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: **Hồ Tuấn Anh**
Lớp: **K67-ITEP01**
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

## 1.1 AWTAccumulator

```java
package hust.soict.itep.swing;

import java.awt.Label;
import java.awt.GridLayout;
import java.awt.TextField;
import java.awt.Frame;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class AWTAccumulator extends Frame {

    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;

    public AWTAccumulator() {
        setLayout(new GridLayout(2,2));
        add(new Label("HoTuanAnh_20226100 Enter an integer: "));

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

        add(new Label("HoTuanAnh_20226100 The Accumulated Sum is: "));

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

        setTitle("HoTuanAnh_20226100 AWT Accumulator");
        setSize(350, 120);
        setVisible(true);
    }

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

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

*Figure 1.1: Source code of AWTAccumulator*

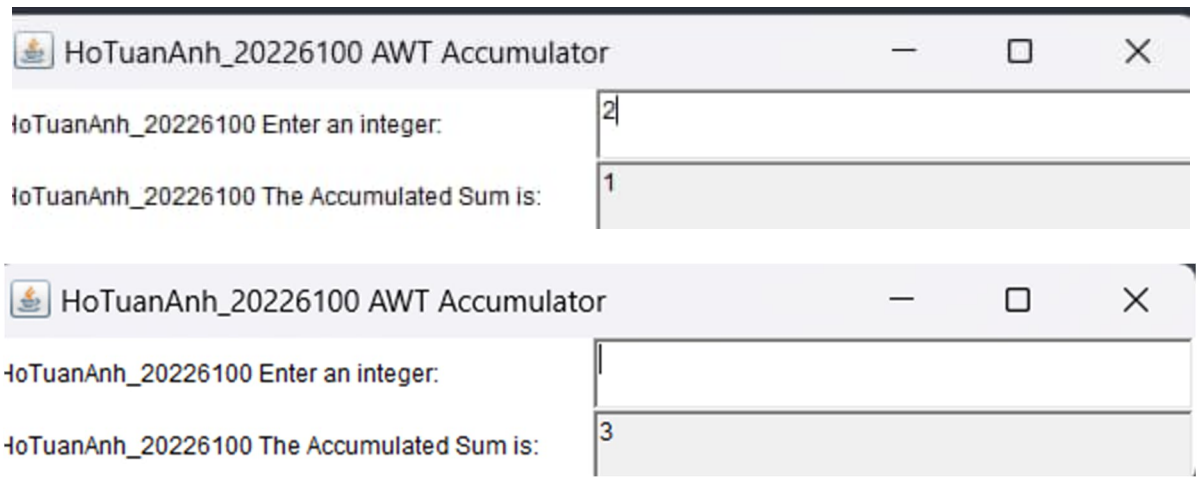
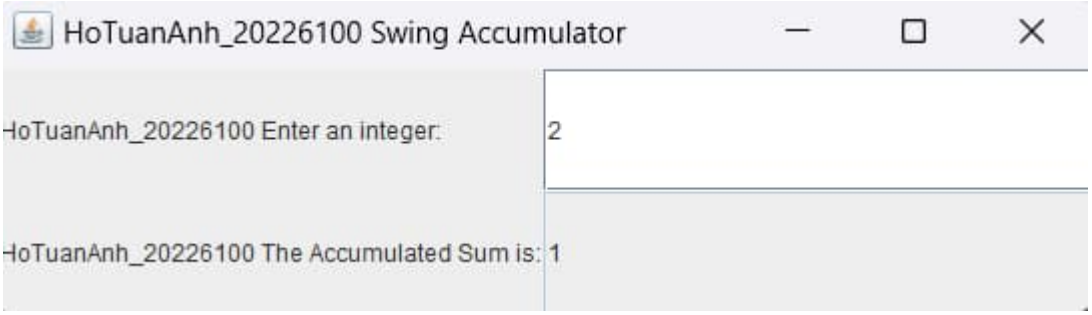



*Figure 1.2: Demo of AWTAccumulator*

## 1.2 SwingAccumulator

```java
package hust.soict.itep.swing;

import javax.swing.JFrame;
import javax.swing.JTextField;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SwingAccumulator extends JFrame {
    private JTextField tfInput;
    private JTextField tfOutput;
    private int sum = 0;
    public SwingAccumulator() {
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(2,2));

        cp.add(new Label("HoTuanAnh_20226100 Enter an integer: "));
        tfInput = new JTextField(10);
        cp.add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        cp.add( new Label("HoTuanAnh_20226100 The Accumulated Sum is: "));
        tfOutput = new JTextField(10);
        tfOutput.setEditable(false);
        cp.add(tfOutput);

        setTitle("HoTuanAnh_20226100 Swing Accumulator");
        setSize(350, 120);
        setVisible(true);
    }

    public static void main(String[] args){
        new SwingAccumulator();
    }

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

*Figure 1.3: Source code of SwingAccumulator*



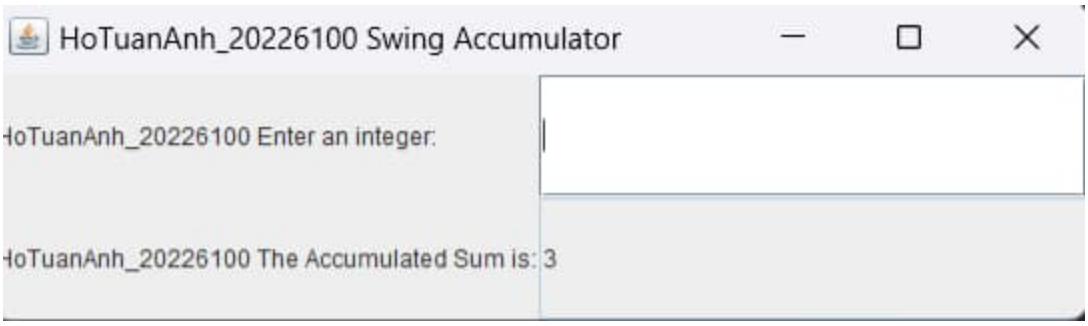



*Figure 1.4: Demo of SwingAccumulator*

# 2 Organizing Swing components with Layout Managers

## 2.1 Code

```java
package hust.soict.itep.swing;

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

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("HoTuanAnh_20226100 Number Grid");
        setSize(200,200);
        setVisible(true);
    }

    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("DEL");
        panelButtons.add(btnDelete);
        btnDelete.addActionListener(btnListener);

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

        btnReset = new JButton("C");
        panelButtons.add(btnReset);
        btnReset.addActionListener(btnListener);

    }
```

*Figure 2.1: Source code of NumberGrid 1*

```java
package hust.soict.itep.swing;

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

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("HoTuanAnh_20226100 Number Grid");
        setSize(200,200);
        setVisible(true);
    }

    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("DEL");
        panelButtons.add(btnDelete);
        btnDelete.addActionListener(btnListener);

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

        btnReset = new JButton("C");
        panelButtons.add(btnReset);
        btnReset.addActionListener(btnListener);

    }
```

*Figure 2.2: Source code of NumberGrid 2*

## 2.2 Demo


*Figure 2.3: Demo buttons 0-9*

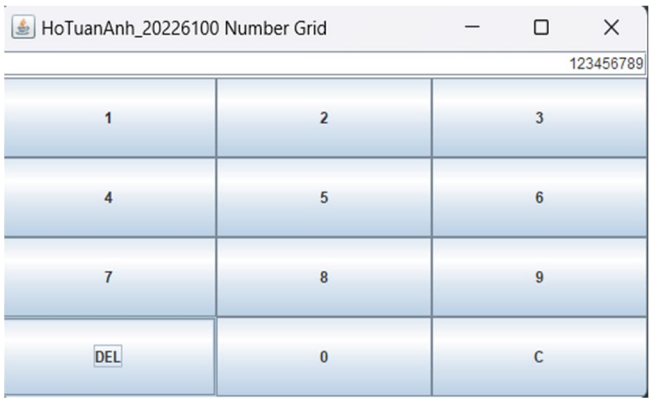



*Figure 2.4: Demo DEL button*

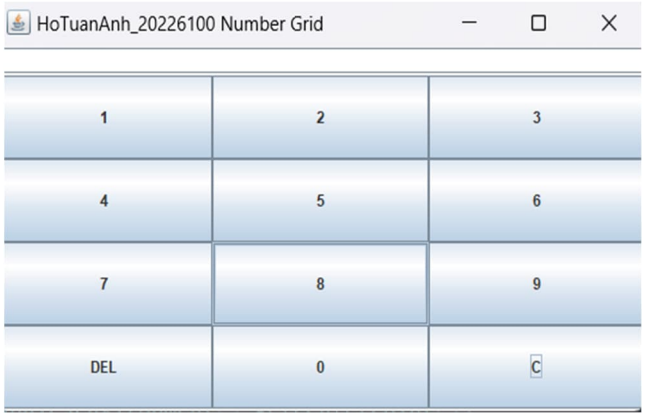



*Figure 2.5: Demo C button*

# 3 Create a graphical user interface for AIMS with Swing

## 3.1 Create class StoreScreen

```
src > hust > soict > itep > aims > screen > StoreScreen.java > StoreScreen > createHeader()
package hust.soict.itep.aims.screen;

import javax.swing.*;

import hust.soict.itep.aims.cart.Cart;
import hust.soict.itep.aims.media.Media;
import hust.soict.itep.aims.store.Store;

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


public class StoreScreen extends JFrame{

    private Store store;
    private Container cp;
    private Cart cart;

    JPanel createNorth() {
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north,BoxLayout.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);
        return menuBar;
    }
```

*Figure 3.1: Class StoreScreen 1*

```
src > hust > soict > itep > aims > screen > StoreScreen.java > StoreScreen > createCenter()
public class StoreScreen extends JFrame{

    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:10, height:10));
        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> mediaStore = store.getItemsInStore();
        for (int i = 0; i < 9; i++) {
            MediaStore cell = new MediaStore(mediaStore.get(i));
            center.add(cell);
        }
        return center;
    }

    public StoreScreen(Store store) {
        this.store  = store;
        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 3.2: Class StoreScreen 2*

## 3.2 Create class MediaStore

src > hust > soict > itep > aims > screen > MediaStore.java > MediaStore

