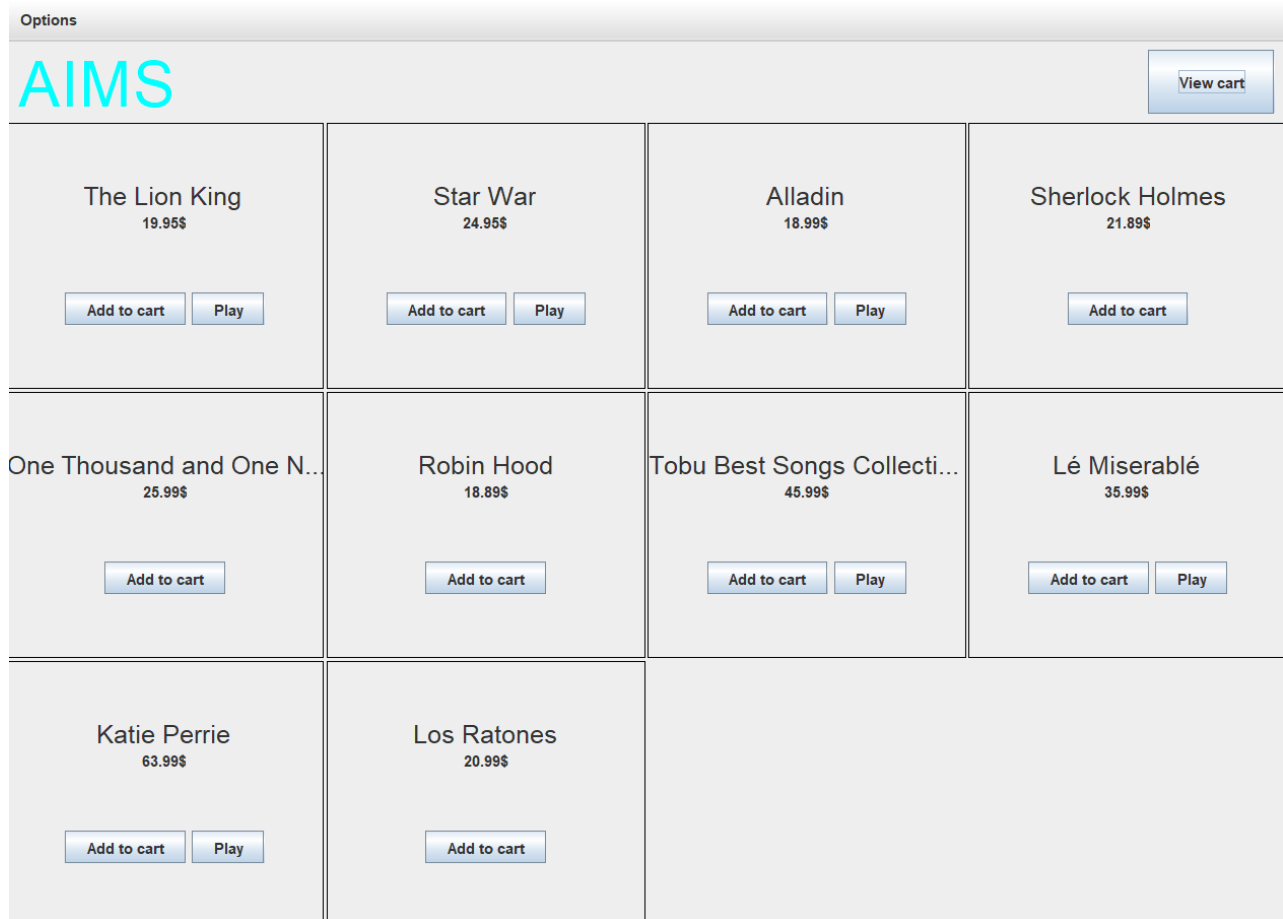


Figure 8.3: Store after add book

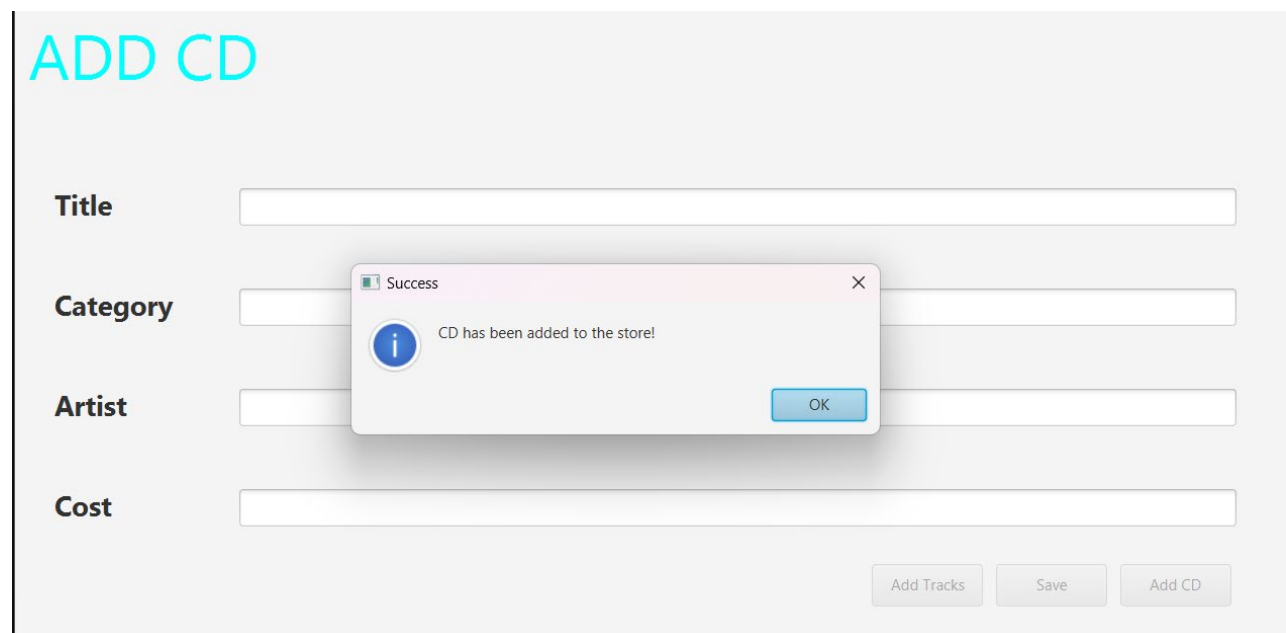


Figure 8.4: Add CD

