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

BÁO CÁO THỰC HÀNH
IT1130-744362-2024.1

BÀI THỰC HÀNH 05

Họ và tên sv: Lê Đồng Cảnh Phú
Lớp: **K67 – Việt Nhật 01**
GVHD: Lê Thị Hoa
TA: **Đăng Mạnh Cường**

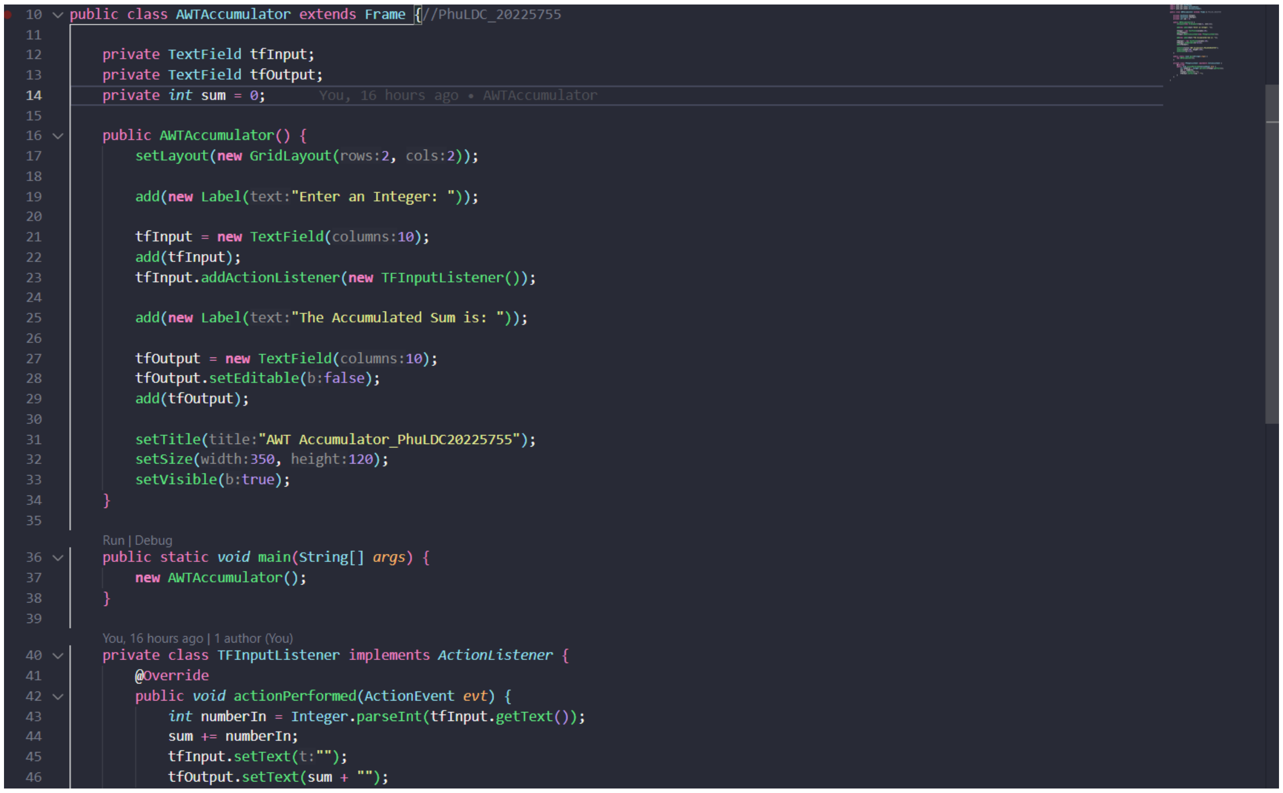
Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 5
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

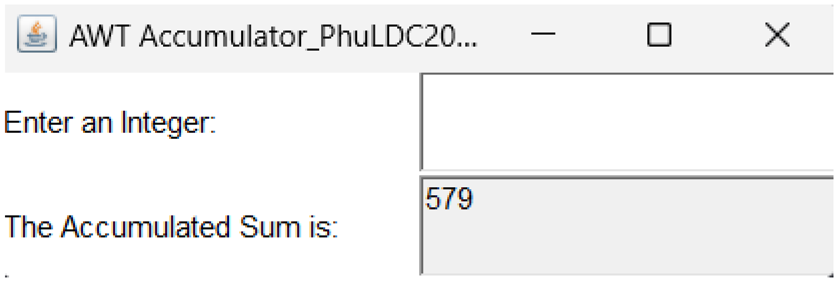
1. Swing components

1.1 AWTAccumulator



```
public class AWTAccumulator extends Frame {//PhuLDC_20225755

    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;        You, 16 hours ago • AWTAccumulator

    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_PhuLDC20225755");
        setSize(width:350, height:120);
        setVisible(b:true);
    }

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

    You, 16 hours ago | 1 author (You)
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText(t:"");
            tfOutput.setText(sum + "");
```

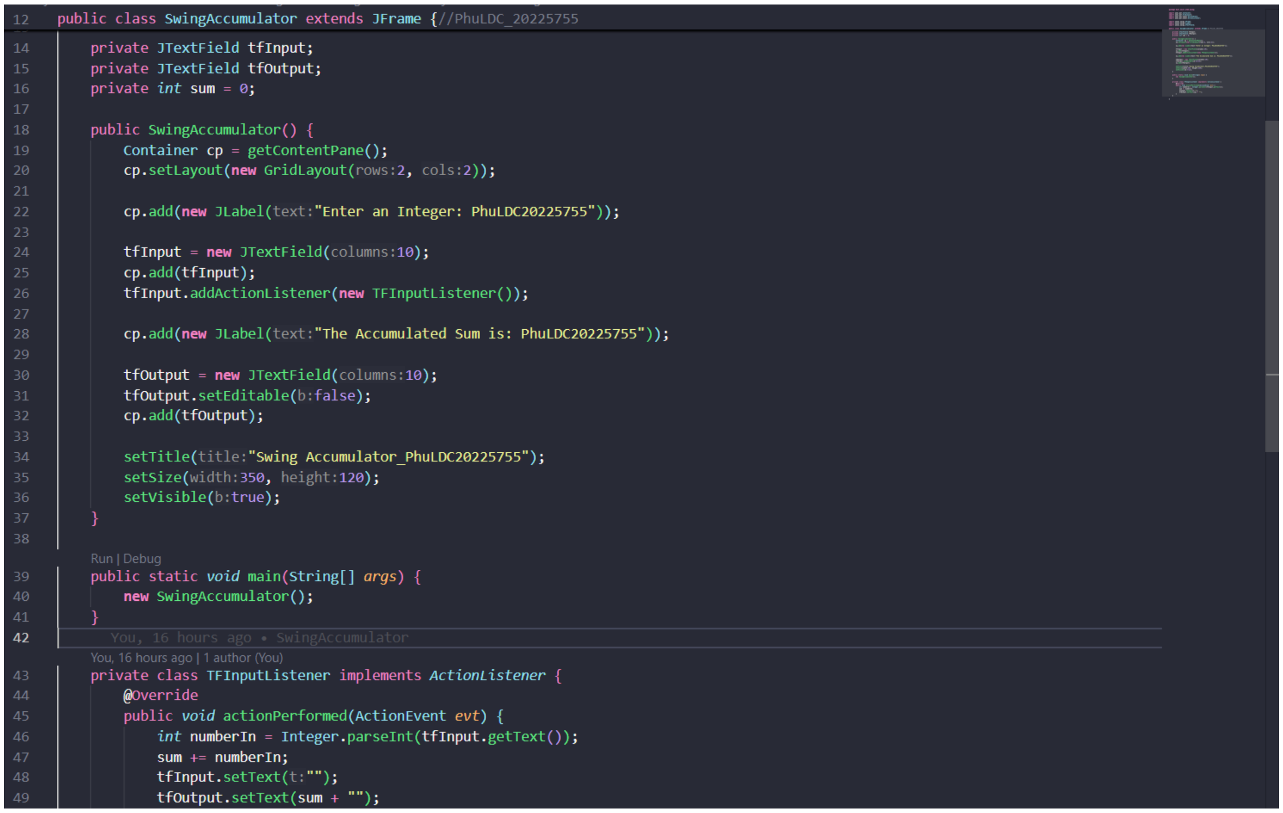
Figure 1.1: Source code of AWTAccumulator



```
Enter an Integer:
The Accumulated Sum is:      579
```

Figure 1.2: Demo of AWTAccumulator

1.2 SwingAccumulator



```java
public class SwingAccumulator extends JFrame {//PhuLDC_20225755

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

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

        cp.add(new JLabel(text:"Enter an Integer: PhuLDC20225755"));

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

        cp.add(new JLabel(text:"The Accumulated Sum is: PhuLDC20225755"));

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

        setTitle(title:"Swing Accumulator_PhuLDC20225755");
        setSize(width:350, height:120);
        setVisible(b:true);
    }

    Run | Debug
    public static void main(String[] args) {
        new SwingAccumulator();
    }
      You, 16 hours ago • SwingAccumulator
    You, 16 hours ago | 1 author (You)
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText(t:"");
            tfOutput.setText(sum + "");
```

Figure 1.3: Source code of SwingAccumulator

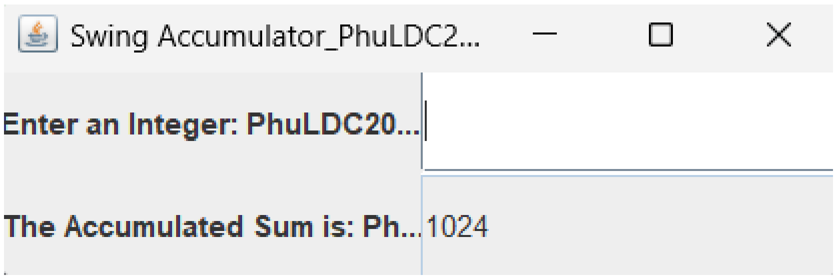
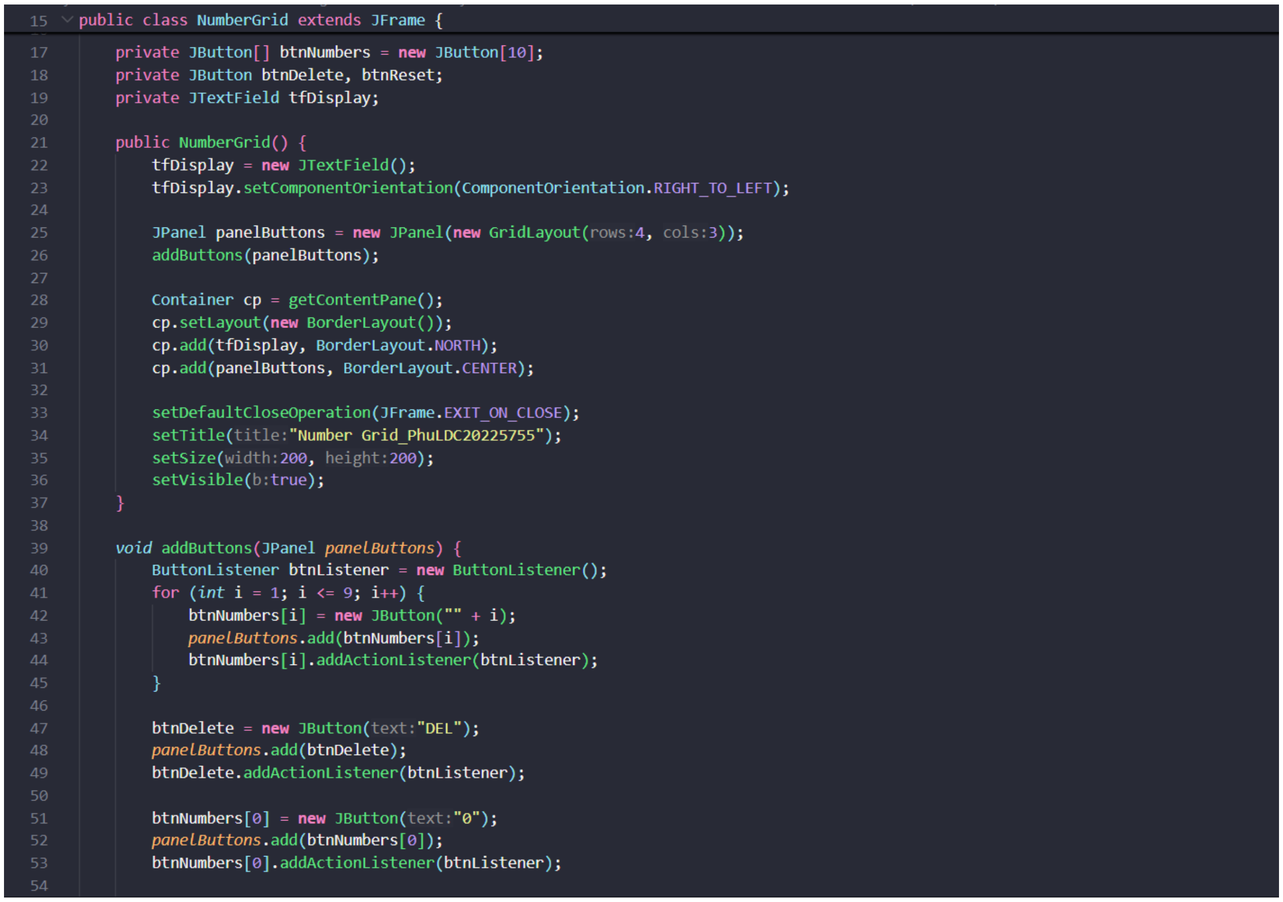

Swing Accumulator_PhuLDC2...
Enter an Integer: PhuLDC20...
The Accumulated Sum is: Ph...
1024

Figure 1.4: Demo of SwingAccumulator

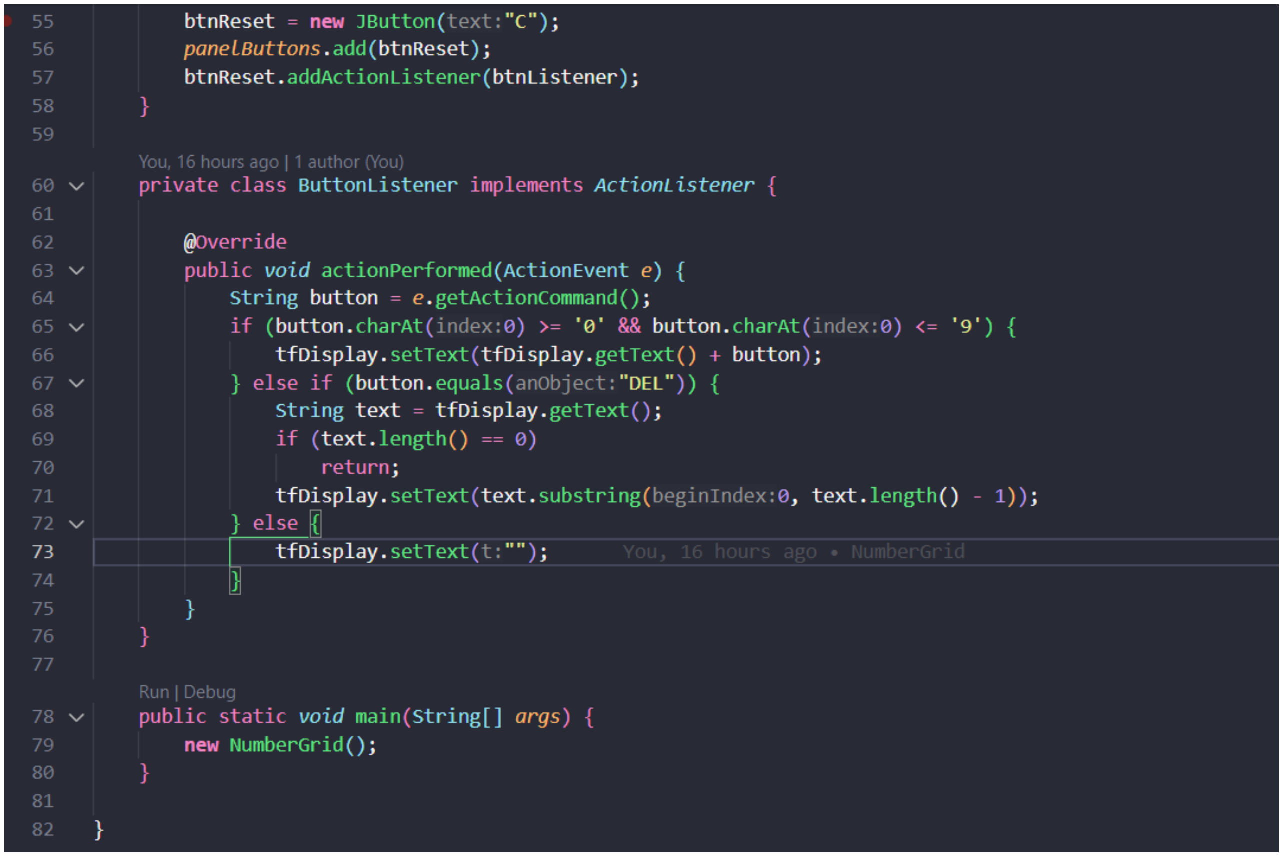
2 Organizing Swing components with Layout Managers