```java
package hust.soict.itep.aims.screen;

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

import hust.soict.itep.aims.media.CompactDisc;
import hust.soict.itep.aims.media.DigitalVideoDisc;
import hust.soict.itep.aims.media.Media;
import hust.soict.itep.aims.media.Playable;
import hust.soict.itep.aims.media.Track;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;



public class MediaStore extends JPanel {

    private Media media;
```

*Figure 3.7: Class MediaStore 1*

src > hust > soict > itep > aims > screen > MediaStore.java > MediaStore > MediaStore(Media)

```java
public class MediaStore extends JPanel {
    public MediaStore(Media media) {
        JLabel title = new JLabel(media.getTitle());
        title.setFont(new Font(title.getFont().getName(),Font.PLAIN,size:20));
        title.setAlignmentX(CENTER_ALIGNMENT);

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

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

        container.add(new JButton(text:"Add to cart"));

        if(media instanceof Playable) {
            JButton playButton = new JButton(text:"Play");
            playButton.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    JDialog playDialog = createPlayDialog(media);
                    playDialog.setVisible(b:true);
                    playDialog.setSize(width:300,height:200);
                    playDialog.pack();
                }
            });
            container.add(playButton);
        }

        this.add(Box.createVerticalGlue());
        this.add(title);
        this.add(cost);
        this.add(Box.createVerticalGlue());
        this.add(container);
        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }

    private JDialog createPlayDialog(Media media) {
        JDialog playDialog = new JDialog();
        Container container = playDialog.getContentPane();
        playDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        container.setLayout(new BoxLayout(container,BoxLayout.Y_AXIS));
        container.add(Box.createRigidArea(new Dimension(width:10,height:10)));
        if(media instanceof DigitalVideoDisc dvd) {
            container.add(new JLabel("Playing DVD:" + dvd.getTitle()));
            container.add(new JLabel("DVD length:" + dvd.getLength() +" min"));
        } else if (media instanceof CompactDisc cd) {
            container.add(new JLabel("Title: " + cd.getTitle()));
            container.add(new JLabel("Artist: " + cd.getArtist()));
            for (Track track : cd.getTracks()) {
                container.add(new JLabel("Play: " + track.getTitle() + ". Length: " + track.getLength() + " min"));
            }
        }
```

*Figure 3.8: Class MediaStore 2*

## 3.3 Demo

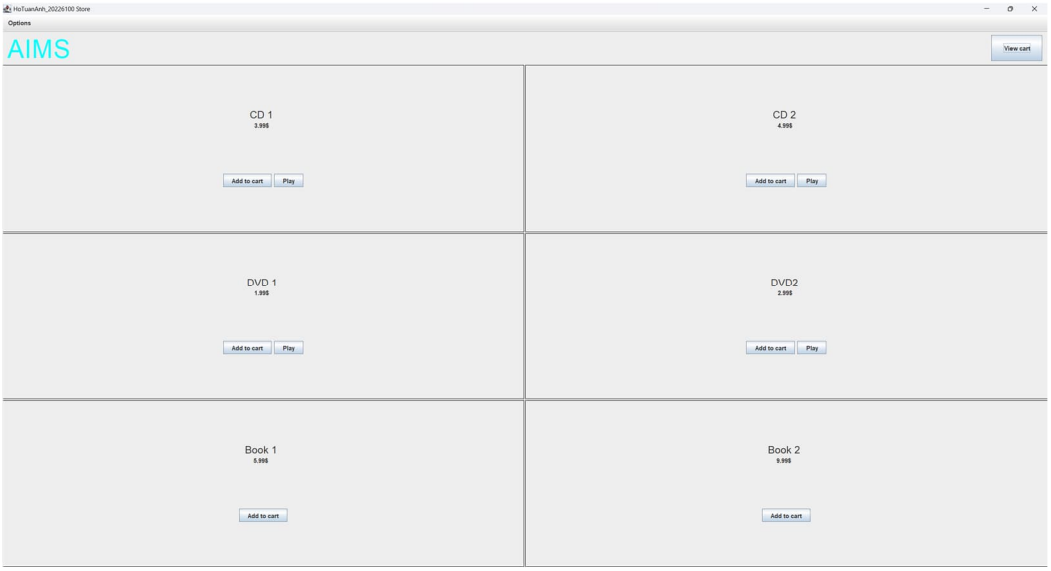



*Figure 3.10: StoreScreen*

*Figure 3.11 Demo Add to cart button*

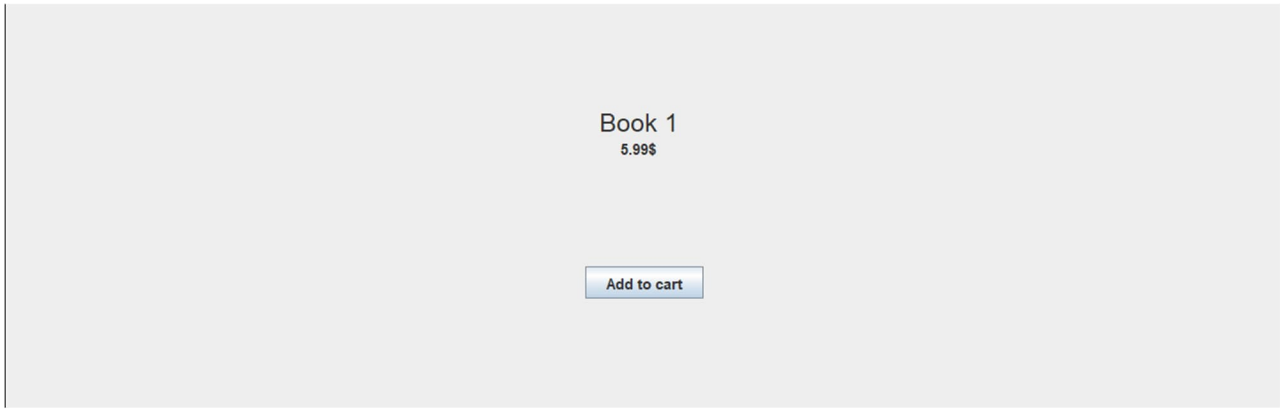
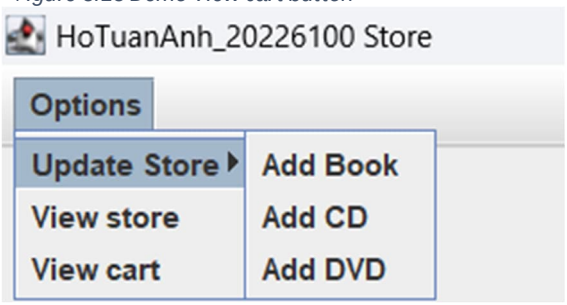
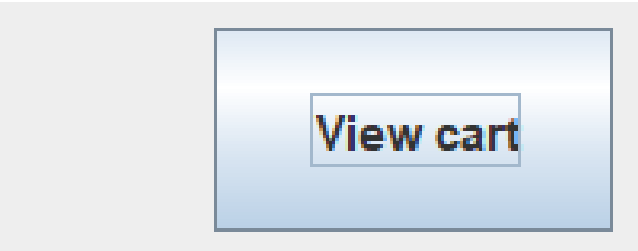



*Figure 3.12 Demo Play button*


*Figure 3.13 Demo View cart button*



# 4 JavaFX API

## 4.1 Create class Painter

```
src > hust > soict > itep > javafx > Painter.java > Painter > start(Stage)
package hust.soict.itep.javafx;

import java.util.Objects;

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 {

    @Override
    public void start(Stage stage) throws Exception {
        try {
            Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource(name:"Painter.fxml")));
            Scene scene = new Scene(root);
            stage.setTitle("Painter");
            stage.setScene(scene);
            stage.show();
        } catch (Exception e) {
            System.out.println(e.toString());
        }
    }

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

*Figure 4.1: Class Painter*

## 4.2 Create Painter.fxml

```
src > hust > soict > itep > javafx > Painter.fxml
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/23" xmlns:fx="

    <center>
        <AnchorPane id="drawingAreaPane" fx:id="drawingAreaPane" onMouseClicked="#drawAreaMouseDragged" onMouseDragged="#drawAreaMouseDragged" prefHeight="200.0" prefWidth="200.0" B
    </center>
    <left>
        <VBox prefHeight="200.0" prefWidth="100.0" style="-fx-background-color: grey;" BorderPane.alignment="CENTER">
            <children>
                <TitledPane animated="false" text="tool">
                    <content>
                        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="80.0" prefWidth="82.0">
                            <children>
                                <RadioButton fx:id="penRadioButton" layoutX="3.0" layoutY="10.0" mnemonicParsing="false" text="Pen" selected="true">
                                    <toggleGroup>
                                        <ToggleGroup fx:id="toggle" />
                                    </toggleGroup>
                                </RadioButton>
                                <RadioButton fx:id="eraserButton" layoutX="2.0" layoutY="40.0" mnemonicParsing="false" text="Eraser" toggleGroup="$toggle" />
                            </children>
                        </AnchorPane>
                    </content>
                </TitledPane>
                <Button fx:id="clearButton" maxWidth="50000" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear" />

            </children>
            <padding>
                <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
            </padding>
        </VBox>
    </left>
</BorderPane>
```