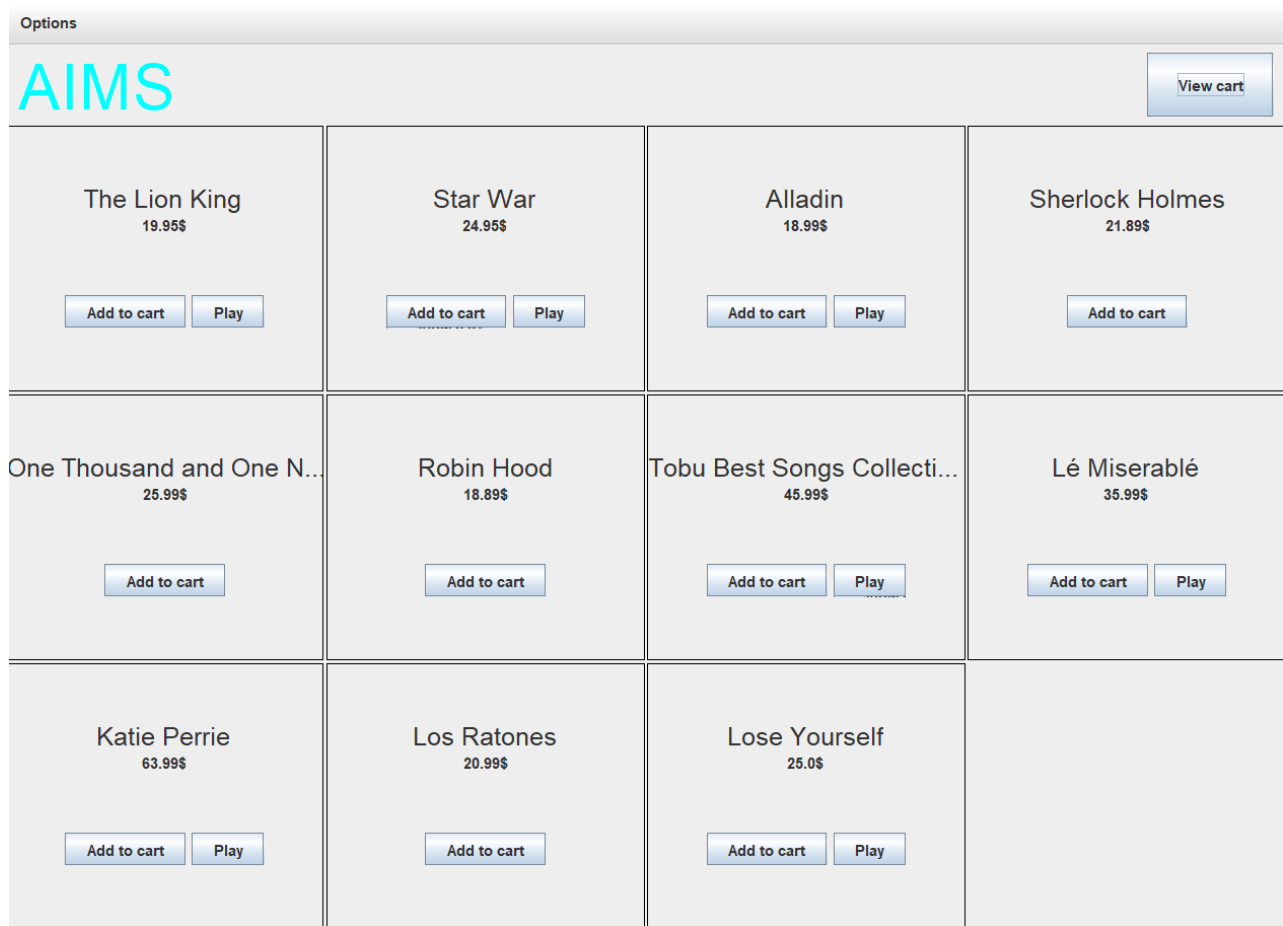


Figure 8.5: Store after add CD

ADD DVD

Title


Category

Director

Length

Cost

Success

 DVD has been added to the store!