2.1 Code



```
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(rows:4, cols: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_PhuLDC20225755");
        setSize(width:200, height:200);
        setVisible(b: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(text:"DEL");
        panelButtons.add(btnDelete);
        btnDelete.addActionListener(btnListener);

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

```

Figure 2.1: Source code of NumberGrid 1



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

    You, 16 hours ago | 1 author (You)
    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 text = tfDisplay.getText();
                if (text.length() == 0)
                    return;
                tfDisplay.setText(text.substring(beginIndex:0, text.length() - 1));
            } else {
                tfDisplay.setText(t:"");        You, 16 hours ago • NumberGrid
            }
        }
    }

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

}
```

Figure 2.2: Source code of NumberGrid 2

2.2 Demo

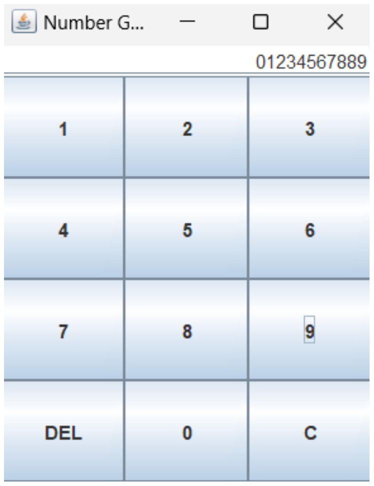


Figure 2.3: Demo buttons 0-9

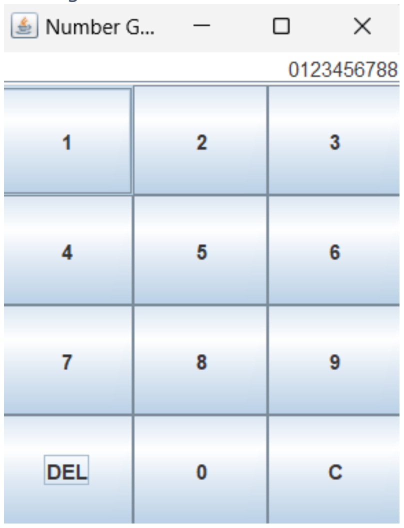


Figure 2.4: Demo DEL button

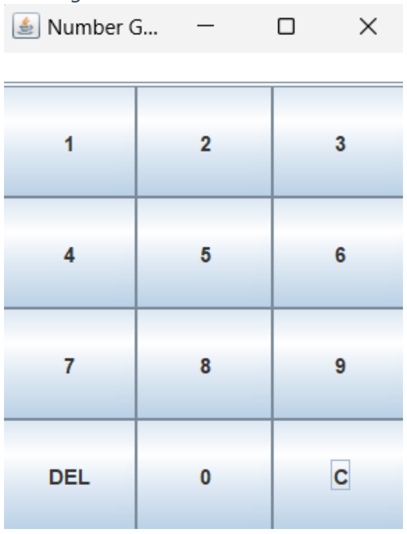
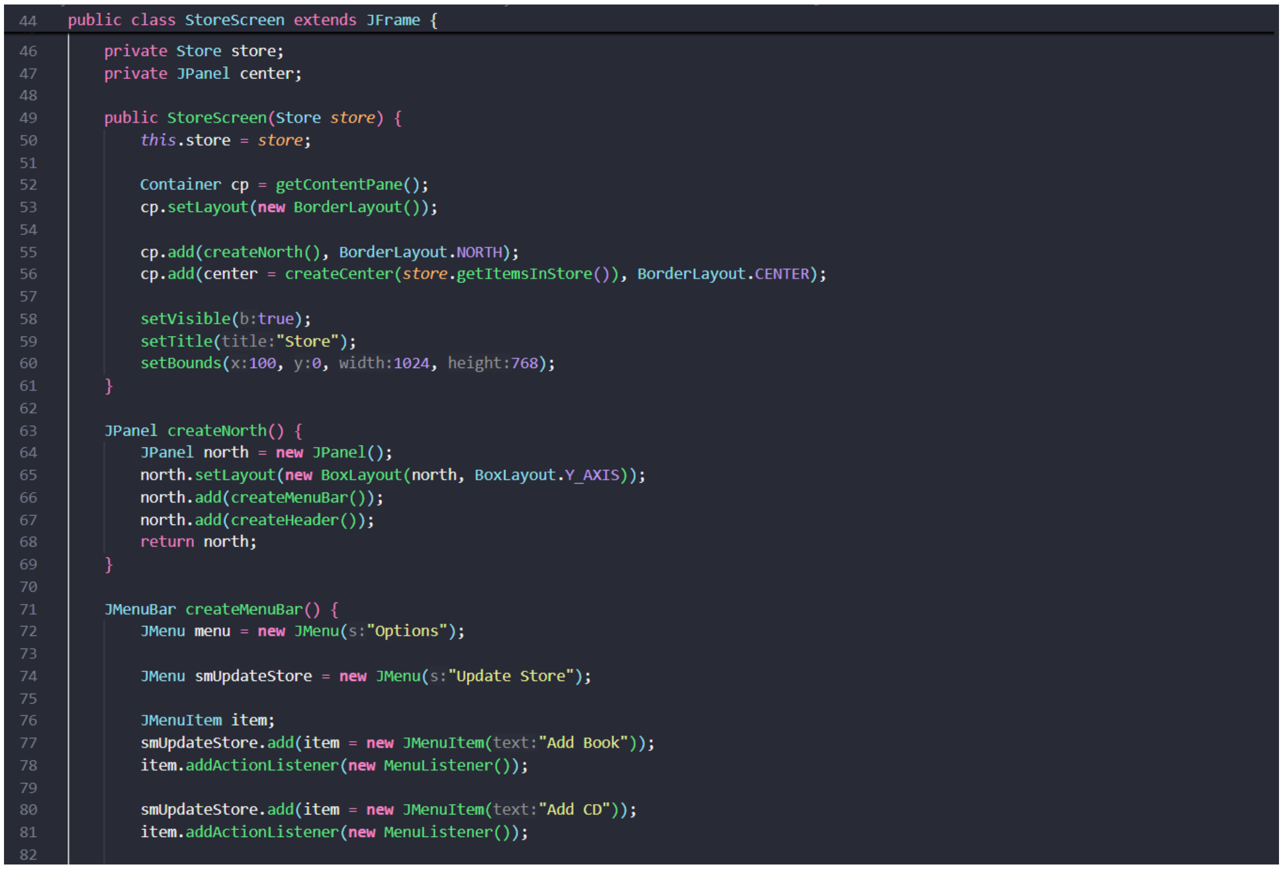


Figure 2.5: Demo C button

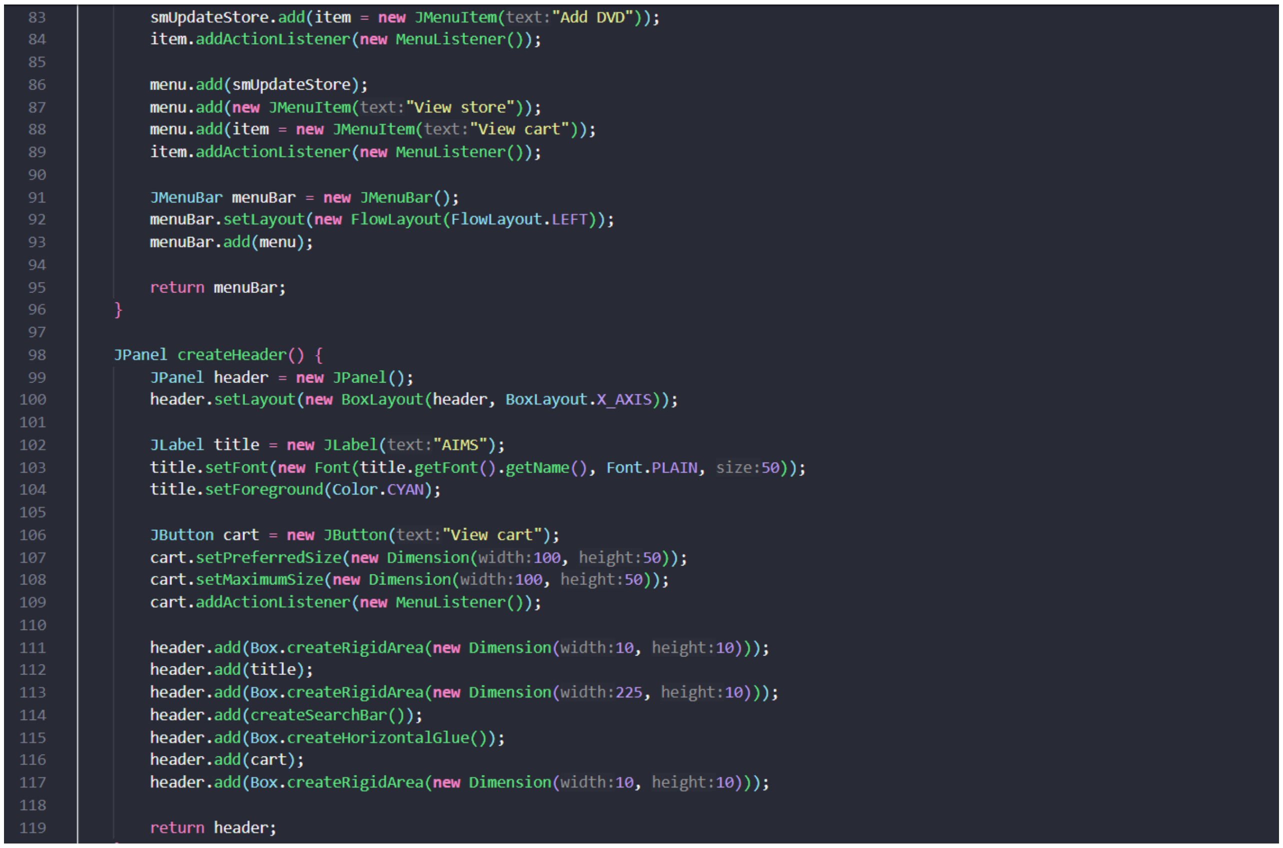
3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen



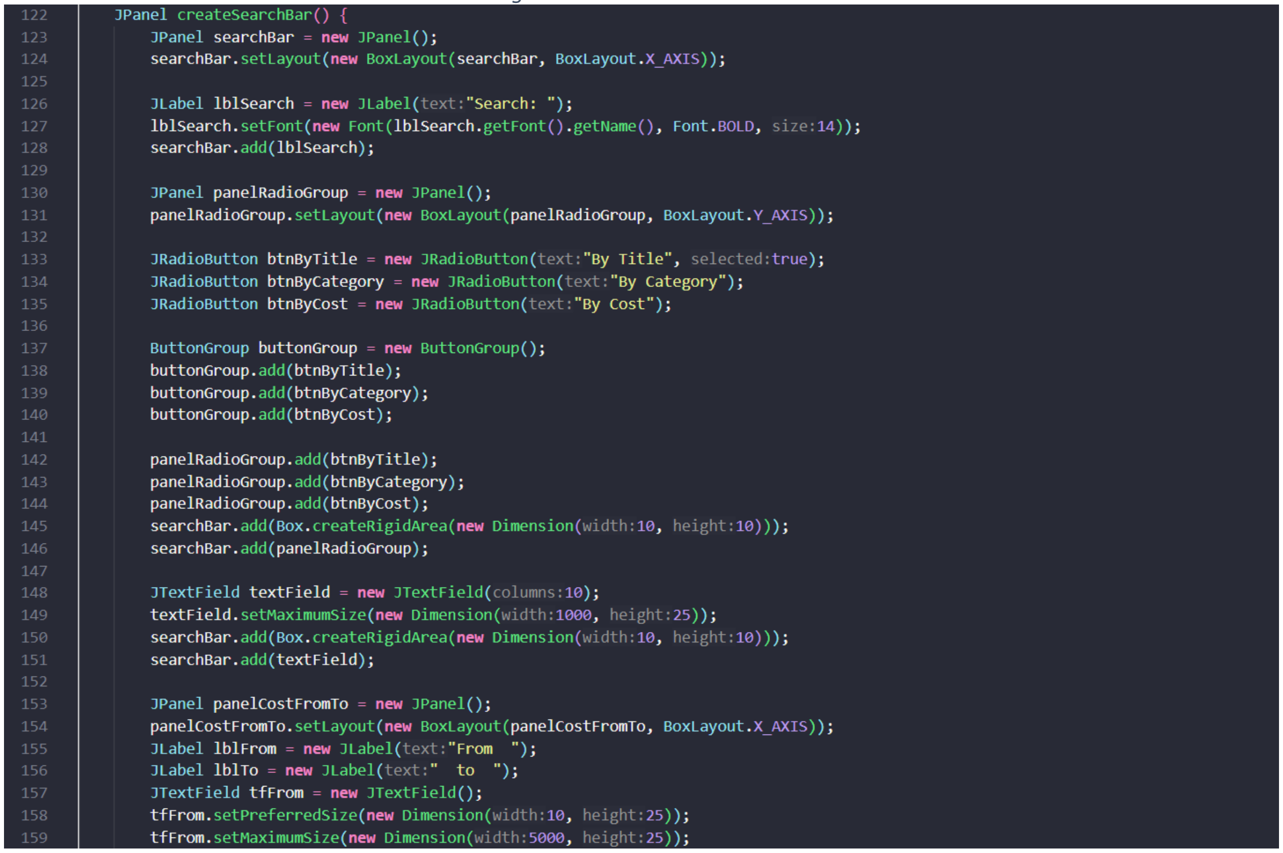
```java
public class StoreScreen extends JFrame {

    private Store store;
    private JPanel center;

    public StoreScreen(Store store) {
        this.store = store;

        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());

        cp.add(createNorth(), BorderLayout.NORTH);
        cp.add(center = createCenter(store.getItemsInStore()), BorderLayout.CENTER);

        setVisible(b:true);
        setTitle(title:"Store");
        setBounds(x:100, y:0, width:1024, height:768);
    }

    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");

        JMenuItem item;
        smUpdateStore.add(item = new JMenuItem(text:"Add Book"));
        item.addActionListener(new MenuListener());

        smUpdateStore.add(item = new JMenuItem(text:"Add CD"));
        item.addActionListener(new MenuListener());

```

Figure 3.1: Class StoreScreen 1

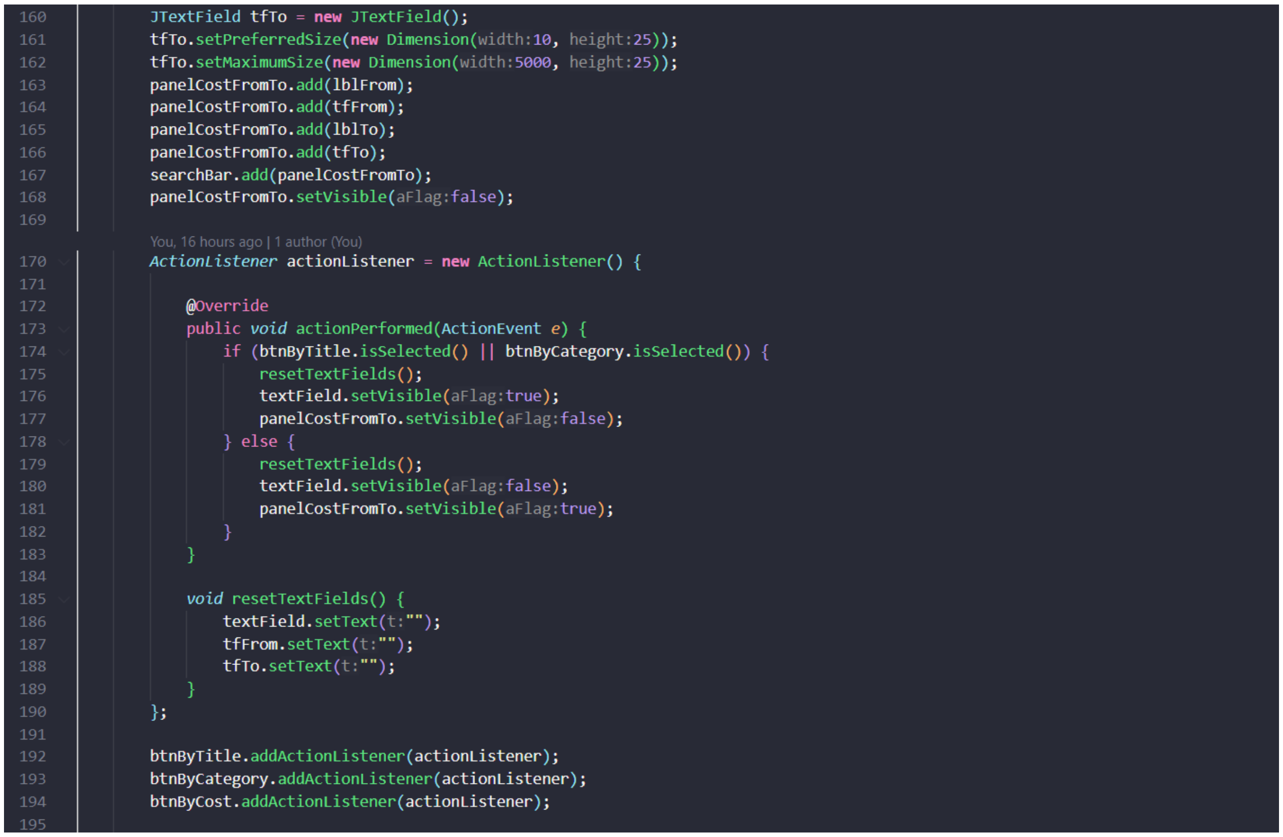