*Figure 4.2: Painter.fxml 1*

## 4.3 Create class PainterController

src > hust > soict > itep > javafx > PainterController.java > PainterController

```java
package hust.soict.itep.javafx;

import java.util.Iterator;
import javafx.scene.Node;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.RadioButton;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;

public class PainterController {

    @FXML
    private RadioButton penRadioButton;

    @FXML
    private RadioButton eraserButton;

    @FXML
    private AnchorPane drawingAreaPane;

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

    @FXML
    void drawAreaMouseDragged(MouseEvent event) {
        if (penRadioButton.isSelected()) {
            Circle circle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
            drawingAreaPane.getChildren().add(circle);
        } else if (eraserButton.isSelected()) {
            Iterator<Node> iter = drawingAreaPane.getChildren().iterator();
            while (iter.hasNext()) {
                Node c = iter.next();
                if (c instanceof Circle circle) {
                    if (circle.getCenterX() <= event.getX() + 4 && circle.getCenterX() >= event.getX() - 4) {
                        if (circle.getCenterY() <= event.getY() + 4 && circle.getCenterY() >= event.getY() - 4) iter.remove();
                    }
                }
            }
        }
    }



}
```

*Figure 4.4: PainterController*

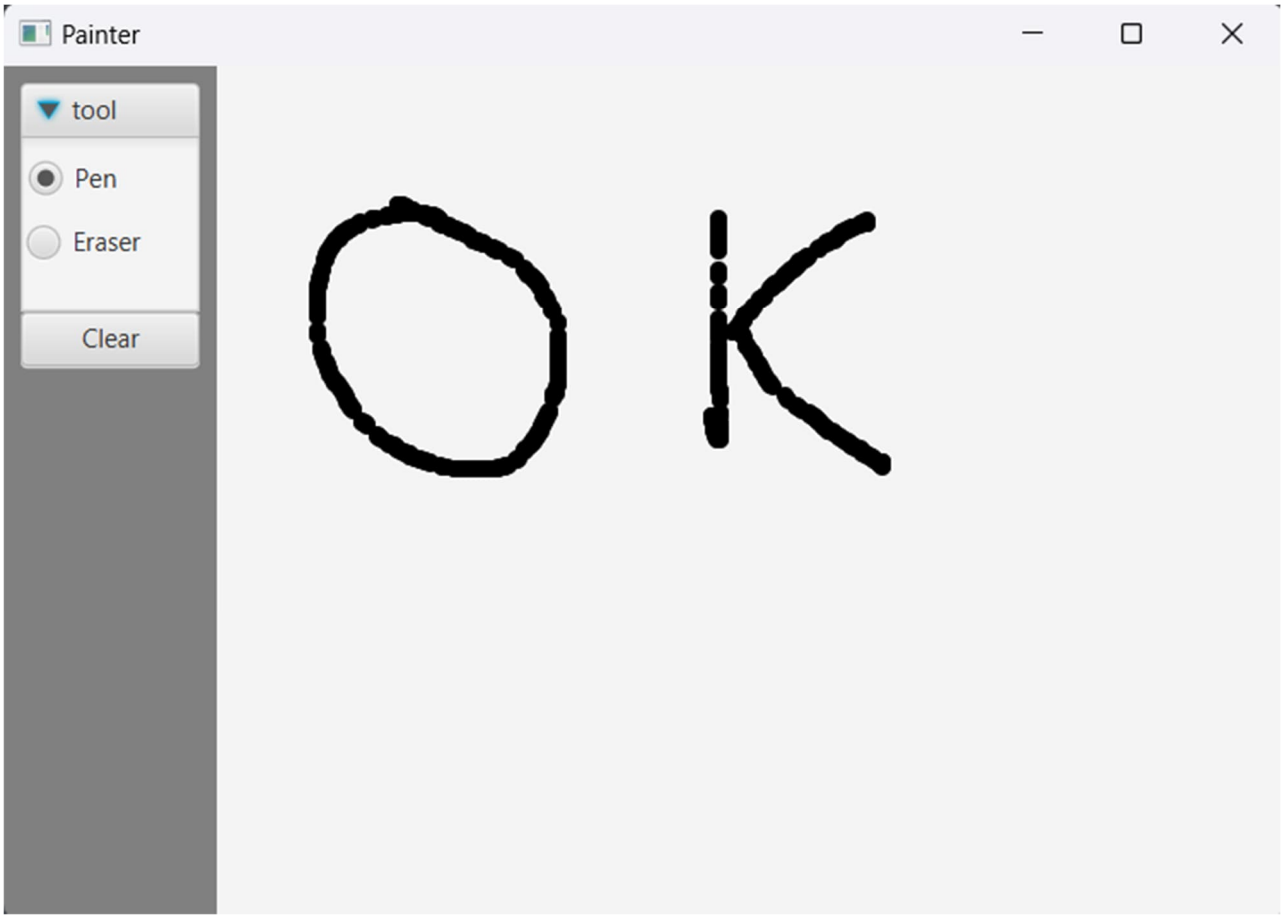



*Figure 4.5: Use Pen*

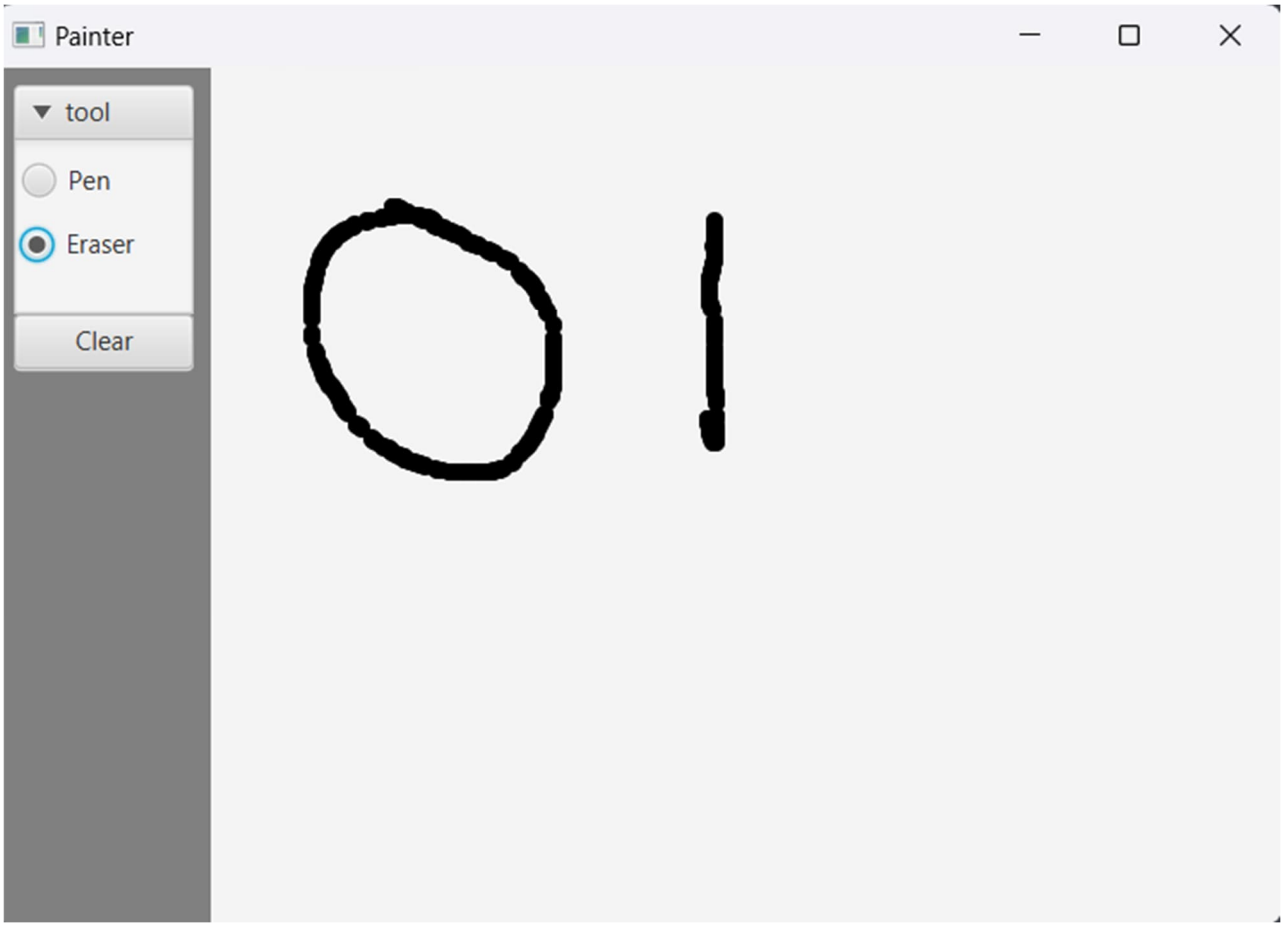



*Figure 4.6: Use Eraser*

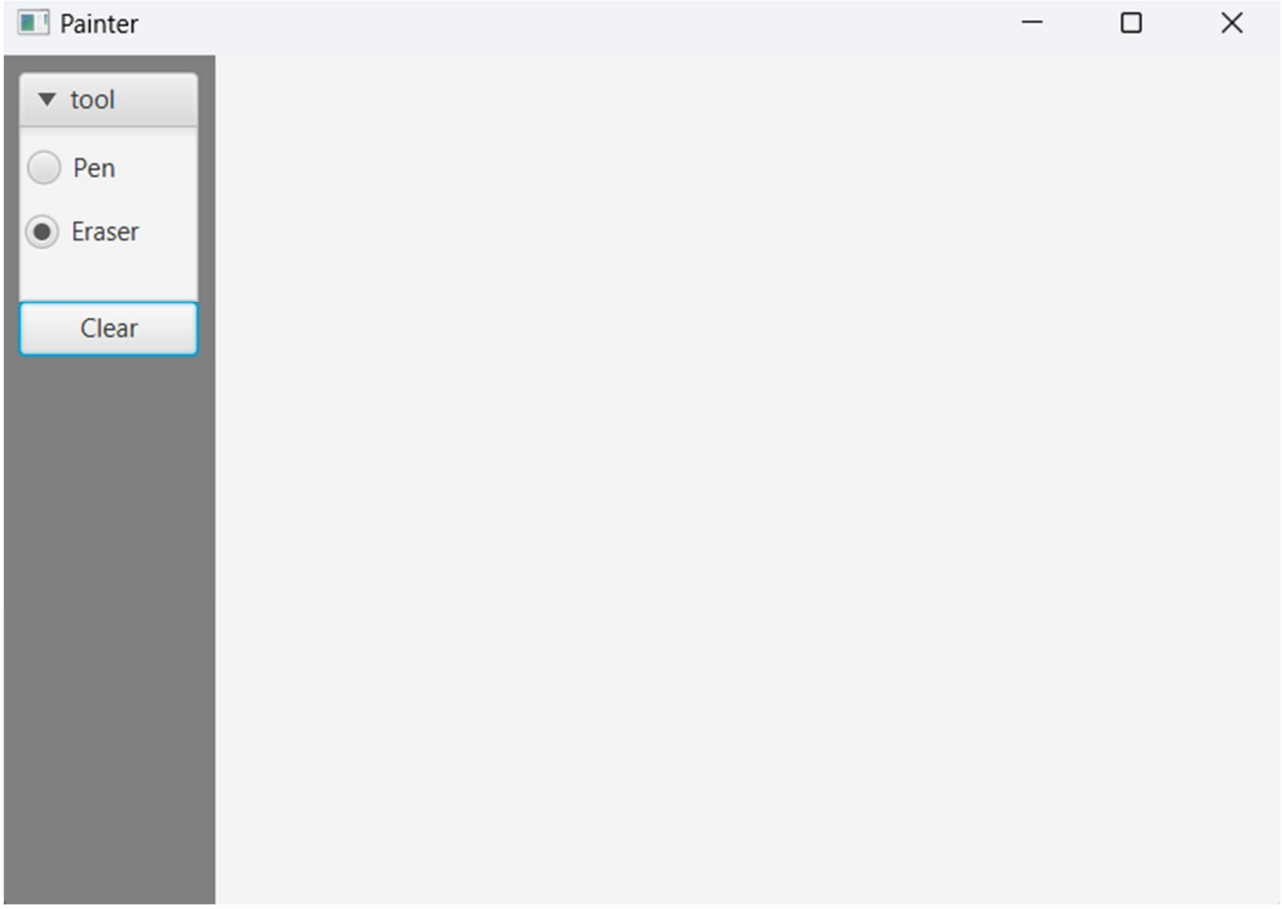



*Figure 4.7: Clear button*

# 5 View Cart Screen

## 5.1 Create cart.fxml

```
src > hust > soict > itep > aims > screen > fxml > cart.fxml
You, 1 second ago | 1 author (You)
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ButtonBar?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/23" xmlns:fx="ht
    <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 DVD" />
                                        <MenuItem mnemonicParsing="false" text="Add CD" />
                                    </items>
                                </Menu>D      You, 1 second ago • Uncommitted changes
                                <Menu mn abc DVD
                                </Menu>
                                <Menu mnemonicParsing="false" text="View cart">
                                </Menu>
                            </items>
                        </Menu>
                    </menus>
                </MenuBar>
                <Label text="CART" textFill="#000fff">
                    <font>
                        <Font size="50.0" />
                    </font>
                </Label>
            </children>
        </VBox>
    </top>
```

*Figure 5.1: Cart.fxml 1*

```
        </VBox>
    </top>
    <center>
        <VBox 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>
                        <Label text="Filter" />
                        <TextField fx:id="tfFilter" />
                        <RadioButton fx:id="radioBtnFilterId" mnemonicParsing="false" selected="true" text="By ID">
                            <toggleGroup>
                                <ToggleGroup fx:id="filterCategory" />
                            </toggleGroup>
                        </RadioButton>
                        <RadioButton fx:id="radioBtnFilterTitle" mnemonicParsing="false" text="By Title" toggleGroup="$filterCategory" />
                    </children>
                </HBox>
                <TableView fx:id="tblMedia" prefHeight="200.0" prefWidth="200.0">
                    <columns>
                        <TableColumn fx:id="colMediaTitle" prefWidth="75.0" text="Title" />
                        <TableColumn fx:id="colMediacategory" prefWidth="75.0" text="Category" />
                        <TableColumn fx:id="colMediaCost" prefWidth="75.0" text="Cost" />
                    </columns>
                    <columnResizePolicy>
                        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
                    </columnResizePolicy>
                </TableView>
                <ButtonBar prefHeight="40.0" prefWidth="200.0">
                    <buttons>
                        <Button fx:id="btnPlay" mnemonicParsing="false" text="Play" />
                        <Button fx:id="btnRemove" mnemonicParsing="false" text="Remove" />
                    </buttons>
                </ButtonBar>
            </children>
        </VBox>
    </center>
```

*Figure 5.2: Cart.fxml 2*

```
    </center>
    <right>
        <VBox alignment="TOP_CENTER" BorderPane.alignment="CENTER">
            <padding>
                <Insets left="10.0" right="10.0" top="50.0" />
            </padding>
            <children>
                <HBox alignment="CENTER">
                    <children>
                        <Label text="Total:">
                            <font>
                                <Font size="24.0" />
                            </font>
                            <padding>
                                <Insets right="10.0" />
                            </padding>
                        </Label>
                        <Label fx:id="totalPrice" text="0 $" textFill="#000fff">
                            <font>
                                <Font size="24.0" />
                            </font>
                        </Label>