OK

Figure 8.6 Add DVD

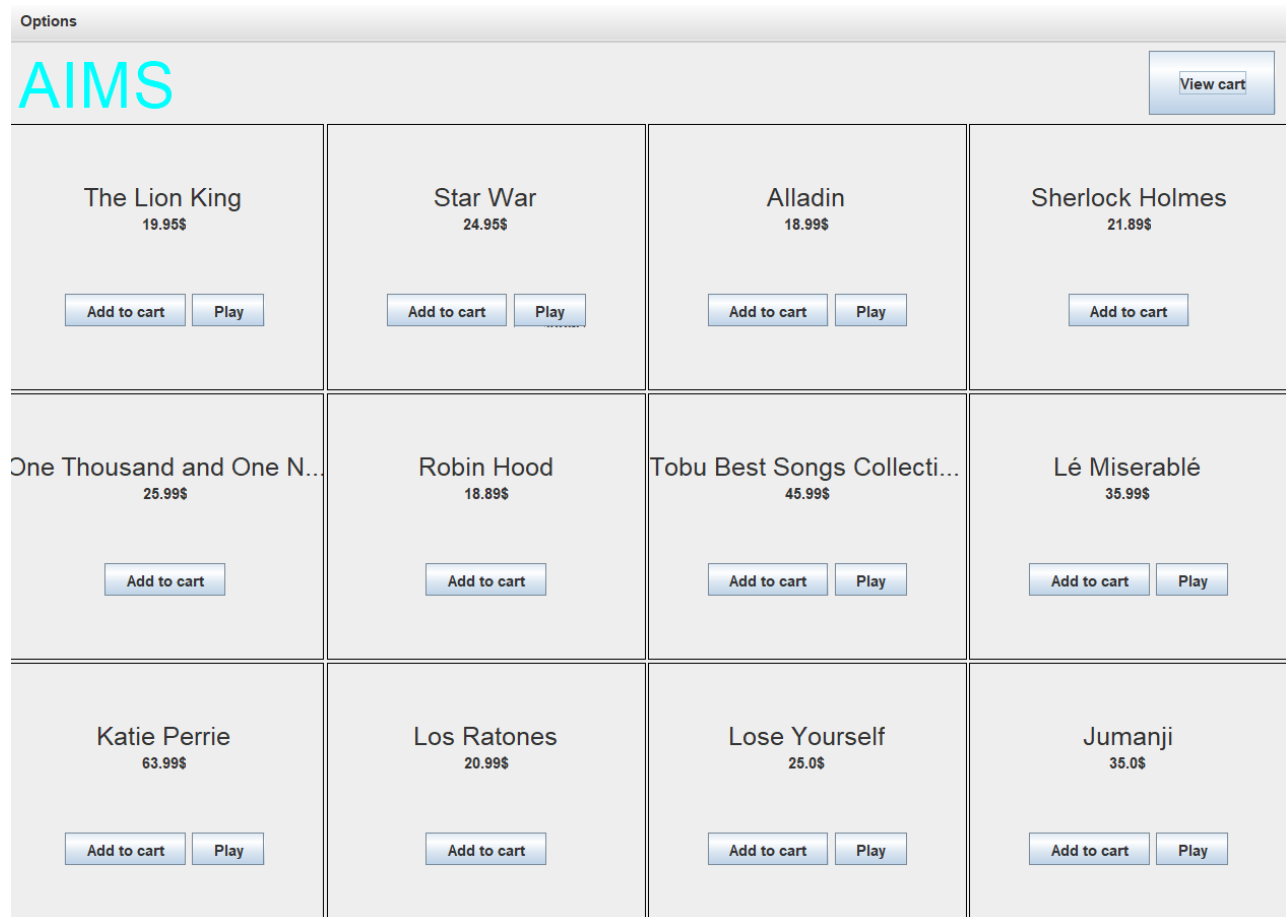


Figure 8.7: Store after add DVD

CART

Filter ☒ By ID ☐ By Title

Title	Category	Cost
Star War	Science Fiction	24.95
Alladin	Animation	18.99