```
        smUpdateStore.add(item = new JMenuItem(text:"Add DVD"));
        item.addActionListener(new MenuListener());

        menu.add(smUpdateStore);
        menu.add(new JMenuItem(text:"View store"));
        menu.add(item = new JMenuItem(text:"View cart"));
        item.addActionListener(new MenuListener());

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

        return menuBar;
    }

    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));
        cart.addActionListener(new MenuListener());

        header.add(Box.createRigidArea(new Dimension(width:10, height:10)));
        header.add(title);
        header.add(Box.createRigidArea(new Dimension(width:225, height:10)));
        header.add(createSearchBar());
        header.add(Box.createHorizontalGlue());
        header.add(cart);
        header.add(Box.createRigidArea(new Dimension(width:10, height:10)));

        return header;
```

Figure 3.2: Class StoreScreen 2

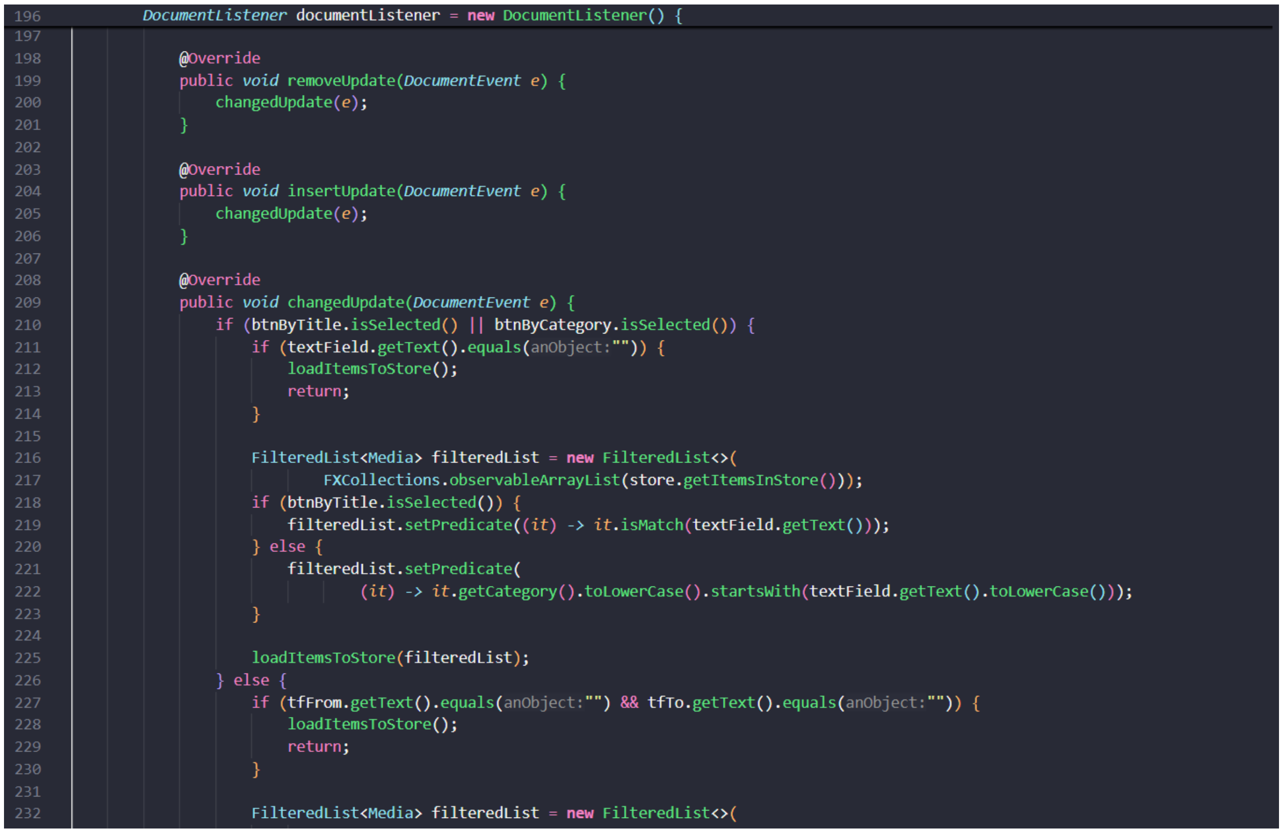


```
    JPanel createSearchBar() {
        JPanel searchBar = new JPanel();
        searchBar.setLayout(new BoxLayout(searchBar, BoxLayout.X_AXIS));

        JLabel lblSearch = new JLabel(text:"Search: ");
        lblSearch.setFont(new Font(lblSearch.getFont().getName(), Font.BOLD, size:14));
        searchBar.add(lblSearch);

        JPanel panelRadioGroup = new JPanel();
        panelRadioGroup.setLayout(new BoxLayout(panelRadioGroup, BoxLayout.Y_AXIS));

        JRadioButton btnByTitle = new JRadioButton(text:"By Title", selected:true);
        JRadioButton btnByCategory = new JRadioButton(text:"By Category");
        JRadioButton btnByCost = new JRadioButton(text:"By Cost");

        ButtonGroup buttonGroup = new ButtonGroup();
        buttonGroup.add(btnByTitle);
        buttonGroup.add(btnByCategory);
        buttonGroup.add(btnByCost);

        panelRadioGroup.add(btnByTitle);
        panelRadioGroup.add(btnByCategory);
        panelRadioGroup.add(btnByCost);
        searchBar.add(Box.createRigidArea(new Dimension(width:10, height:10)));
        searchBar.add(panelRadioGroup);

        JTextField textField = new JTextField(columns:10);
        textField.setMaximumSize(new Dimension(width:1000, height:25));
        searchBar.add(Box.createRigidArea(new Dimension(width:10, height:10)));
        searchBar.add(textField);

        JPanel panelCostFromTo = new JPanel();
        panelCostFromTo.setLayout(new BoxLayout(panelCostFromTo, BoxLayout.X_AXIS));
        JLabel lblFrom = new JLabel(text:"From  ");
        JLabel lblTo = new JLabel(text:"  to  ");
        JTextField tfFrom = new JTextField();
        tfFrom.setPreferredSize(new Dimension(width:10, height:25));
        tfFrom.setMaximumSize(new Dimension(width:5000, height:25));
```



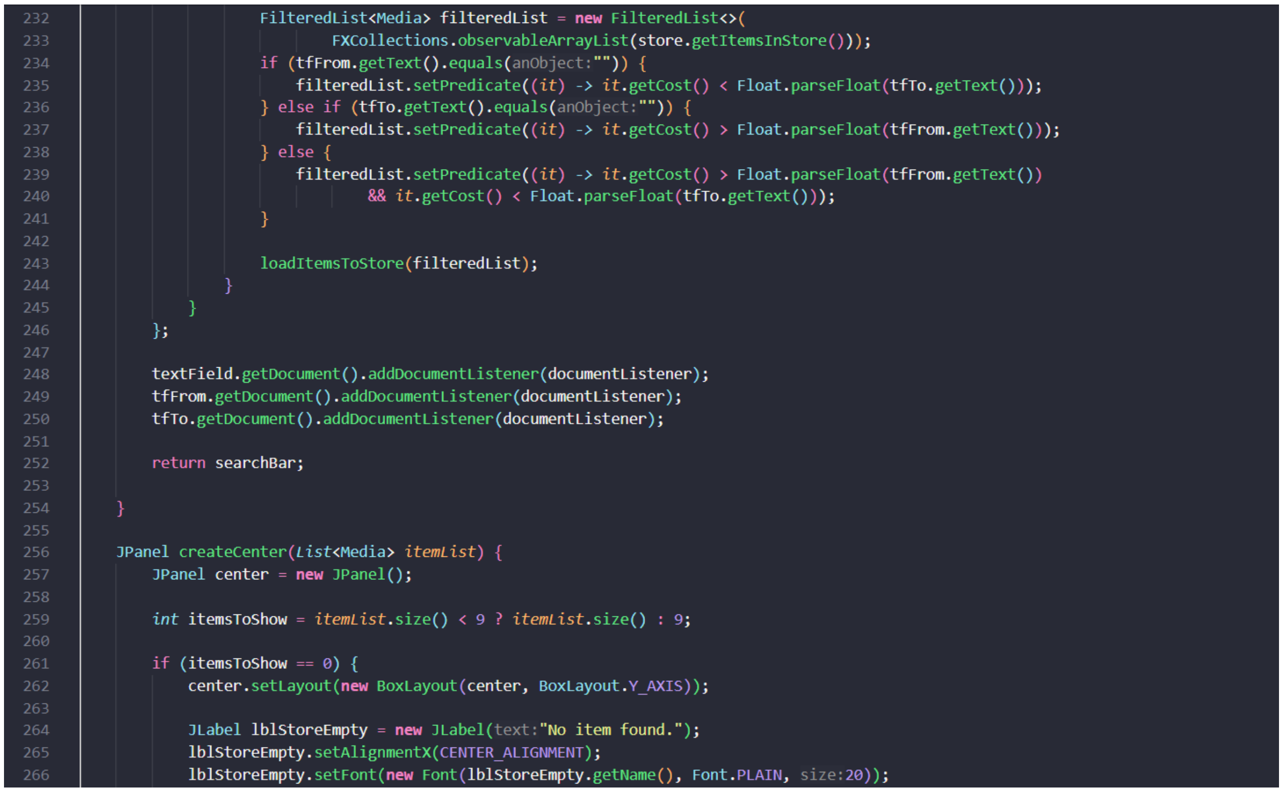
```
        JTextField tfTo = new JTextField();
        tfTo.setPreferredSize(new Dimension(width:10, height:25));
        tfTo.setMaximumSize(new Dimension(width:5000, height:25));
        panelCostFromTo.add(lblFrom);
        panelCostFromTo.add(tfFrom);
        panelCostFromTo.add(lblTo);
        panelCostFromTo.add(tfTo);
        searchBar.add(panelCostFromTo);
        panelCostFromTo.setVisible(aFlag:false);

        You, 16 hours ago | 1 author (You)
        ActionListener actionListener = new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                if (btnByTitle.isSelected() || btnByCategory.isSelected()) {
                    resetTextFields();
                    textField.setVisible(aFlag:true);
                    panelCostFromTo.setVisible(aFlag:false);
                } else {
                    resetTextFields();
                    textField.setVisible(aFlag:false);
                    panelCostFromTo.setVisible(aFlag:true);
                }
            }

            void resetTextFields() {
                textField.setText(t:"");
                tfFrom.setText(t:"");
                tfTo.setText(t:"");
            }
        };

        btnByTitle.addActionListener(actionListener);
        btnByCategory.addActionListener(actionListener);
        btnByCost.addActionListener(actionListener);
```

Figure 3.3: Class StoreScreen 3



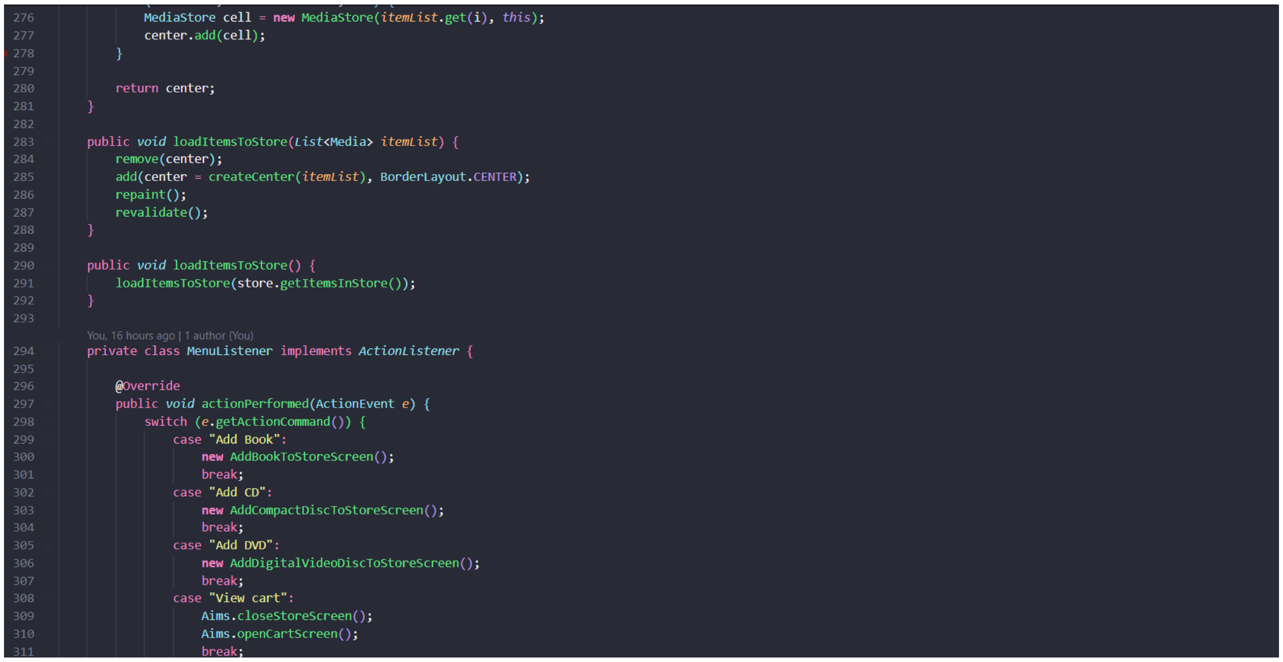
```java
        DocumentListener documentListener = new DocumentListener() {

            @Override
            public void removeUpdate(DocumentEvent e) {
                changedUpdate(e);
            }

            @Override
            public void insertUpdate(DocumentEvent e) {
                changedUpdate(e);
            }

            @Override
            public void changedUpdate(DocumentEvent e) {
                if (btnByTitle.isSelected() || btnByCategory.isSelected()) {
                    if (textField.getText().equals(anObject:"")) {
                        loadItemsToStore();
                        return;
                    }

                    FilteredList<Media> filteredList = new FilteredList<>(
                            FXCollections.observableArrayList(store.getItemsInStore()));
                    if (btnByTitle.isSelected()) {
                        filteredList.setPredicate((it) -> it.isMatch(textField.getText()));
                    } else {
                        filteredList.setPredicate(
                                (it) -> it.getCategory().toLowerCase().startsWith(textField.getText().toLowerCase()));
                    }

                    loadItemsToStore(filteredList);
                } else {
                    if (tfFrom.getText().equals(anObject:"") && tfTo.getText().equals(anObject:"")) {
                        loadItemsToStore();
                        return;
                    }

                    FilteredList<Media> filteredList = new FilteredList<>(
```

Figure 3.4: Class StoreScreen 4



```java
                    FilteredList<Media> filteredList = new FilteredList<>(
                            FXCollections.observableArrayList(store.getItemsInStore()));
                    if (tfFrom.getText().equals(anObject:"")) {
                        filteredList.setPredicate((it) -> it.getCost() < Float.parseFloat(tfTo.getText()));
                    } else if (tfTo.getText().equals(anObject:"")) {
                        filteredList.setPredicate((it) -> it.getCost() > Float.parseFloat(tfFrom.getText()));
                    } else {
                        filteredList.setPredicate((it) -> it.getCost() > Float.parseFloat(tfFrom.getText())
                                && it.getCost() < Float.parseFloat(tfTo.getText()));
                    }

                    loadItemsToStore(filteredList);
                }
            }
        };

        textField.getDocument().addDocumentListener(documentListener);
        tfFrom.getDocument().addDocumentListener(documentListener);
        tfTo.getDocument().addDocumentListener(documentListener);

        return searchBar;

    }

    JPanel createCenter(List<Media> itemList) {
        JPanel center = new JPanel();

        int itemsToShow = itemList.size() < 9 ? itemList.size() : 9;

        if (itemsToShow == 0) {
            center.setLayout(new BoxLayout(center, BoxLayout.Y_AXIS));