                    </children>      You, 8 hours ago • update lab5
                </HBox>
                <Button fx:id="btnPlaceOrder" mnemonicParsing="false" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">
                    <font>
                        <Font size="24.0" />
                    </font>
                </Button>
            </children>
        </VBox>
    </right>
</BorderPane>
```

*Figure 5.3: Cart.fxml 3*

## 5.2 Create class CartScreen

```
src > hust > soict > itep > aims > screen > CartScreen.java > CartScreen > cart
You, 9 hours ago | 1 author (You)
package hust.soict.itep.aims.screen;

import java.io.IOException;

import javax.swing.JFrame;

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

You, 9 hours ago | 1 author (You)
public class CartScreen extends JFrame {

    private Cart cart;      You, 9 hours ago • update lab5

    public CartScreen(Cart cart) {
        super();
        this.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle(title:"HoTuanAnh_20226100 Cart");
        this.setVisible(b:true);
        setSize(width:1100, height:768);

        You, 9 hours ago | 1 author (You)
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"../screen/fxml/cart.fxml"));
                    CartScreenController controller = new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (IOException e) {
                    System.err.println(e.getMessage());
                }
            }
        });
    }
}
```

*Figure 5.4: CartScreen class*

## 5.3 Create class CartScreenController

src > hust > soict > itep > aims > screen > CartScreenController.java > CartScreenController > showFilteredMedia(String)

```java
public class CartScreenController {

    private Cart cart;

    @FXML
    private Button btnPlaceOrder;
    @FXML
    private TextField tfFilter;
    @FXML
    private ToggleGroup filterCategory;
    @FXML
    private RadioButton radioBtnFilterId;
    @FXML
    private RadioButton radioBtnFilterTitle;
    @FXML
    private Button btnPlay;
    @FXML
    private Button btnRemove;
    @FXML
    private TableView<Media> tblMedia;
    @FXML
    private TableColumn<Media, String> colMediaTitle;
    @FXML
    private TableColumn<Media, String> colMediacategory;
    @FXML
    private TableColumn<Media,Float> colMediaCost;
    @FXML
    private Label totalPrice;


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

    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
        colMediacategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));
        tblMedia.setItems(FXCollections.observableList(this.cart.getItemsOrdered()));
        tblMedia.setPlaceholder(new Label("No item in cart"));
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);
        You, 9 hours ago | 1 author (You)
        btnRemove.setOnAction((EventHandler<ActionEvent>) new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                Media media = tblMedia.getSelectionModel().getSelectedItem();
                try {
                    cart.removeMedia(media);
                } catch (Exception e) {
                    e.printStackTrace();
```

*Figure 5.5: CartScreenController 1*

src > hust > soict > itep > aims > screen > CartScreenController.java > CartScreenController > brnRemovePressed(ActionEvent)

```java
public class CartScreenController {
    private void initialize() {
        btnPlaceOrder.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                createPopUp();
                cart.getItemsOrdered().clear();
                tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
                totalPrice.setText(Float.toString(cart.totalCost()) + "$");
            }
        });
    }

    @FXML
    void showFilteredMedia(String s) {
        if (filterCategory.getSelectedToggle() ==
            ArrayList<Media> filterByTitle = new ArrayList<Media>();
            for (Media item : cart.getItemsOrdered()) {
                if (item.getTitle().contains(s)) {
                    filterByTitle.add(item);
                }
            }
            tblMedia.setItems(FXCollections.observableList(filterByTitle));
        } else if (filterCategory.getSelectedToggle() == radioBtnFilterId) {
            ArrayList<Media> filterByID = new ArrayList<Media>();
            for (Media item : cart.getItemsOrdered()) {
                if (item.getId() == Integer.parseInt(s)) {
                    filterByID.add(item);
                }
            }
            tblMedia.setItems(FXCollections.observableList(filterByID));
        }
    }

    @FXML
    void createPopUp() {
        Stage popupwindow =new Stage();
        popupwindow.initModality(Modality.APPLICATION_MODAL);
        popupwindow.setTitle("Place order");

        Label label1 = new Label("You have place your order !");
        label1.setFont(Font.font("San Serif", FontWeight.BOLD,14));
        Label label2 = new Label("Your bill total is " + Float.toString(cart.totalCost()) + "$");
        Button button1= new Button("OK !");
        label2.setTextFill(Color.RED);
        button1.setOnAction(e -> popupwindow.close());
        VBox layout= new VBox(10);
        layout.getChildren().addAll(label1, label2,button1);
        layout.setAlignment(Pos.CENTER);
        Scene scene1= new Scene(layout, 300, 200);
        popupwindow.setScene(scene1);
        popupwindow.show();
    }
```

java.util.ArrayList.ArrayList<Media>()
Constructs an empty list with an initial capacity of ten.