The Lion King	Animation	19.95
One Thousand and One Nights	Adventure	25.99

Total
89.88\$

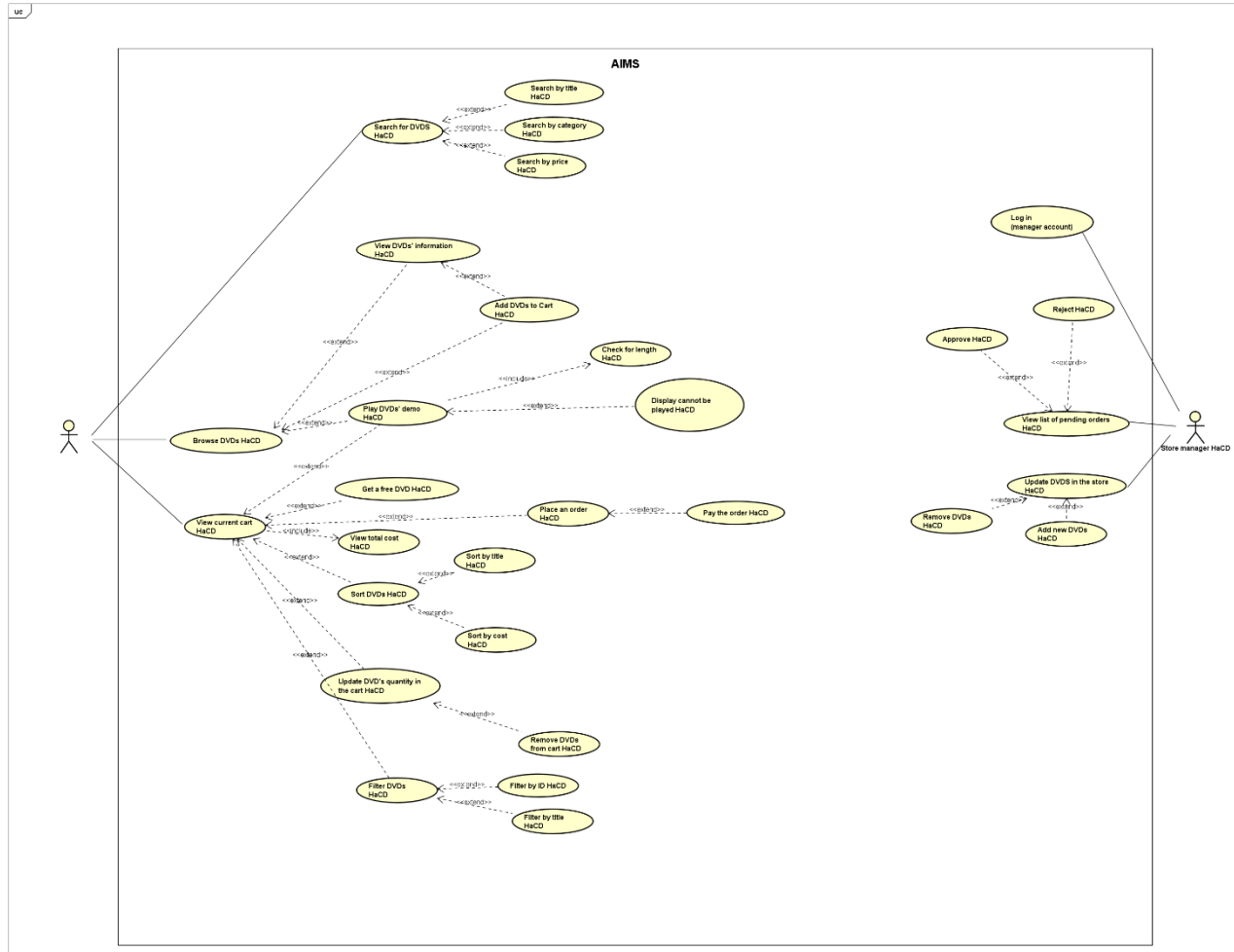
Place Order

Play

Remove

Figure 8.8: Car

9 Use case Diagram



10 Class Diagram