            JLabel lblStoreEmpty = new JLabel(text:"No item found.");
            lblStoreEmpty.setAlignmentX(CENTER_ALIGNMENT);
            lblStoreEmpty.setFont(new Font(lblStoreEmpty.getName(), Font.PLAIN, size:20));
```

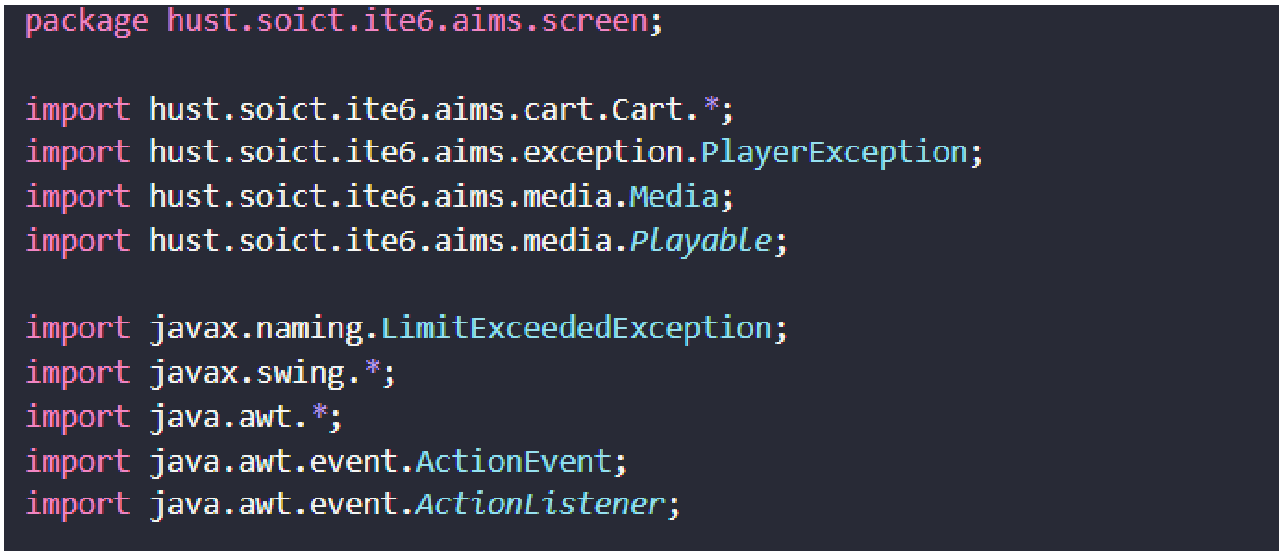
Figure 3.5: Class StoreScreen 5



```
            MediaStore cell = new MediaStore(itemList.get(i), this);
            center.add(cell);
        }

        return center;
    }

    public void loadItemsToStore(List<Media> itemList) {
        remove(center);
        add(center = createCenter(itemList), BorderLayout.CENTER);
        repaint();
        revalidate();
    }

    public void loadItemsToStore() {
        loadItemsToStore(store.getItemsInStore());
    }

    You, 16 hours ago | 1 author (You)
    private class MenuListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent e) {
            switch (e.getActionCommand()) {
                case "Add Book":
                    new AddBookToStoreScreen();
                    break;
                case "Add CD":
                    new AddCompactDiscToStoreScreen();
                    break;
                case "Add DVD":
                    new AddDigitalVideoDiscToStoreScreen();
                    break;
                case "View cart":
                    Aims.closeStoreScreen();
                    Aims.openCartScreen();
                    break;
```

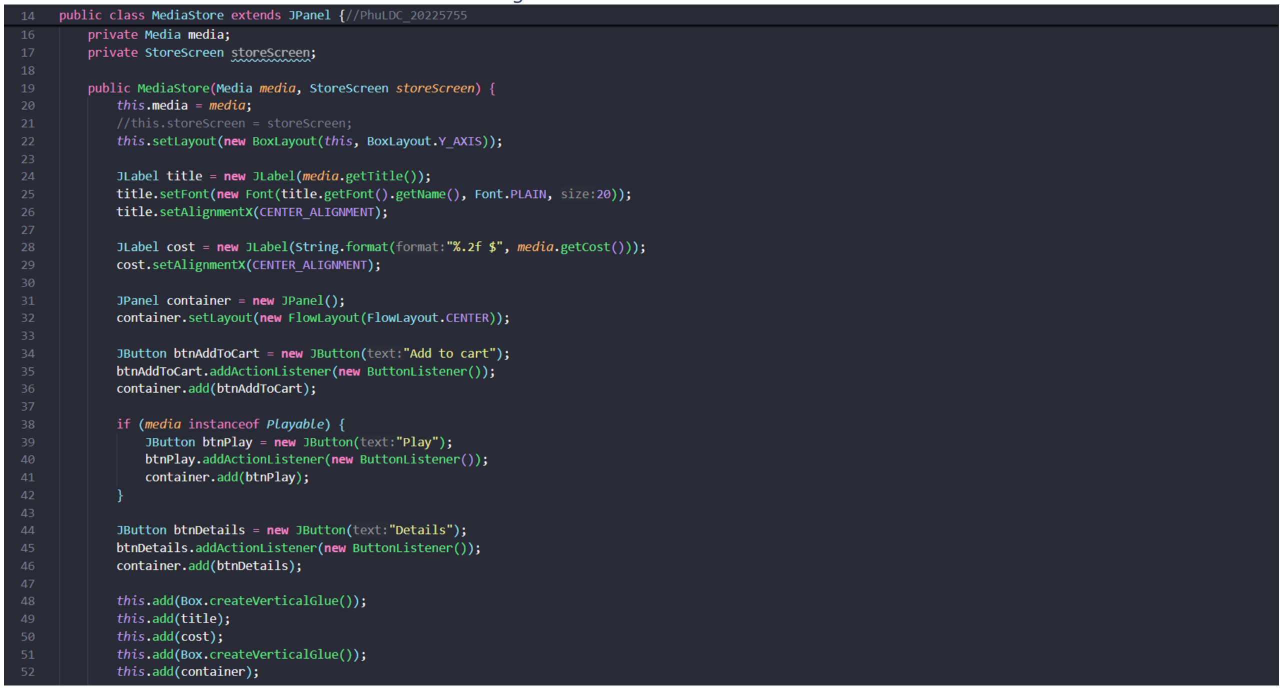
Figure 3.6: Class StoreScreen 6

3.2 Create class MediaStore



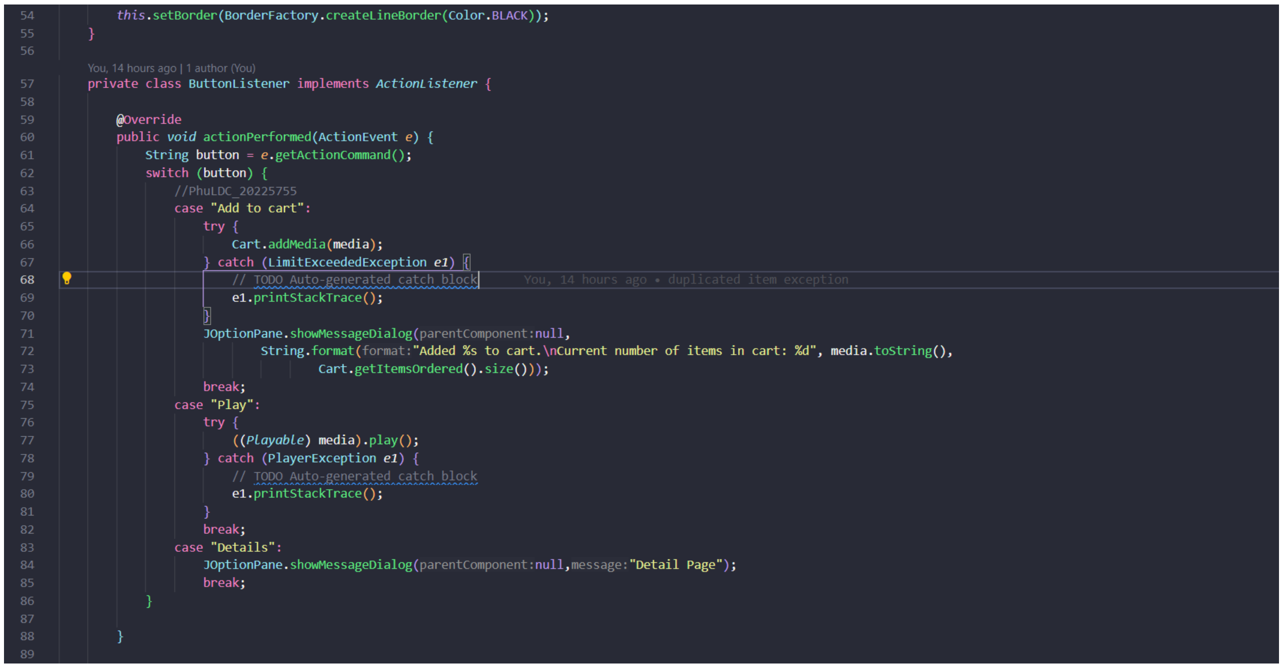
```java
package hust.soict.ite6.aims.screen;

import hust.soict.ite6.aims.cart.Cart.*;
import hust.soict.ite6.aims.exception.PlayerException;
import hust.soict.ite6.aims.media.Media;
import hust.soict.ite6.aims.media.Playable;

import javax.naming.LimitExceededException;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

Figure 3.7: Class MediaStore 1



```java
public class MediaStore extends JPanel {//PhuLDC_20225755
    private Media media;
    private StoreScreen storeScreen;

    public MediaStore(Media media, StoreScreen storeScreen) {
        this.media = media;
        //this.storeScreen = storeScreen;
        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

        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(String.format(format:"%.2f $", media.getCost()));
        cost.setAlignmentX(CENTER_ALIGNMENT);

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

        JButton btnAddToCart = new JButton(text:"Add to cart");
        btnAddToCart.addActionListener(new ButtonListener());
        container.add(btnAddToCart);

        if (media instanceof Playable) {
            JButton btnPlay = new JButton(text:"Play");
            btnPlay.addActionListener(new ButtonListener());
            container.add(btnPlay);
        }

        JButton btnDetails = new JButton(text:"Details");
        btnDetails.addActionListener(new ButtonListener());
        container.add(btnDetails);

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

Figure 3.8: Class MediaStore 2



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

    You, 14 hours ago | 1 author (You)
    private class ButtonListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
            switch (button) {
                //PhuLDC_20225755
                case "Add to cart":
                    try {
                        Cart.addMedia(media);
                    } catch (LimitExceededException e1) {
                        // TODO Auto-generated catch block      You, 14 hours ago • duplicated item exception
                        e1.printStackTrace();
                    }
                    JOptionPane.showMessageDialog(parentComponent:null,
                            String.format(format:"Added %s to cart.\nCurrent number of items in cart: %d", media.toString(),
                                    Cart.getItemsOrdered().size()));
                    break;
                case "Play":
                    try {
                        ((Playable) media).play();
                    } catch (PlayerException e1) {
                        // TODO Auto-generated catch block
                        e1.printStackTrace();
                    }
                    break;
                case "Details":
                    JOptionPane.showMessageDialog(parentComponent:null,message:"Detail Page");
                    break;
            }

        }

```

Figure 3.9: Class MediaStore 3

3.3 Demo

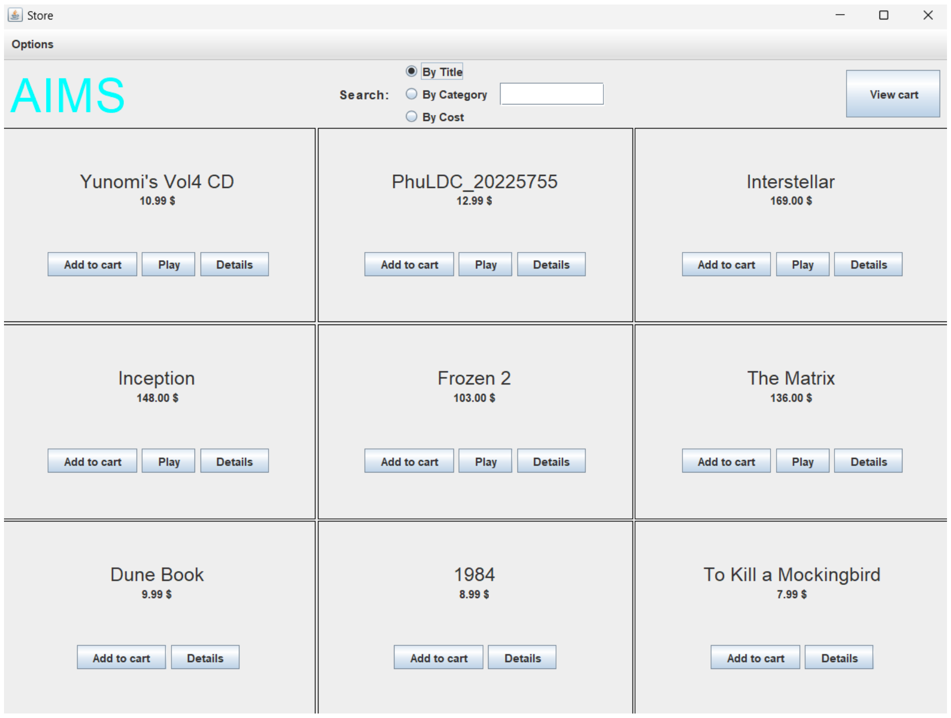
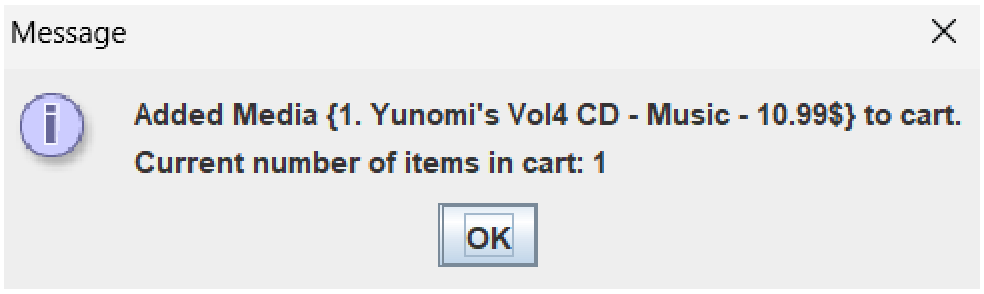
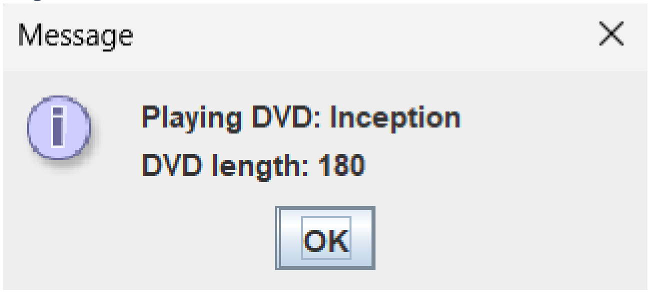


Figure 3.10: StoreScreen



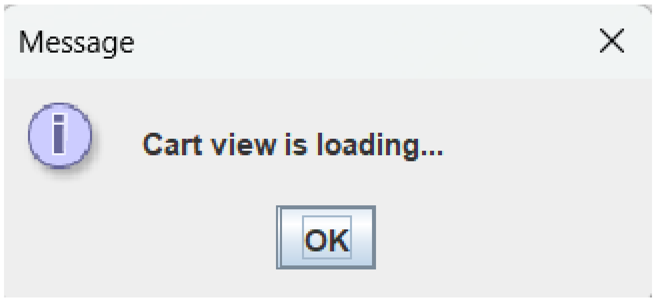
Added Media {1. Yunomi's Vol4 CD - Music - 10.99$} to cart.
Current number of items in cart: 1

Figure 3.11 Demo Add to cart button



Playing DVD: Inception
DVD length: 180

Figure 3.12 Demo Play button

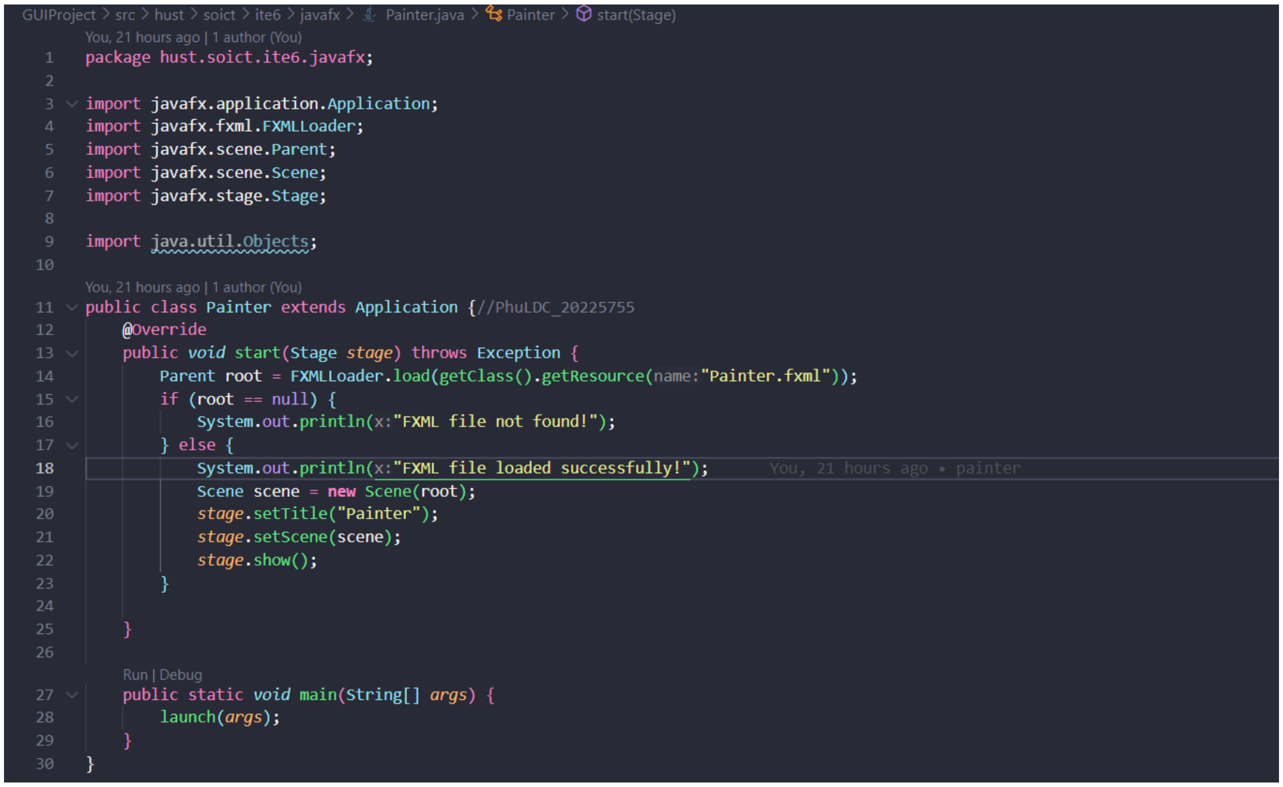


Cart view is loading...

Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

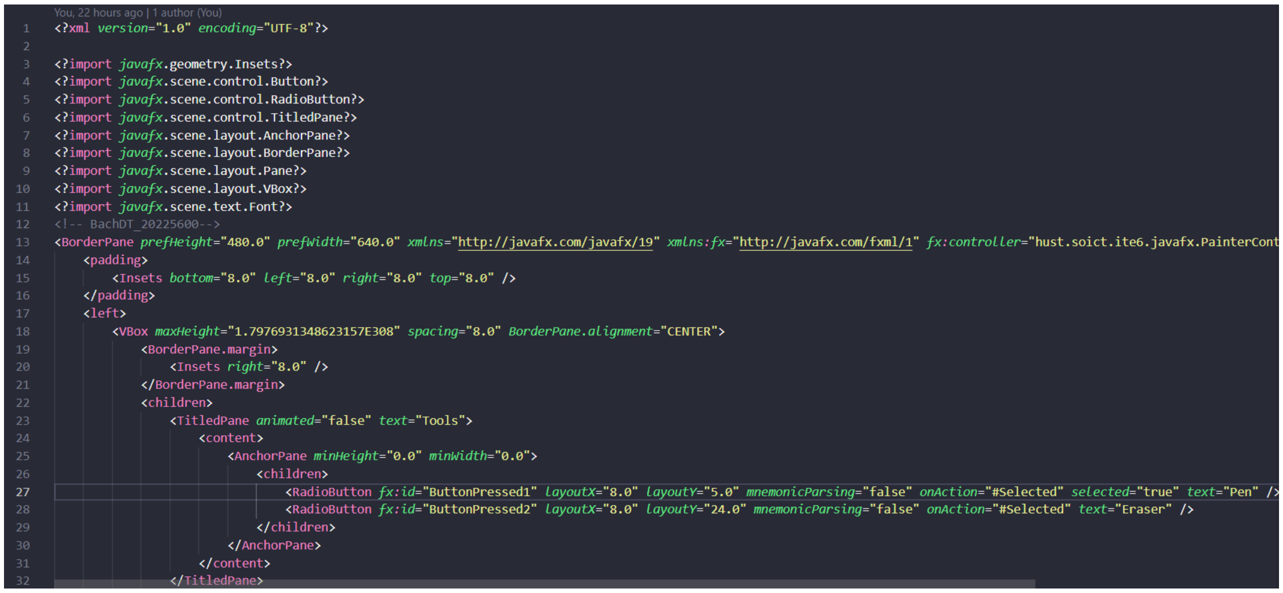


GUIProject > src > hust > soict > ite6 > javafx > Painter.java > Painter > start(Stage)

```java
You, 21 hours ago | 1 author (You)
package hust.soict.ite6.javafx;

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

import java.util.Objects;

You, 21 hours ago | 1 author (You)
public class Painter extends Application {//PhuLDC_20225755
    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource(name:"Painter.fxml"));
        if (root == null) {
            System.out.println(x:"FXML file not found!");
        } else {
            System.out.println(x:"FXML file loaded successfully!");      You, 21 hours ago • painter
            Scene scene = new Scene(root);
            stage.setTitle("Painter");
            stage.setScene(scene);
            stage.show();
        }

    }

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

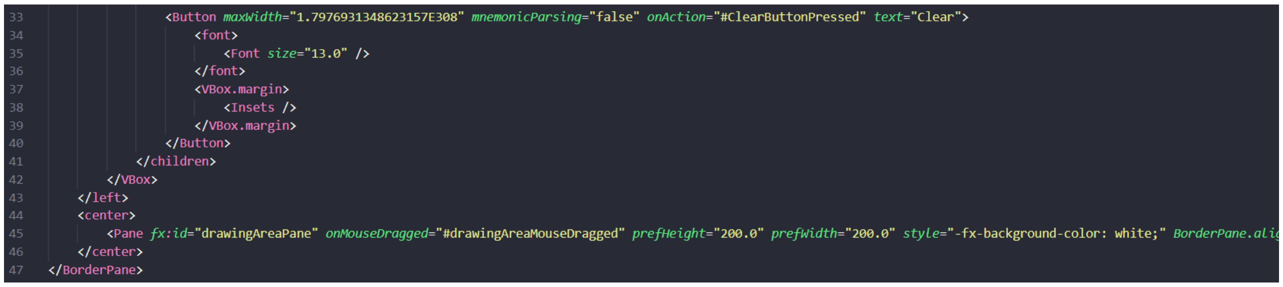
Figure 4.1: Class Painter

4.2 Create Painter.fxml



```xml
You, 22 hours ago | 1 author (You)
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TitledPane?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>
<!-- BachDT_20225600-->
<BorderPane prefHeight="480.0" prefWidth="640.0" xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1" fx:controller="hust.soict.ite6.javafx.PainterCont
    <padding>
        <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
    </padding>
    <left>
        <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
            <BorderPane.margin>
                <Insets right="8.0" />
            </BorderPane.margin>
            <children>
                <TitledPane animated="false" text="Tools">
                    <content>
                        <AnchorPane minHeight="0.0" minWidth="0.0">
                            <children>
                                <RadioButton fx:id="ButtonPressed1" layoutX="8.0" layoutY="5.0" mnemonicParsing="false" onAction="#Selected" selected="true" text="Pen" />
                                <RadioButton fx:id="ButtonPressed2" layoutX="8.0" layoutY="24.0" mnemonicParsing="false" onAction="#Selected" text="Eraser" />
                            </children>
                        </AnchorPane>
                    </content>
                </TitledPane>
```

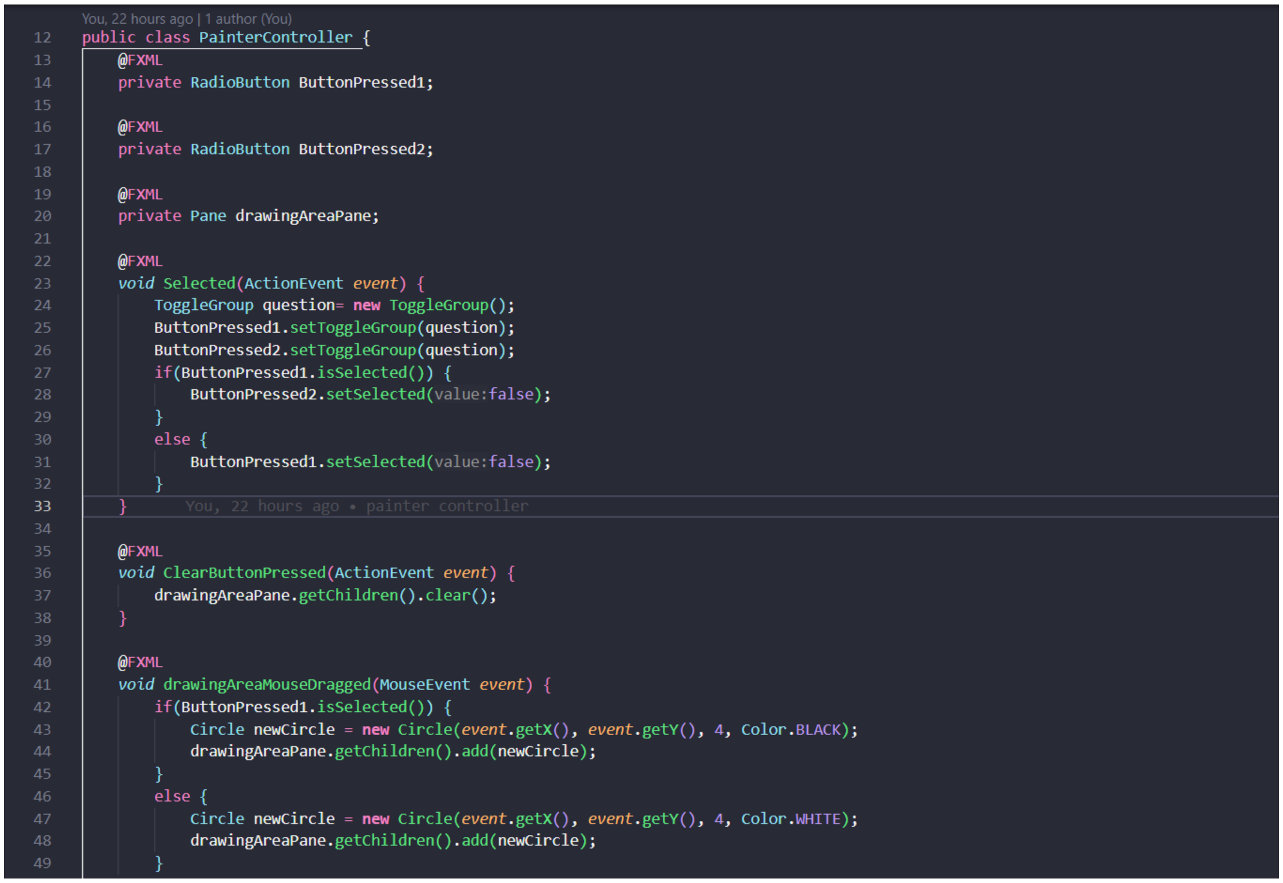
Figure 4.2: Painter.fxml 1



```xml
                <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#ClearButtonPressed" text="Clear">
                    <font>
                        <Font size="13.0" />
                    </font>
                    <VBox.margin>
                        <Insets />
                    </VBox.margin>
                </Button>
            </children>
        </VBox>
    </left>
    <center>
        <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" prefHeight="200.0" prefWidth="200.0" style="-fx-background-color: white;" BorderPane.ali
    </center>
</BorderPane>
```

Figure 4.3: Painter.fxml 2

4.3 Create class PainterController



```java
You, 22 hours ago | 1 author (You)
public class PainterController {
    @FXML
    private RadioButton ButtonPressed1;

    @FXML
    private RadioButton ButtonPressed2;

    @FXML
    private Pane drawingAreaPane;

    @FXML
    void Selected(ActionEvent event) {
        ToggleGroup question= new ToggleGroup();
        ButtonPressed1.setToggleGroup(question);
        ButtonPressed2.setToggleGroup(question);
        if(ButtonPressed1.isSelected()) {
            ButtonPressed2.setSelected(value:false);
        }
        else {
            ButtonPressed1.setSelected(value:false);
        }
    }       You, 22 hours ago • painter controller