*Figure 5.6: CartScreenController 2*

## 5.4 Demo

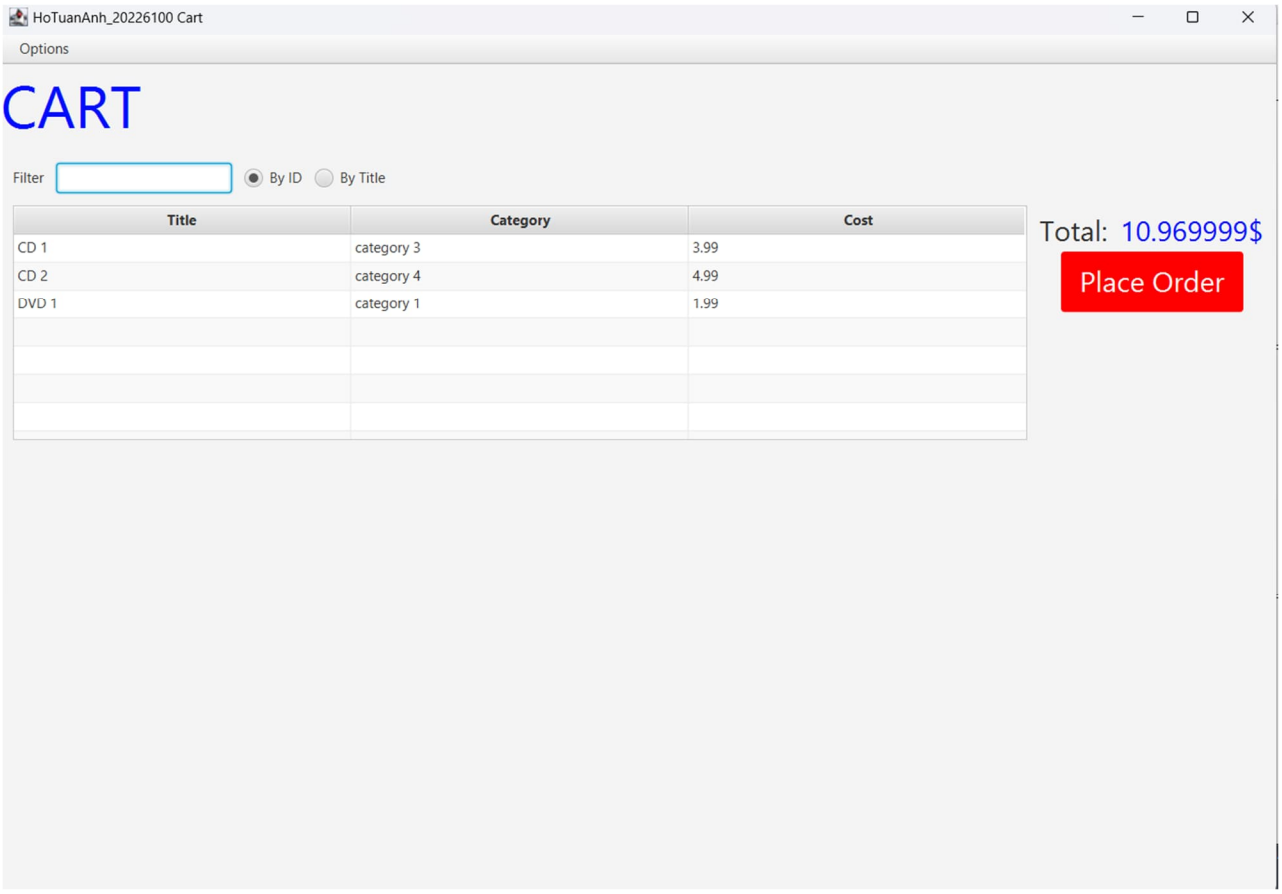



*Figure 5.7: Demo CartScreen*

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

## 6.1 Edit class CartScreenController

```
@FXML
void updateButtonBar(Media media) {
    btnRemove.setVisible(true);
    if(media instanceof Playable) {
        btnPlay.setVisible(true);
    } else {
        btnPlay.setVisible(false);
    }
}
```

*Figure 6.1: CartScreenController 1*

## 6.2 Demo

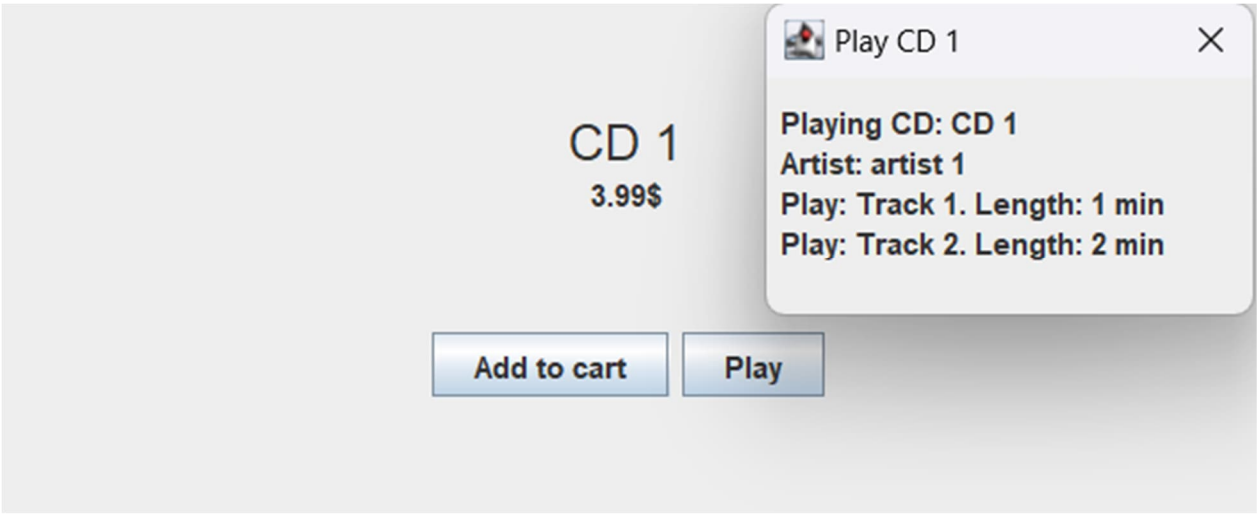



*Figure 6.3: Demo media playable*

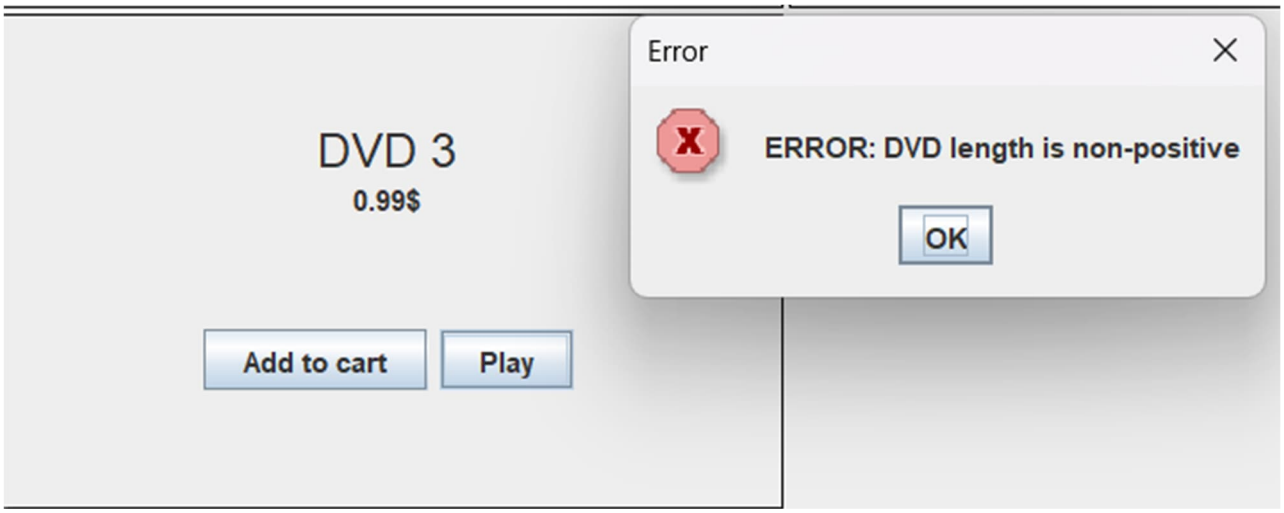


*Figure 6.4: Demo media unplayable*

# 7 Deleting a media

## 7.1 Code

```
@FXML
void brnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    try {
        cart.removeMedia(media);
    } catch (Exception e) {
        e.printStackTrace();
    }
}      You, 1 second ago • Uncommitted changes
}
```

*Figure 7.1: btnRemovePressed Method*

## 2.1 Demo


*Figure 7.2: button Rem*

## Complete the Aims GUI application

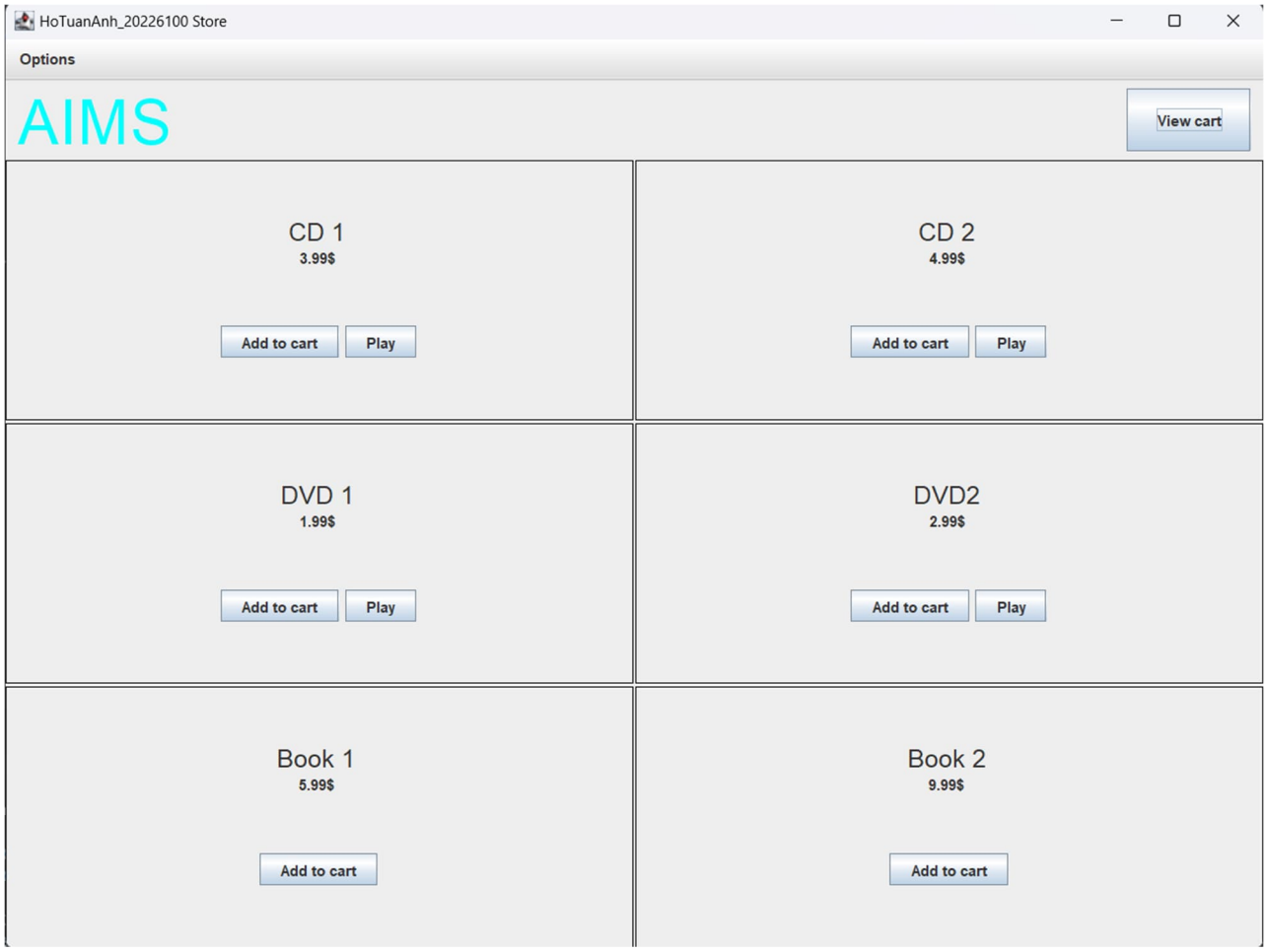



*Figure 8.1: Store before add book*

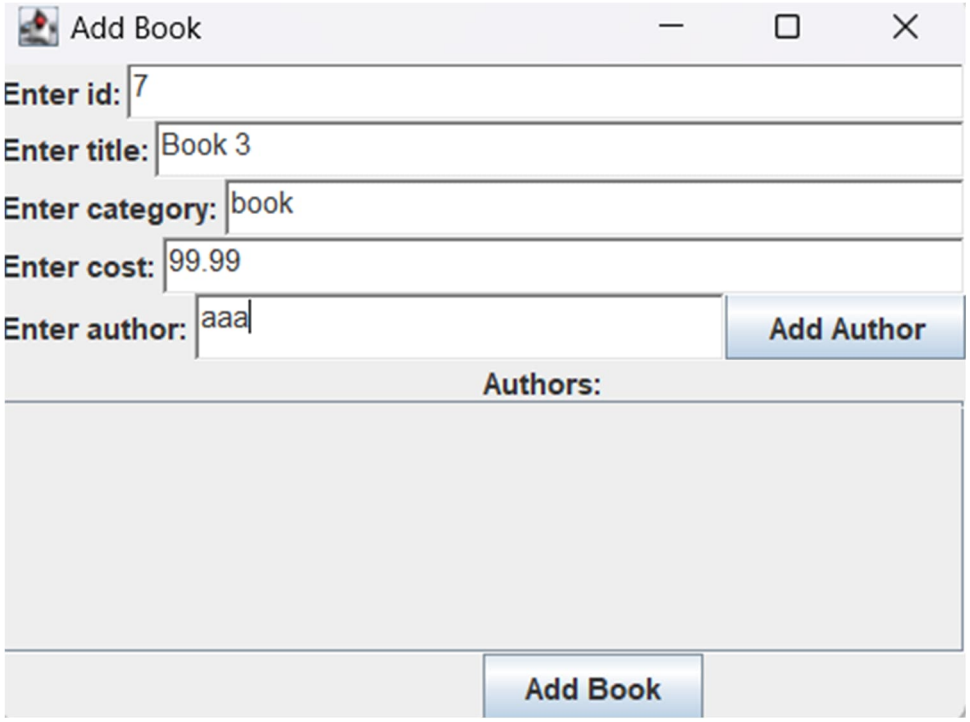



*Figure 8.2: Add book*

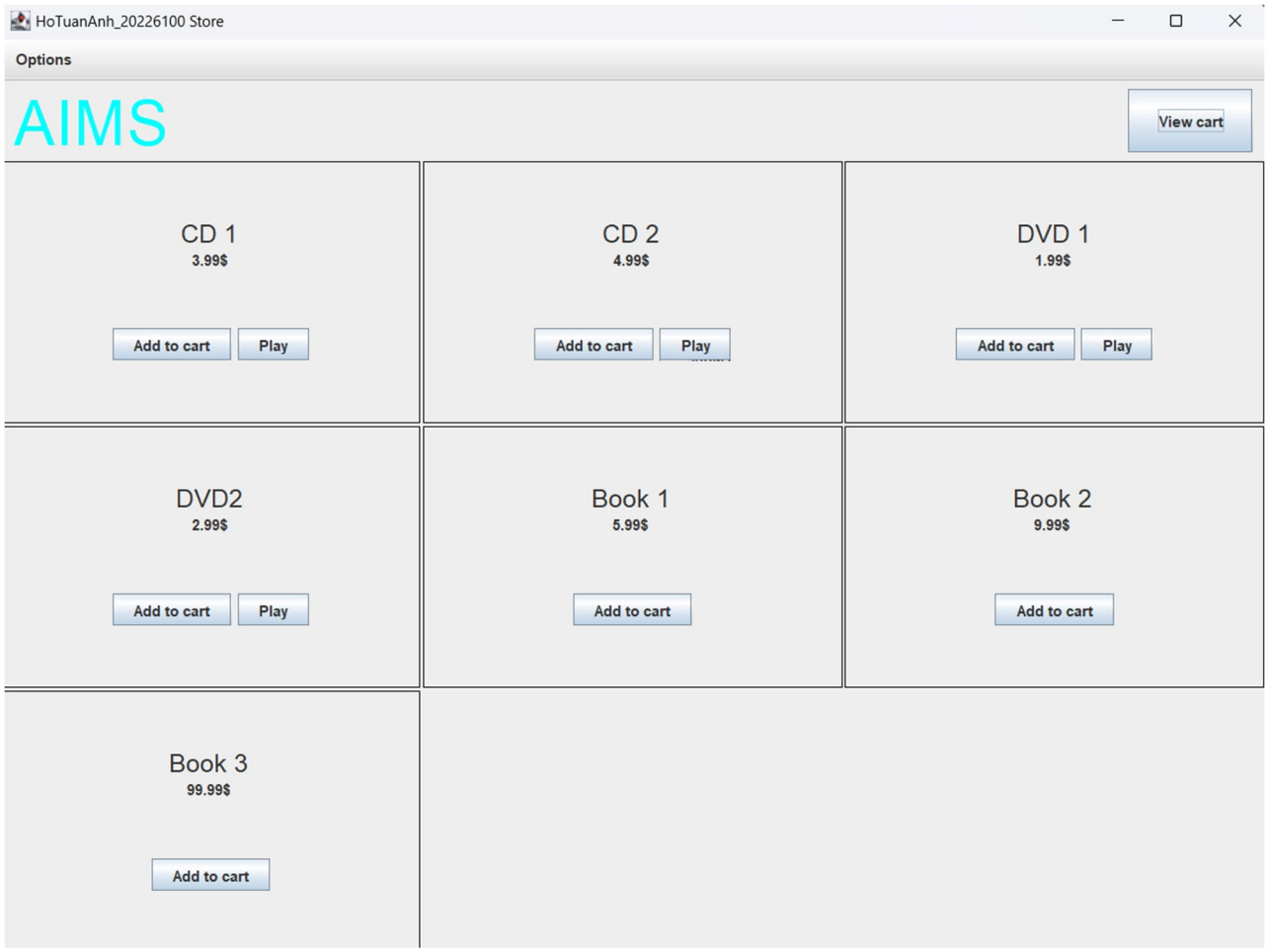



*Figure 8.3: Store after add book*

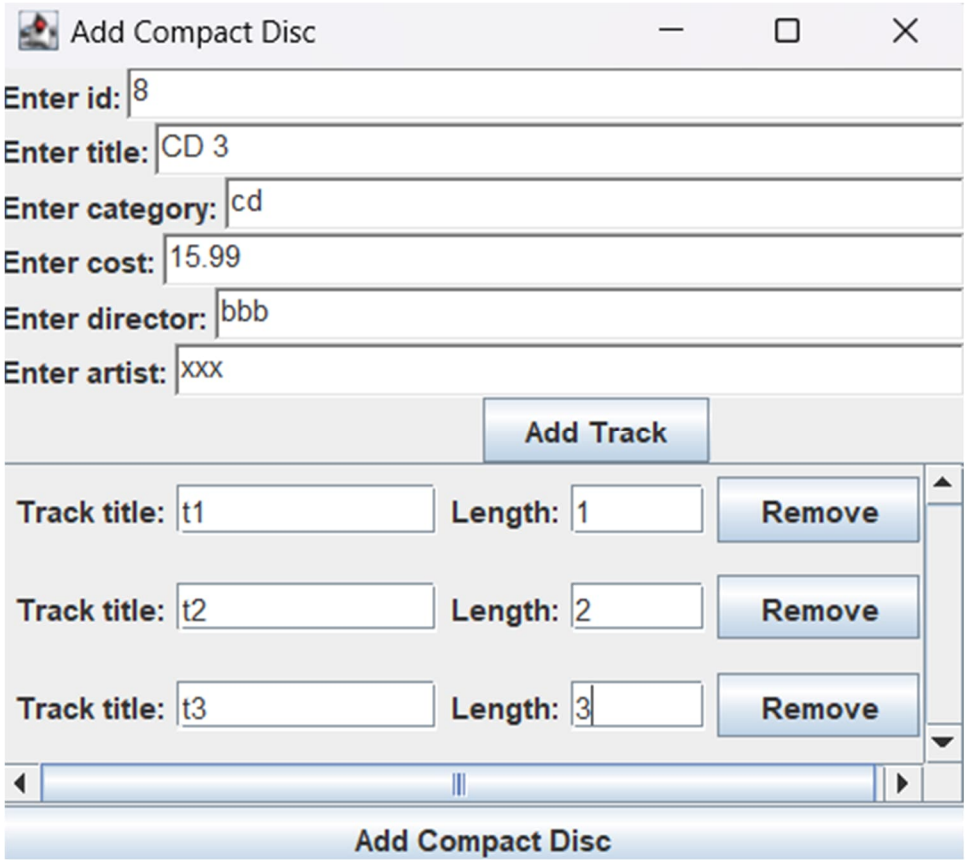