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

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        if(ButtonPressed1.isSelected()) {
            Circle newCircle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
            drawingAreaPane.getChildren().add(newCircle);
        }
        else {
            Circle newCircle = new Circle(event.getX(), event.getY(), 4, Color.WHITE);
            drawingAreaPane.getChildren().add(newCircle);
        }
```

Figure 4.4: PainterController

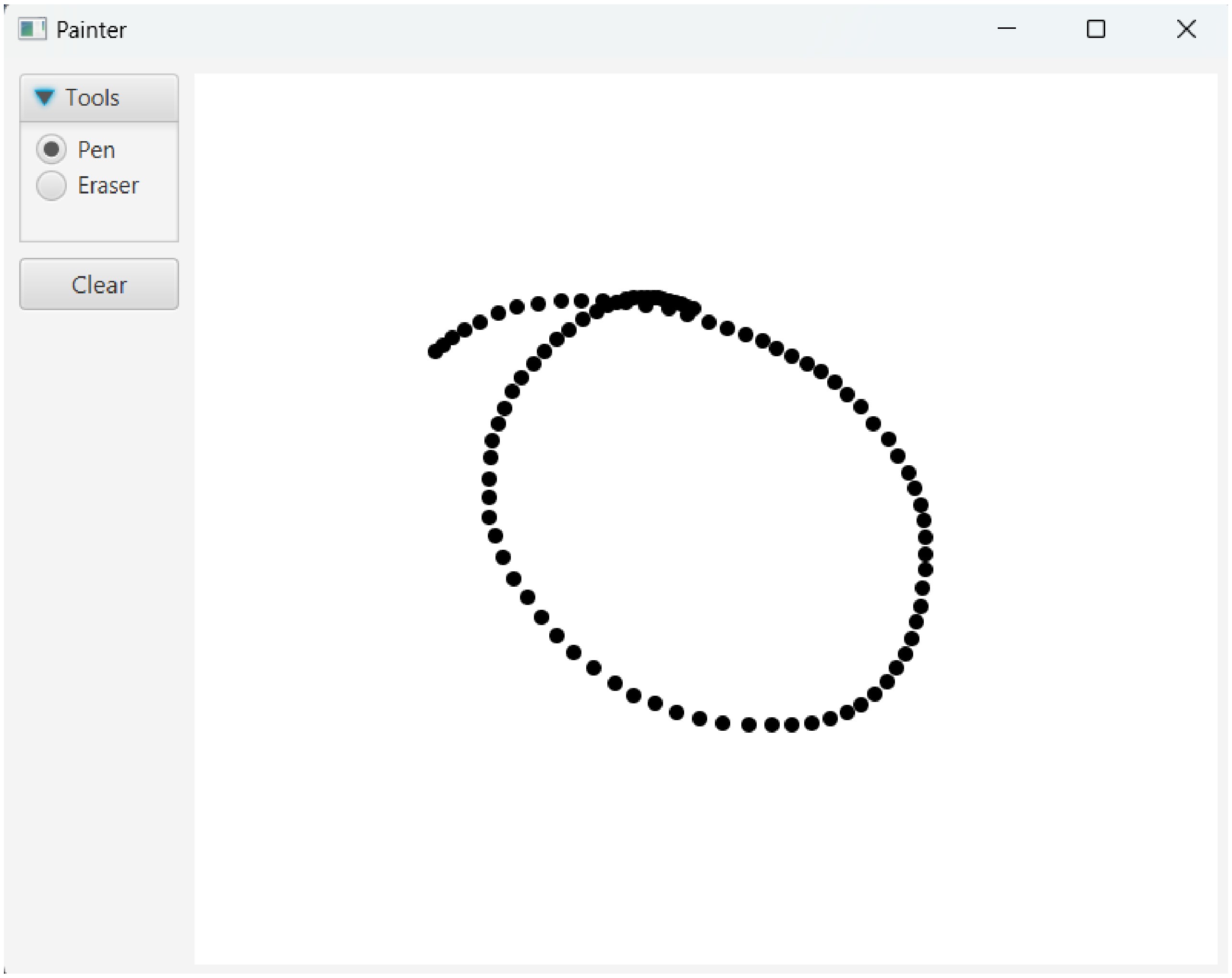


Figure 4.5: Use Pen

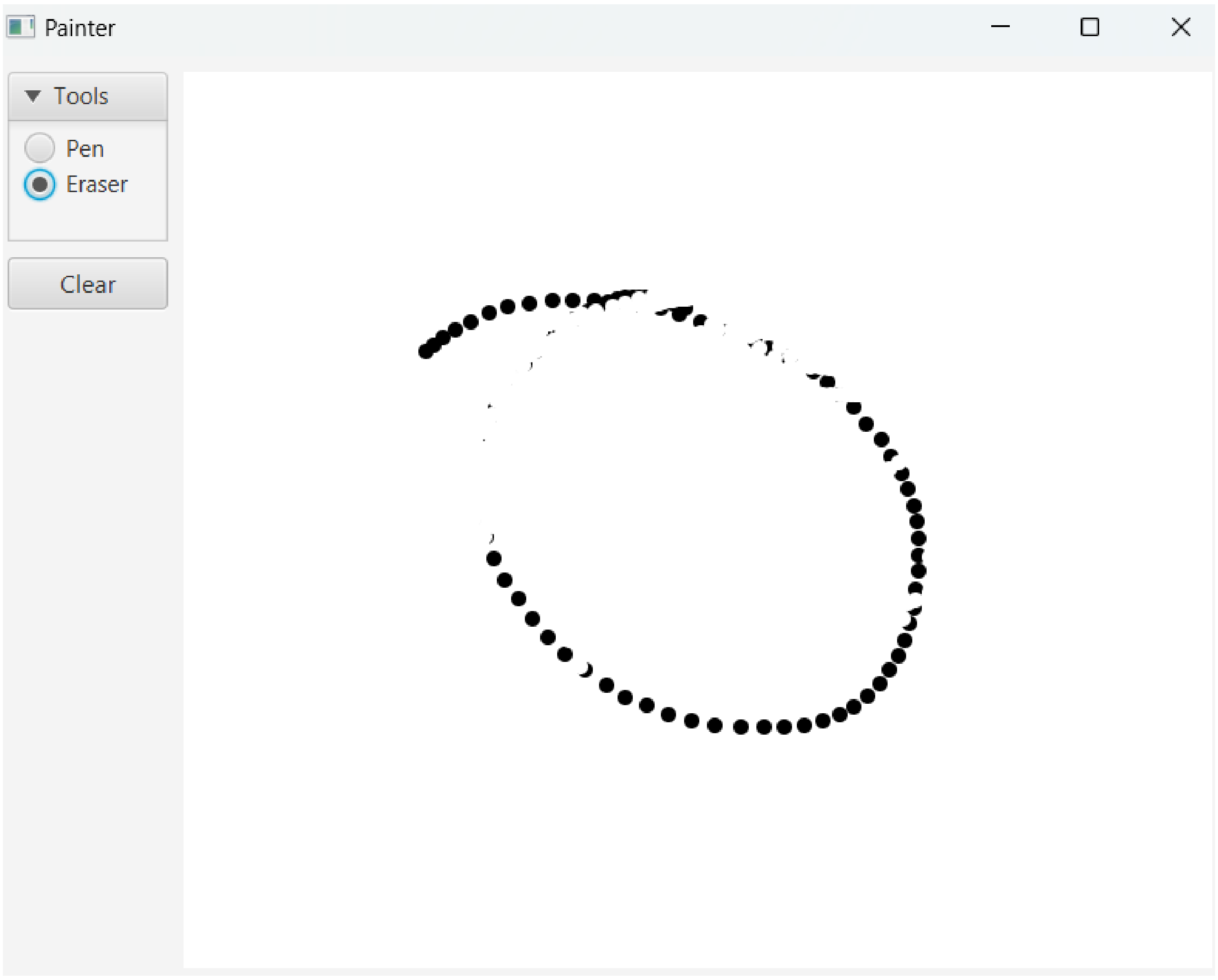


Figure 4.6: Use Eraser

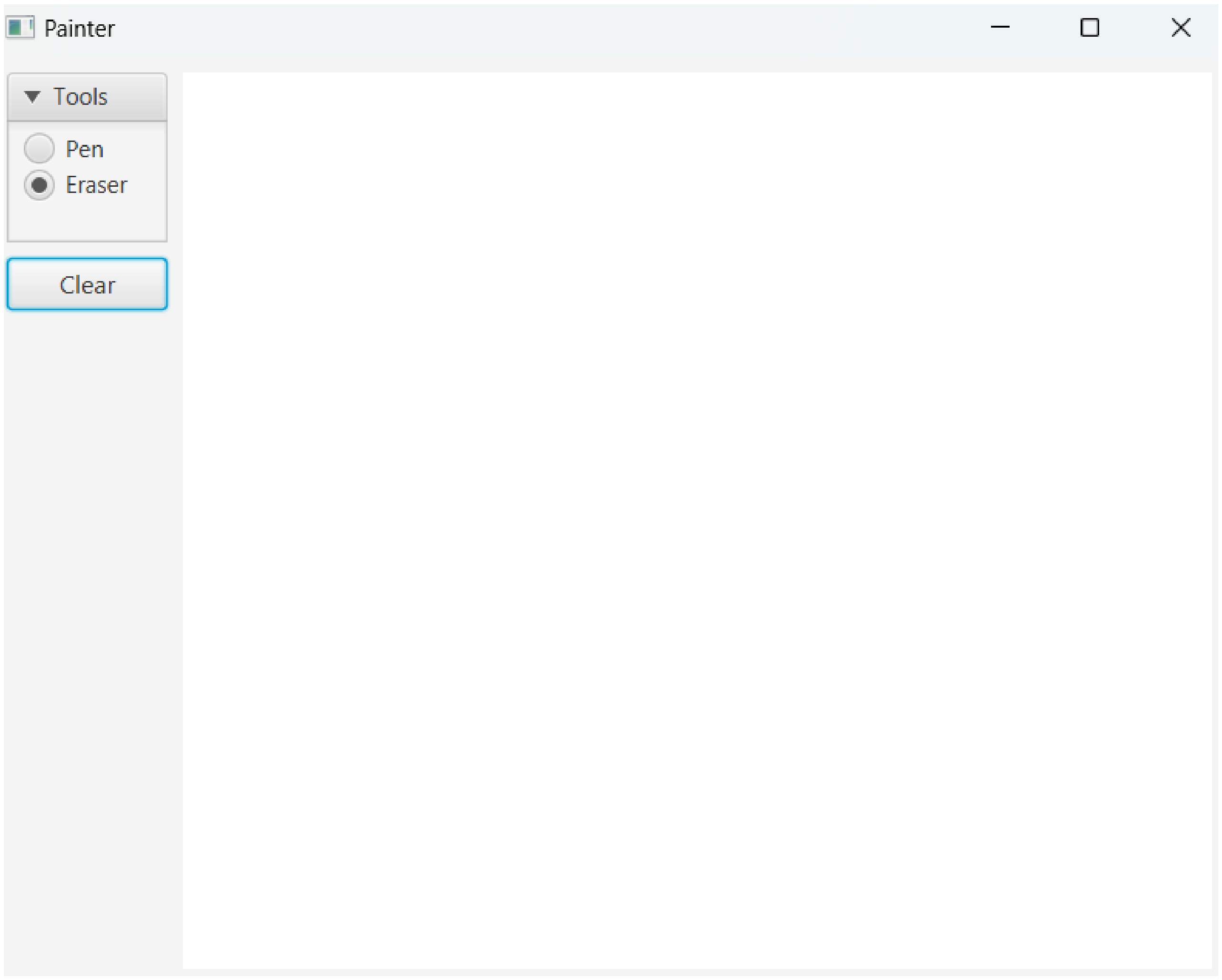
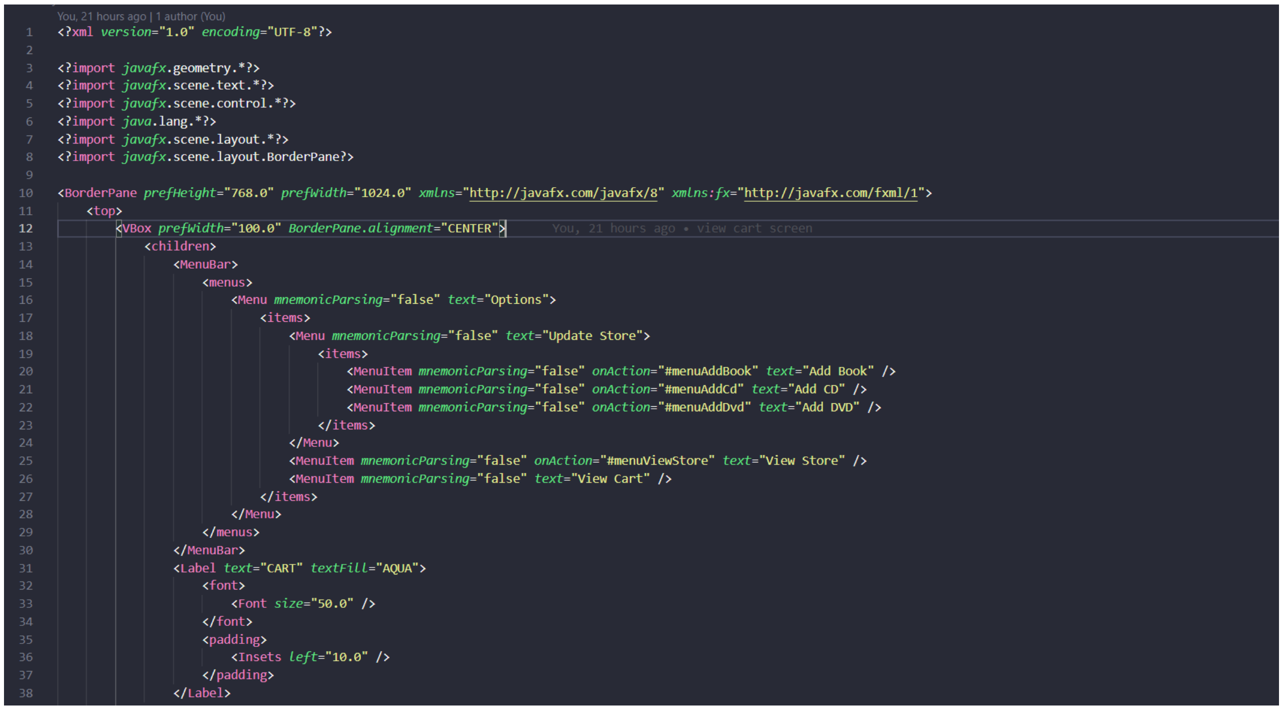


Figure 4.7: Clear button

5 View Cart Screen

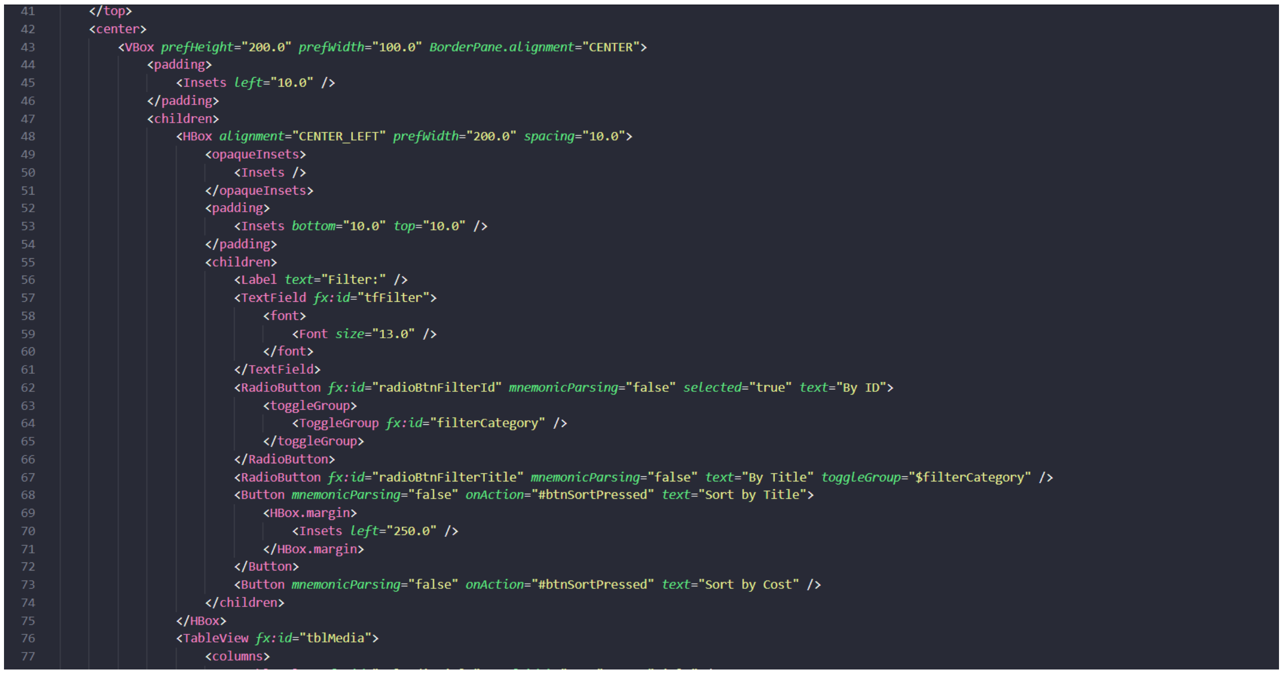
5.1 Create cart.fxml



You, 21 hours ago | 1 author (You)

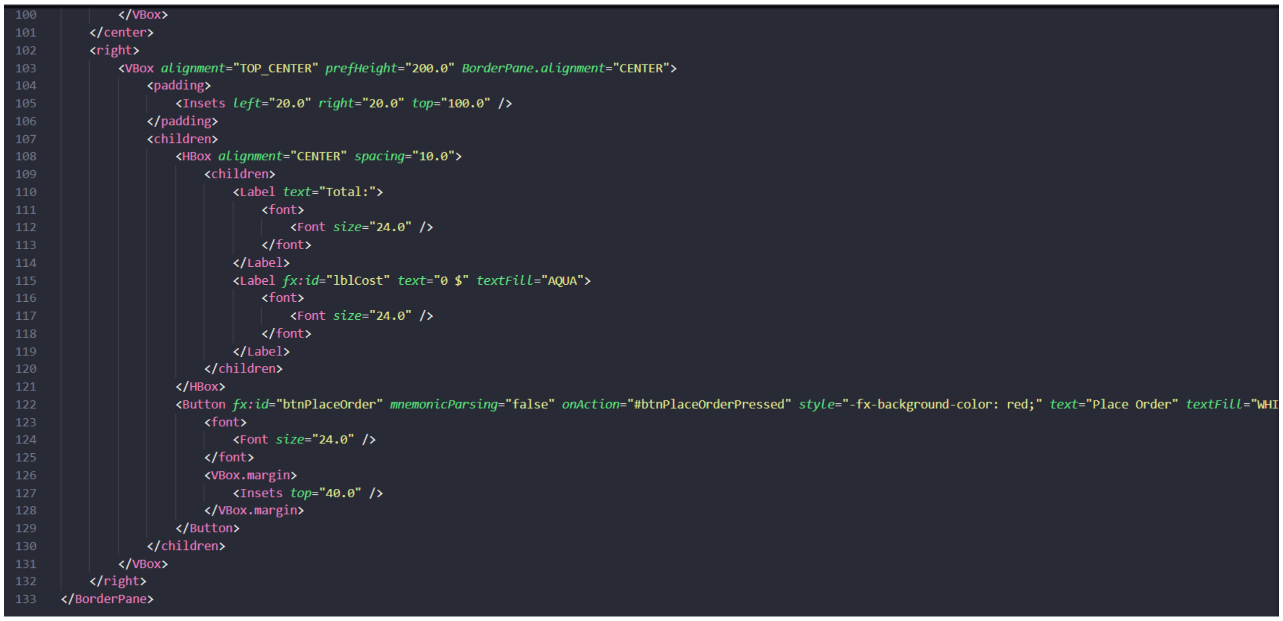
```xml
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<BorderPane prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">
    <top>
        <VBox prefWidth="100.0" BorderPane.alignment="CENTER">      You, 21 hours ago • view cart screen
            <children>
                <MenuBar>
                    <menus>
                        <Menu mnemonicParsing="false" text="Options">
                            <items>
                                <Menu mnemonicParsing="false" text="Update Store">
                                    <items>
                                        <MenuItem mnemonicParsing="false" onAction="#menuAddBook" text="Add Book" />
                                        <MenuItem mnemonicParsing="false" onAction="#menuAddCd" text="Add CD" />
                                        <MenuItem mnemonicParsing="false" onAction="#menuAddDvd" text="Add DVD" />
                                    </items>
                                </Menu>
                                <MenuItem mnemonicParsing="false" onAction="#menuViewStore" text="View Store" />
                                <MenuItem mnemonicParsing="false" text="View Cart" />
                            </items>
                        </Menu>
                    </menus>
                </MenuBar>
                <Label text="CART" textFill="AQUA">
                    <font>
                        <Font size="50.0" />
                    </font>
                    <padding>
                        <Insets left="10.0" />
                    </padding>
                </Label>
```

Figure 5.1: Cart.fxml 1