*Figure 8.4: Add CD*

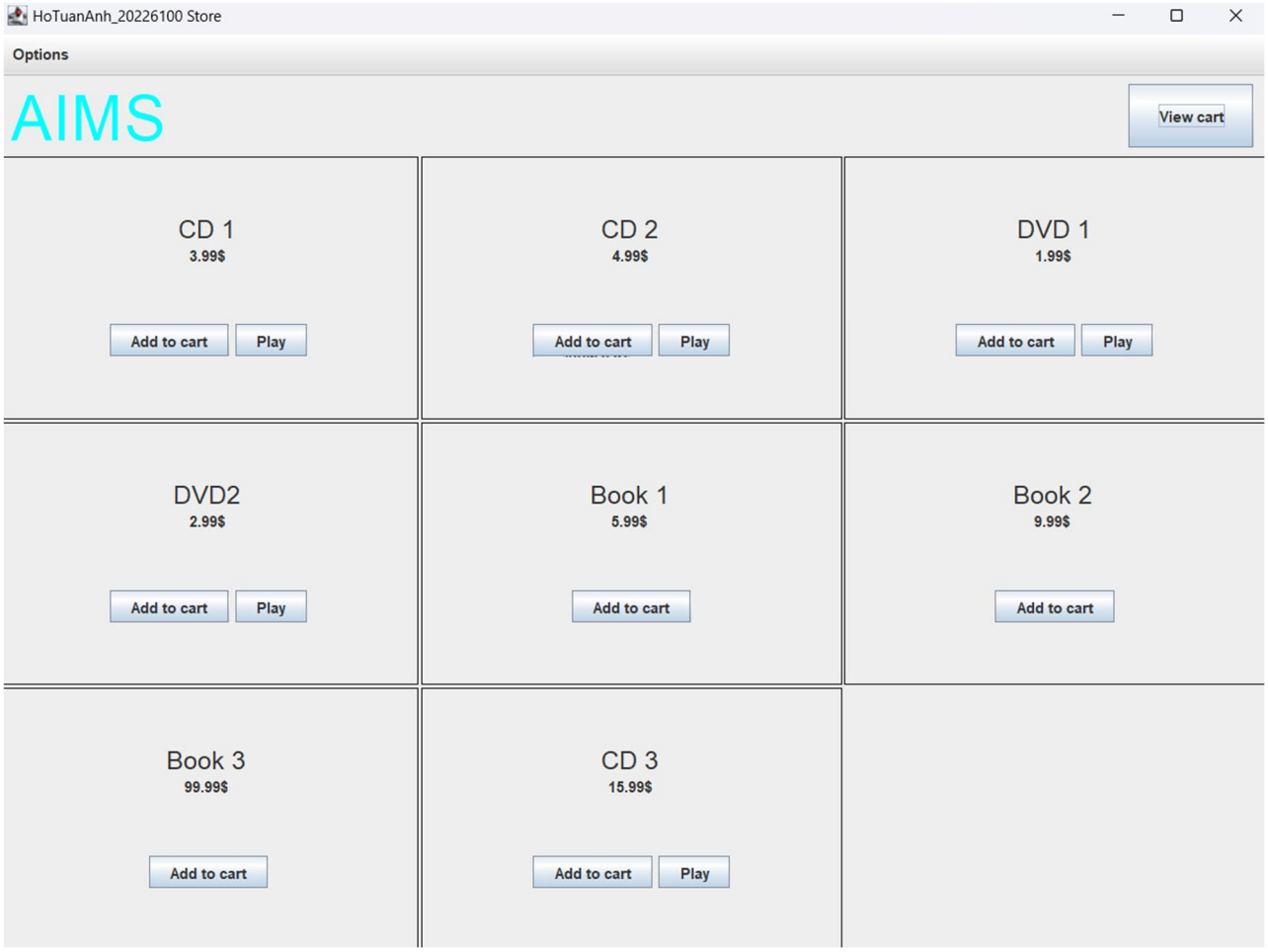



*Figure 8.5: Store after add CD*

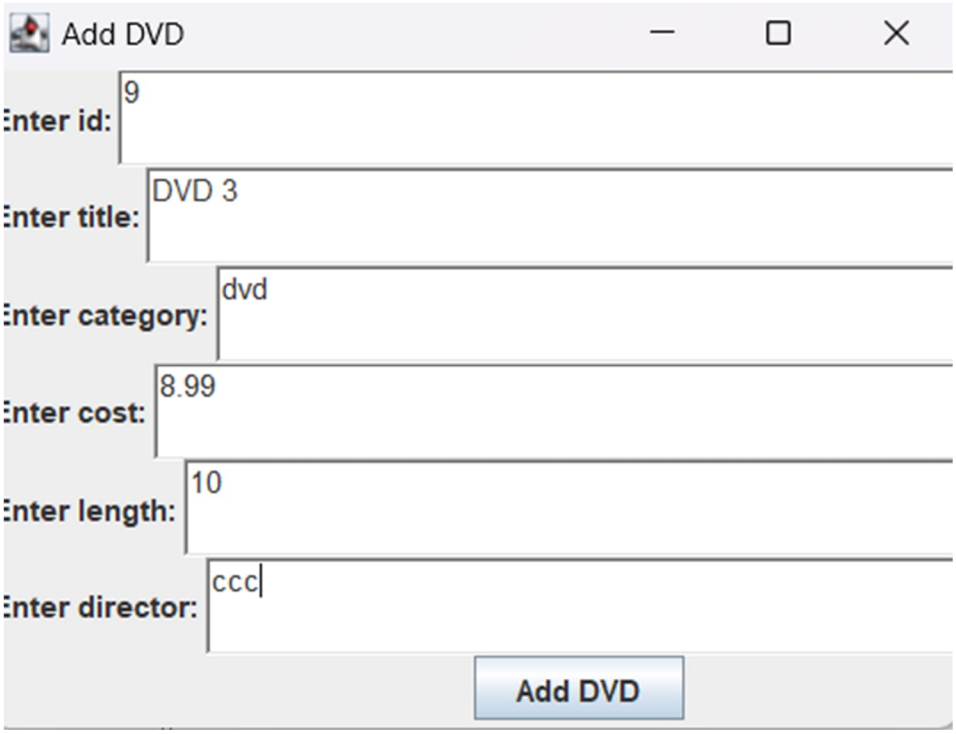



*Figure 8.6 Add DVD*

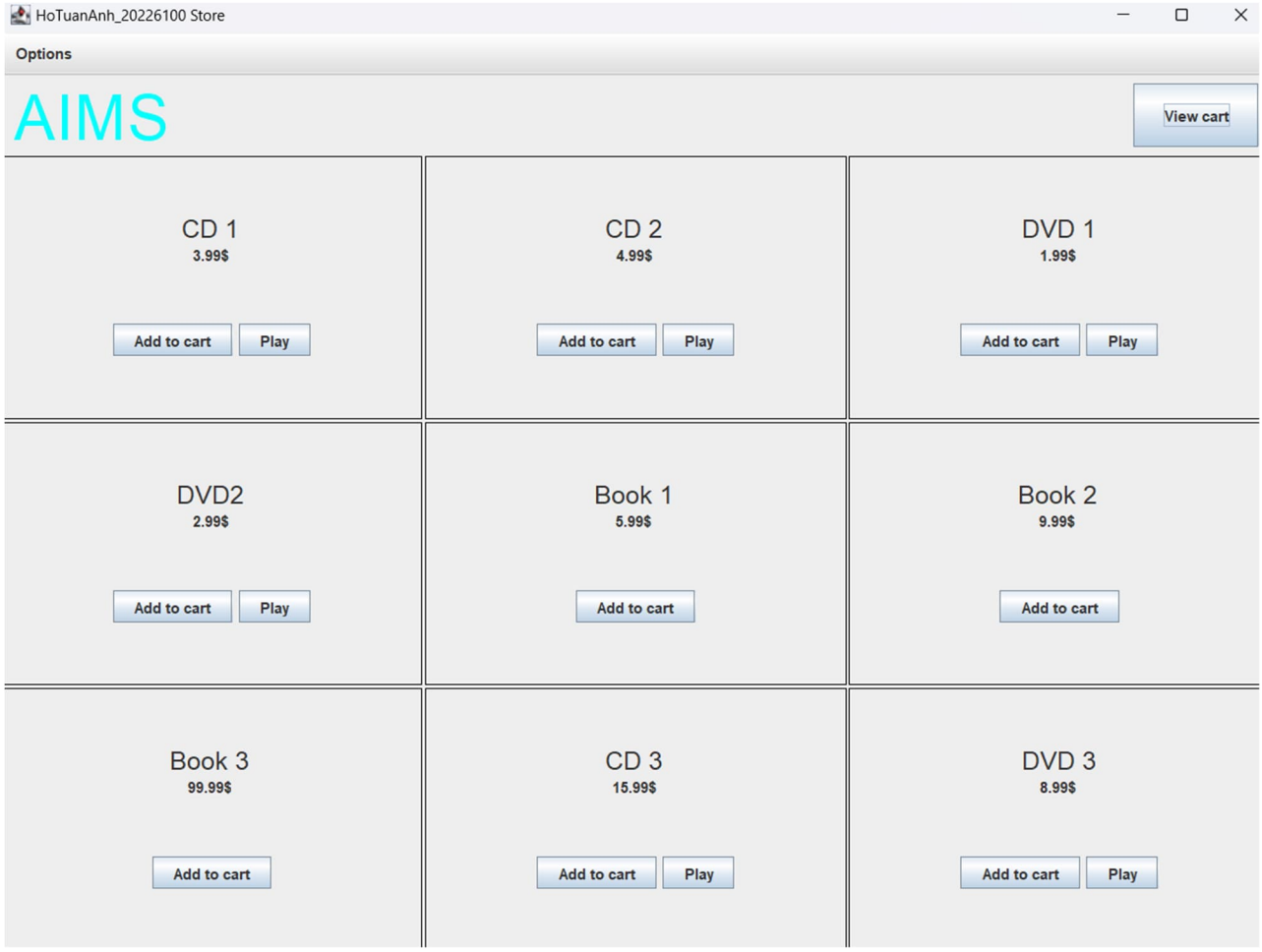



*Figure 8.7: Store after add DVD*

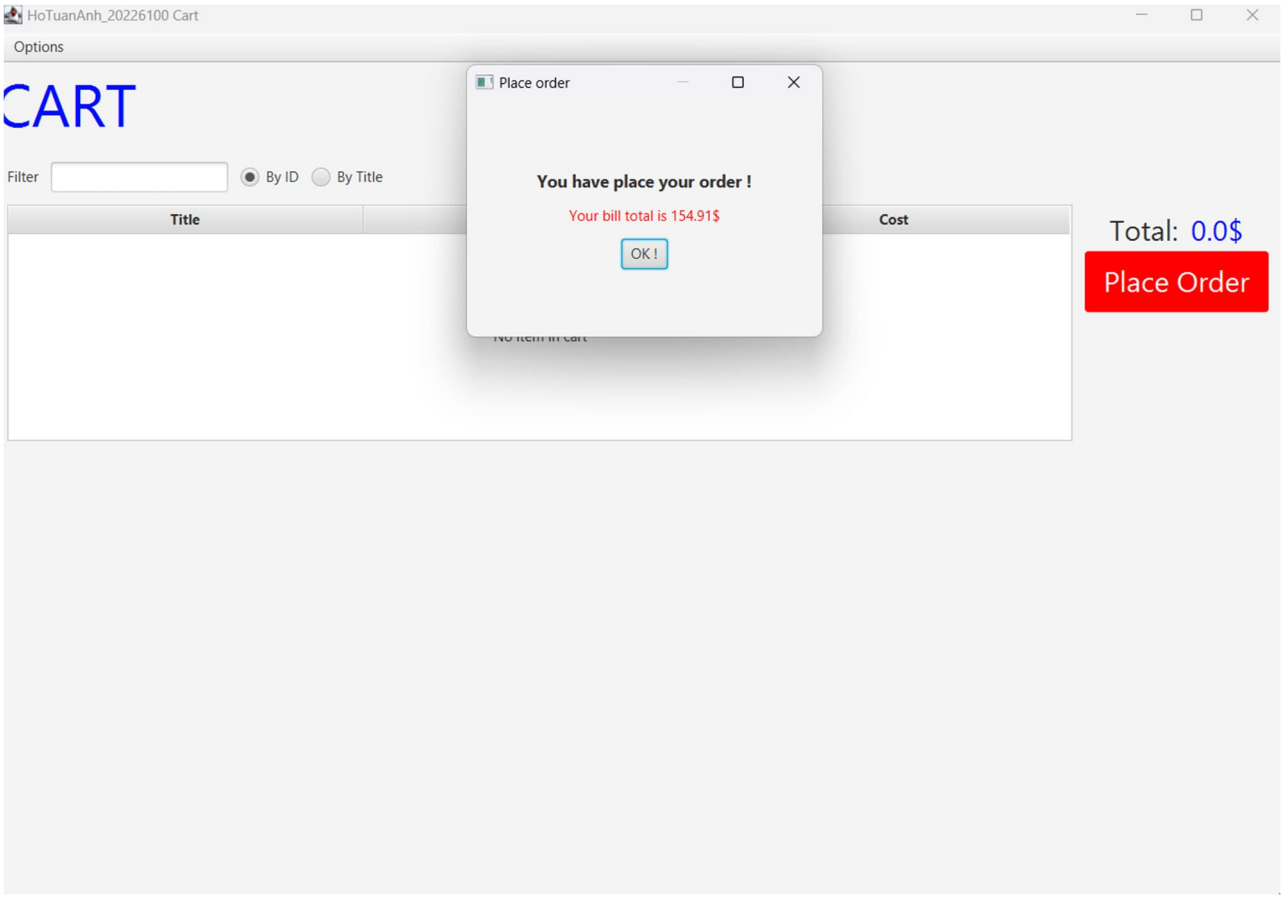
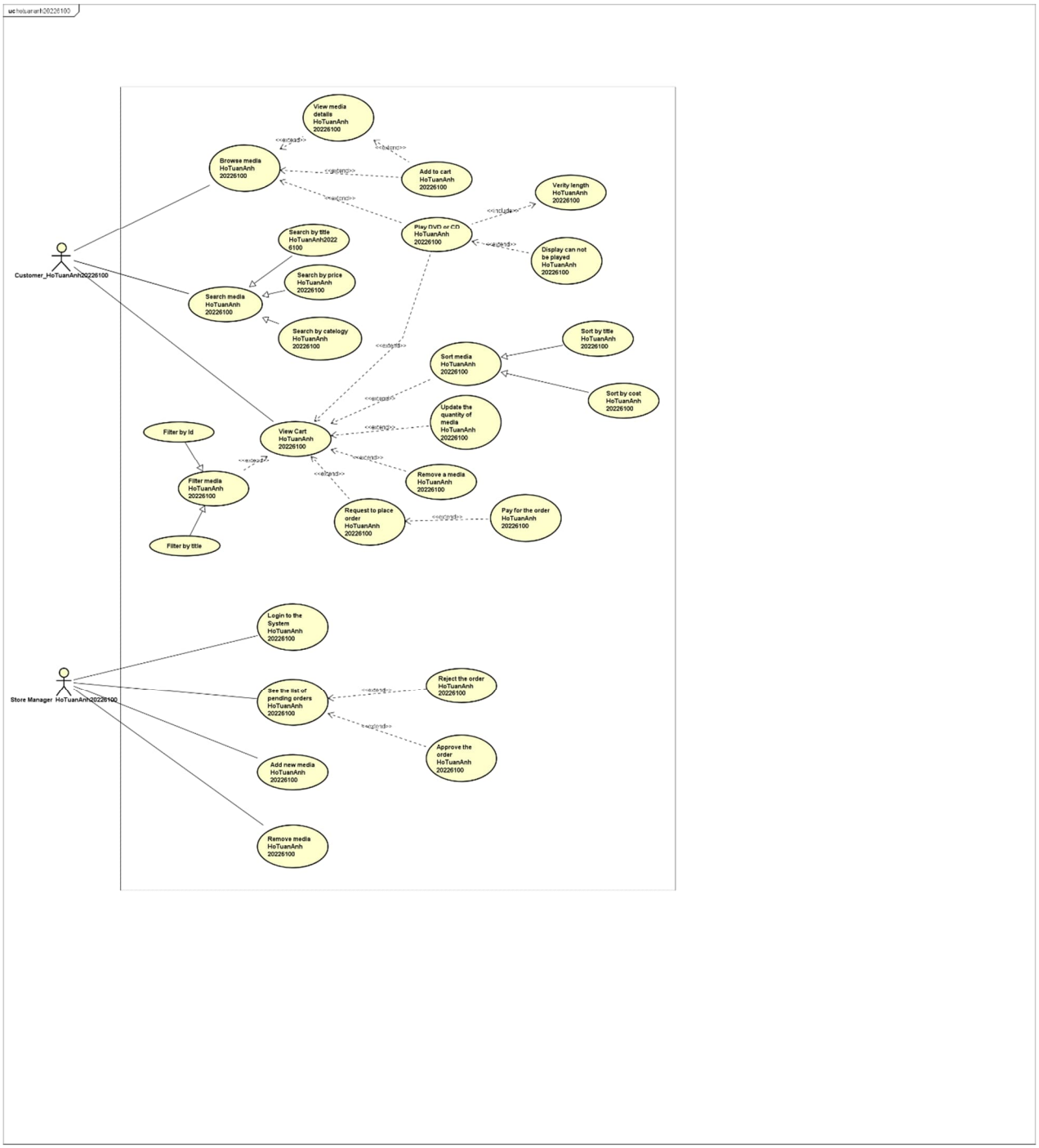



*Figure 8.8: Cart*

```
src > hust > soict > itep > aims > exception > PlayerException.java > PlayerException
You, 9 hours ago | 1 author (You)
package hust.soict.itep.aims.exception;

You, 9 hours ago | 1 author (You)
public class PlayerException extends Exception {


    public PlayerException(String message) {
        super(message);
    }
      You, 9 hours ago • update lab5
}
```

*Figure 8.9: Exception*

# 3. Usecase Diagram



# 4. Class Diagram