```xml
    </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">
                    <opaqueInsets>
                        <Insets />
                    </opaqueInsets>
                    <padding>
                        <Insets bottom="10.0" top="10.0" />
                    </padding>
                    <children>
                        <Label text="Filter:" />
                        <TextField fx:id="tfFilter">
                            <font>
                                <Font size="13.0" />
                            </font>
                        </TextField>
                        <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" />
                        <Button mnemonicParsing="false" onAction="#btnSortPressed" text="Sort by Title">
                            <HBox.margin>
                                <Insets left="250.0" />
                            </HBox.margin>
                        </Button>
                        <Button mnemonicParsing="false" onAction="#btnSortPressed" text="Sort by Cost" />
                    </children>
                </HBox>
                <TableView fx:id="tblMedia">
                    <columns>
```

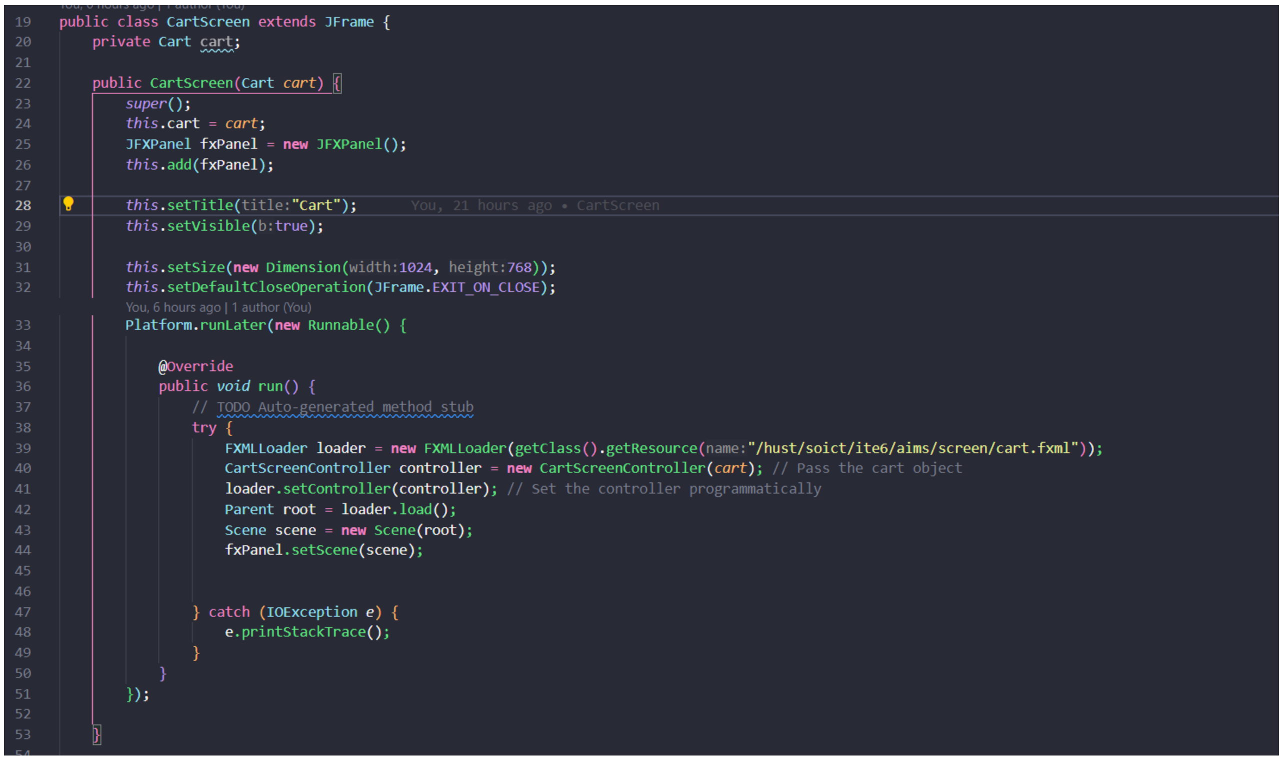
Figure 5.2: Cart.fxml 2



```xml
            </VBox>
        </center>
        <right>
            <VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
                <padding>
                    <Insets left="20.0" right="20.0" top="100.0" />
                </padding>
                <children>
                    <HBox alignment="CENTER" spacing="10.0">
                        <children>
                            <Label text="Total:">
                                <font>
                                    <Font size="24.0" />
                                </font>
                            </Label>
                            <Label fx:id="lblCost" text="0 $" textFill="AQUA">
                                <font>
                                    <Font size="24.0" />
                                </font>
                            </Label>
                        </children>
                    </HBox>
                    <Button fx:id="btnPlaceOrder" mnemonicParsing="false" onAction="#btnPlaceOrderPressed" style="-fx-background-color: red;" text="Place Order" textFill="WHI
                        <font>
                            <Font size="24.0" />
                        </font>
                        <VBox.margin>
                            <Insets top="40.0" />
                        </VBox.margin>
                    </Button>
                </children>
            </VBox>
        </right>
</BorderPane>
```

Figure 5.3: Cart.fxml 3

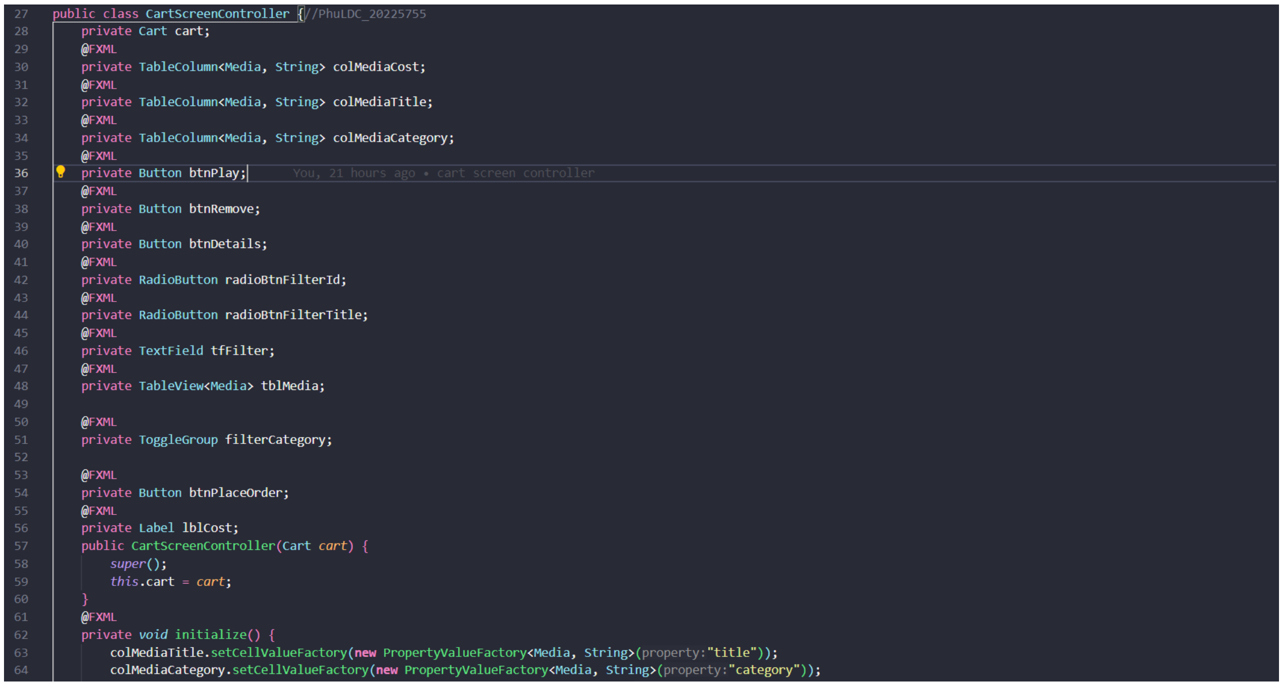
5.2 Create class CartScreen



```java
public class CartScreen extends JFrame {
    private Cart cart;

    public CartScreen(Cart cart) {
        super();
        this.cart = cart;
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);

        this.setTitle(title:"Cart");        You, 21 hours ago • CartScreen
        this.setVisible(b:true);

        this.setSize(new Dimension(width:1024, height:768));
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        You, 6 hours ago | 1 author (You)
        Platform.runLater(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"/hust/soict/ite6/aims/screen/cart.fxml"));
                    CartScreenController controller = new CartScreenController(cart); // Pass the cart object
                    loader.setController(controller); // Set the controller programmatically
                    Parent root = loader.load();
                    Scene scene = new Scene(root);
                    fxPanel.setScene(scene);


                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });

    }
```

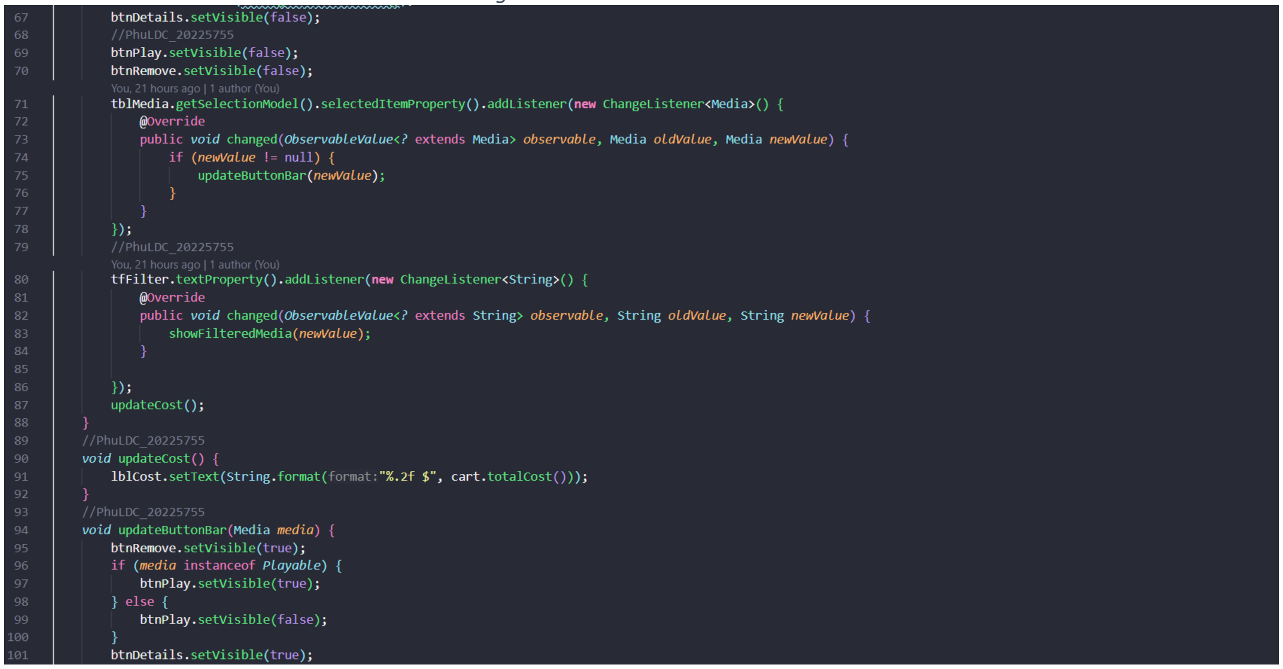
Figure 5.4: CartScreen class

5.3 Create class CartScreenController



```java
public class CartScreenController {//PhuLDC_20225755
    private Cart cart;
    @FXML
    private TableColumn<Media, String> colMediaCost;
    @FXML
    private TableColumn<Media, String> colMediaTitle;
    @FXML
    private TableColumn<Media, String> colMediaCategory;
    @FXML
    private Button btnPlay;      You, 21 hours ago • cart screen controller
    @FXML
    private Button btnRemove;
    @FXML
    private Button btnDetails;
    @FXML
    private RadioButton radioBtnFilterId;
    @FXML
    private RadioButton radioBtnFilterTitle;
    @FXML
    private TextField tfFilter;
    @FXML
    private TableView<Media> tblMedia;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private Button btnPlaceOrder;
    @FXML
    private Label lblCost;
    public CartScreenController(Cart cart) {
        super();
        this.cart = cart;
    }
    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>(property:"title"));
        colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>(property:"category"));
```

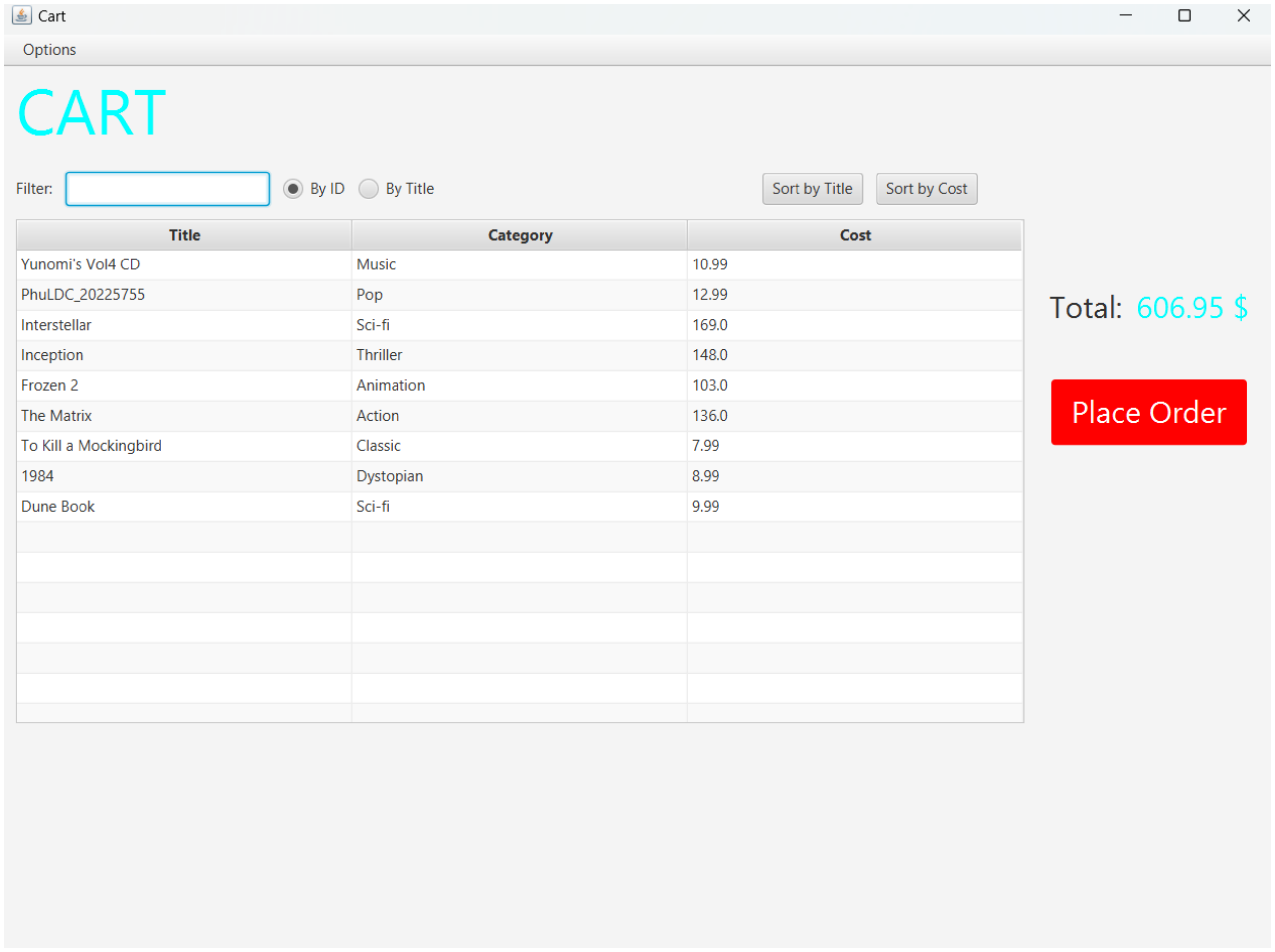
Figure 5.5: CartScreenController 1



```java
        btnDetails.setVisible(false);
        //PhuLDC_20225755
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);
        You, 21 hours ago | 1 author (You)
        tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
            }
        });
        //PhuLDC_20225755
        You, 21 hours ago | 1 author (You)
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
                showFilteredMedia(newValue);
            }

        });
        updateCost();
    }
    //PhuLDC_20225755
    void updateCost() {
        lblCost.setText(String.format(format:"%.2f $", cart.totalCost()));
    }
    //PhuLDC_20225755
    void updateButtonBar(Media media) {
        btnRemove.setVisible(true);
        if (media instanceof Playable) {
            btnPlay.setVisible(true);
        } else {
            btnPlay.setVisible(false);
        }
        btnDetails.setVisible(true);
```

Figure 5.6: CartScreenController 2

5.4 Demo

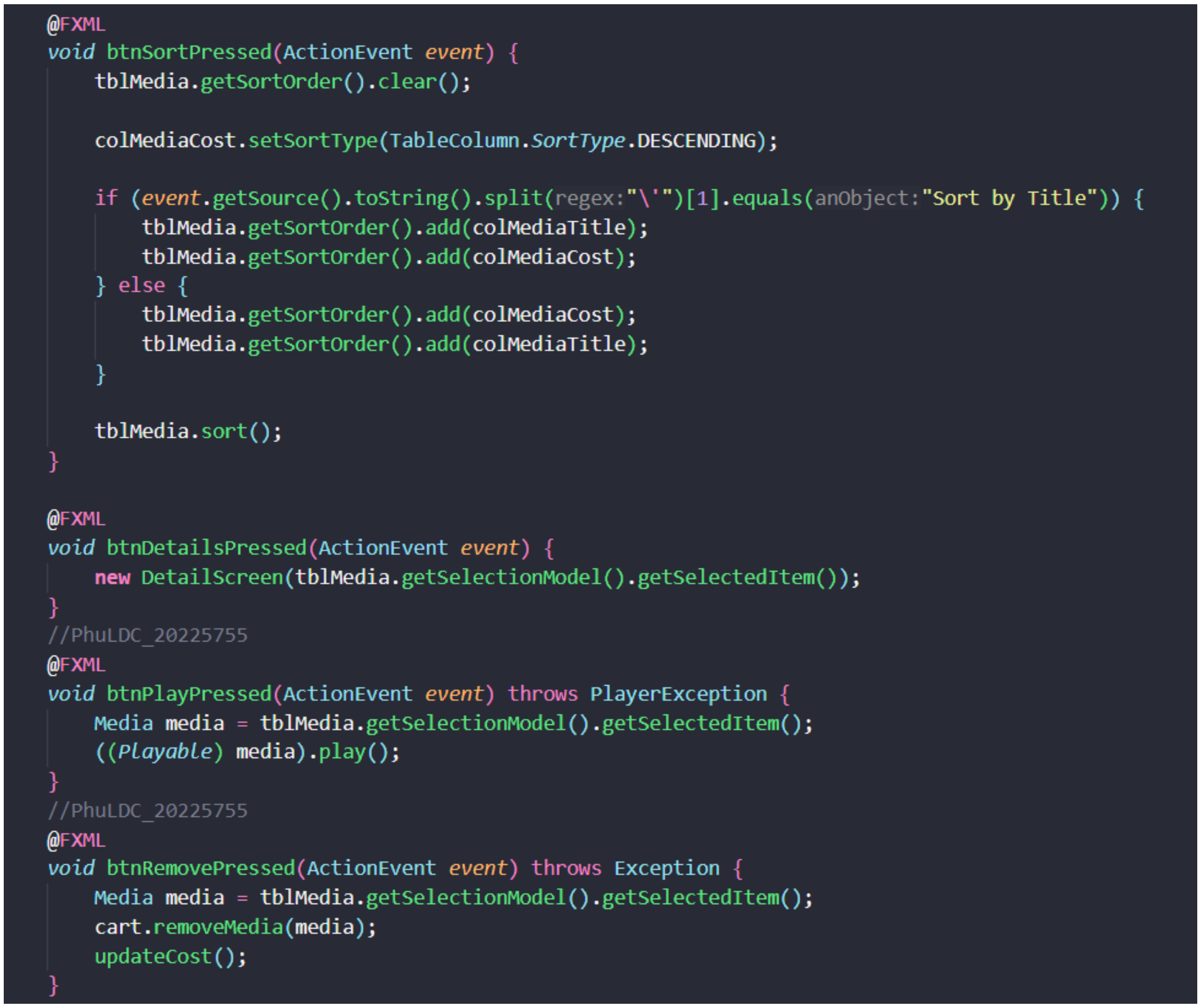


Title	Category	Cost
Yunomi's Vol4 CD	Music	10.99
PhuLDC_20225755	Pop	12.99
Interstellar	Sci-fi	169.0
Inception	Thriller	148.0
Frozen 2	Animation	103.0
The Matrix	Action	136.0
To Kill a Mockingbird	Classic	7.99
1984	Dystopian	8.99
Dune Book	Sci-fi	9.99

Figure 5.7: Demo CartScreen

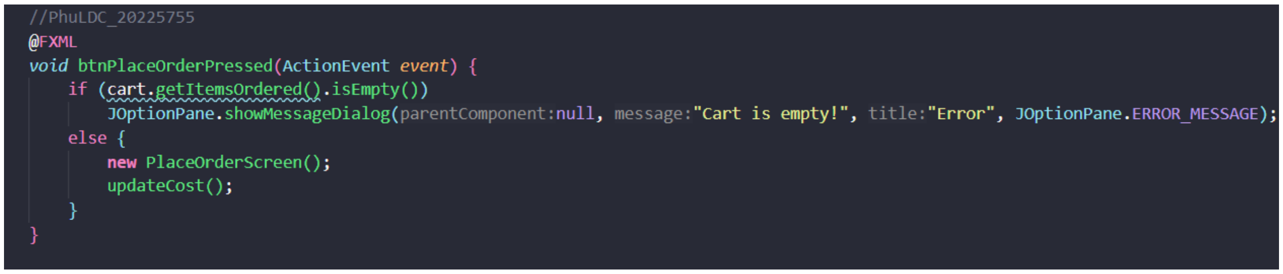
6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController



```
@FXML
void btnSortPressed(ActionEvent event) {
    tblMedia.getSortOrder().clear();

    colMediaCost.setSortType(TableColumn.SortType.DESCENDING);

    if (event.getSource().toString().split(regex:"\'")[1].equals(anObject:"Sort by Title")) {
        tblMedia.getSortOrder().add(colMediaTitle);
        tblMedia.getSortOrder().add(colMediaCost);
    } else {
        tblMedia.getSortOrder().add(colMediaCost);
        tblMedia.getSortOrder().add(colMediaTitle);
    }

    tblMedia.sort();
}

@FXML
void btnDetailsPressed(ActionEvent event) {
    new DetailScreen(tblMedia.getSelectionModel().getSelectedItem());
}
//PhuLDC_20225755
@FXML
void btnPlayPressed(ActionEvent event) throws PlayerException {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    ((Playable) media).play();
}
//PhuLDC_20225755
@FXML
void btnRemovePressed(ActionEvent event) throws Exception {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    updateCost();
}
```

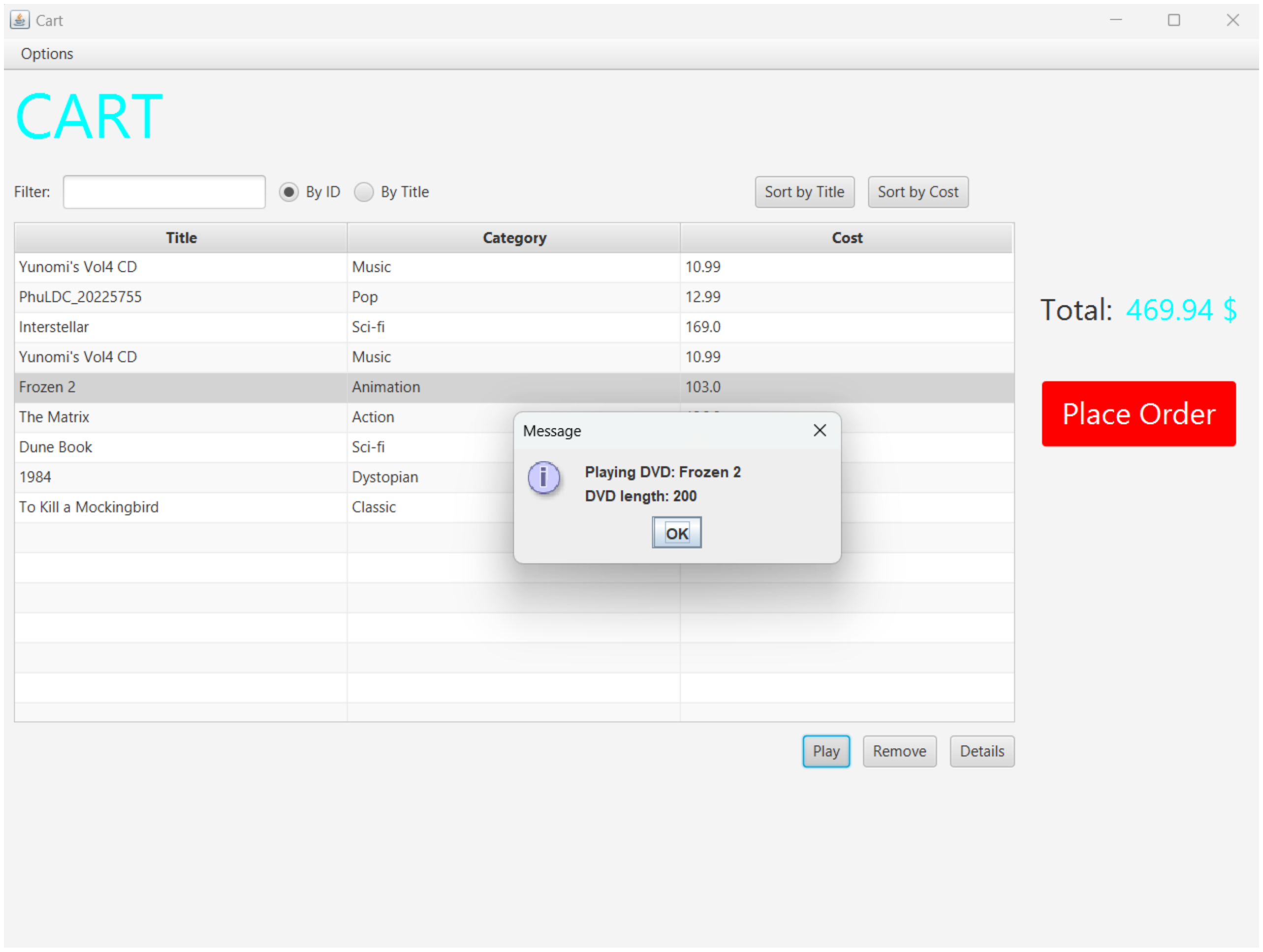
Figure 6.1: CartScreenController 1



```
//PhuLDC_20225755
@FXML
void btnPlaceOrderPressed(ActionEvent event) {
    if (cart.getItemsOrdered().isEmpty())
        JOptionPane.showMessageDialog(parentComponent:null, message:"Cart is empty!", title:"Error", JOptionPane.ERROR_MESSAGE);
    else {
        new PlaceOrderScreen();
        updateCost();
    }
}
```

Figure 6.2: CartScreenController 2

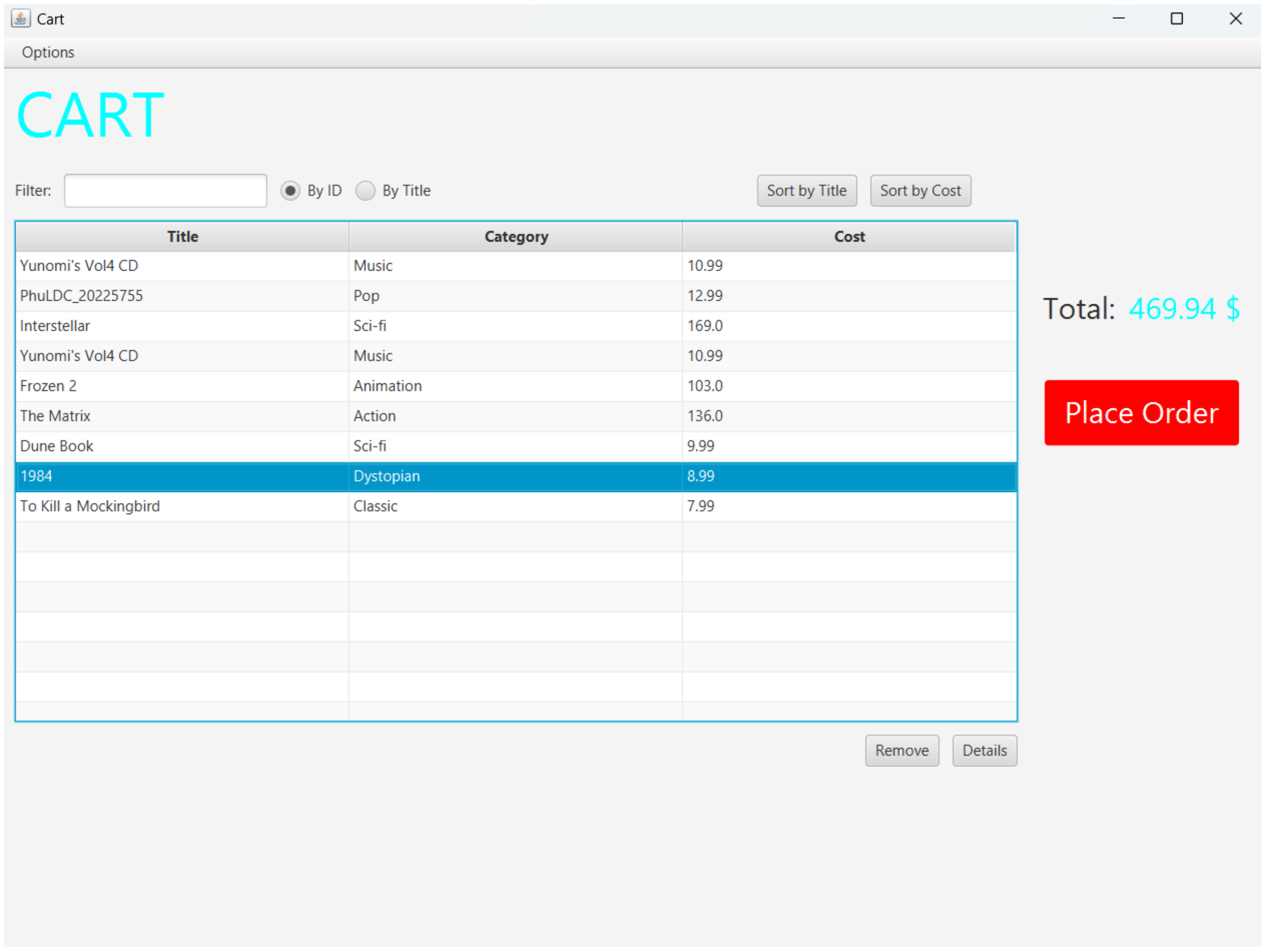
6.2 Demo



Title	Category	Cost
Yunomi's Vol4 CD	Music	10.99
PhuLDC_20225755	Pop	12.99
Interstellar	Sci-fi	169.0
Yunomi's Vol4 CD	Music	10.99
Frozen 2	Animation	103.0
The Matrix	Action	
Dune Book	Sci-fi	
1984	Dystopian	
To Kill a Mockingbird	Classic	

Playing DVD: Frozen 2
DVD length: 200

Figure 6.3: Demo media playable

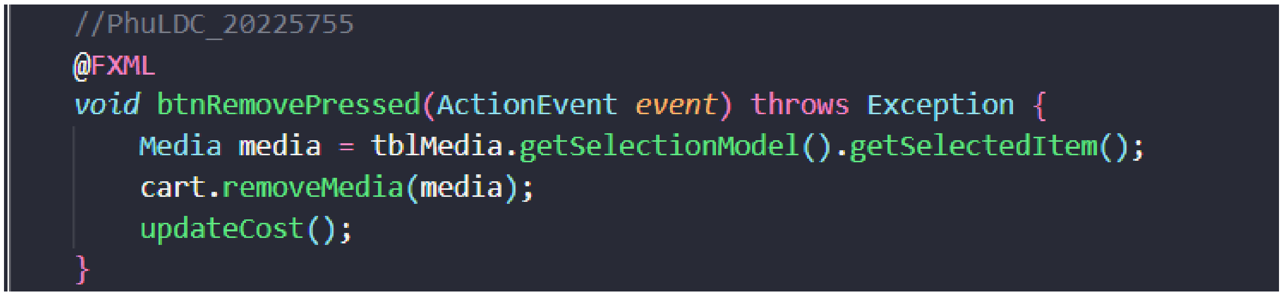


Title	Category	Cost
Yunomi's Vol4 CD	Music	10.99
PhuLDC_20225755	Pop	12.99
Interstellar	Sci-fi	169.0
Yunomi's Vol4 CD	Music	10.99
Frozen 2	Animation	103.0
The Matrix	Action	136.0
Dune Book	Sci-fi	9.99
1984	Dystopian	8.99
To Kill a Mockingbird	Classic	7.99

Figure 6.4: Demo media unplayable

7 Deleting a media

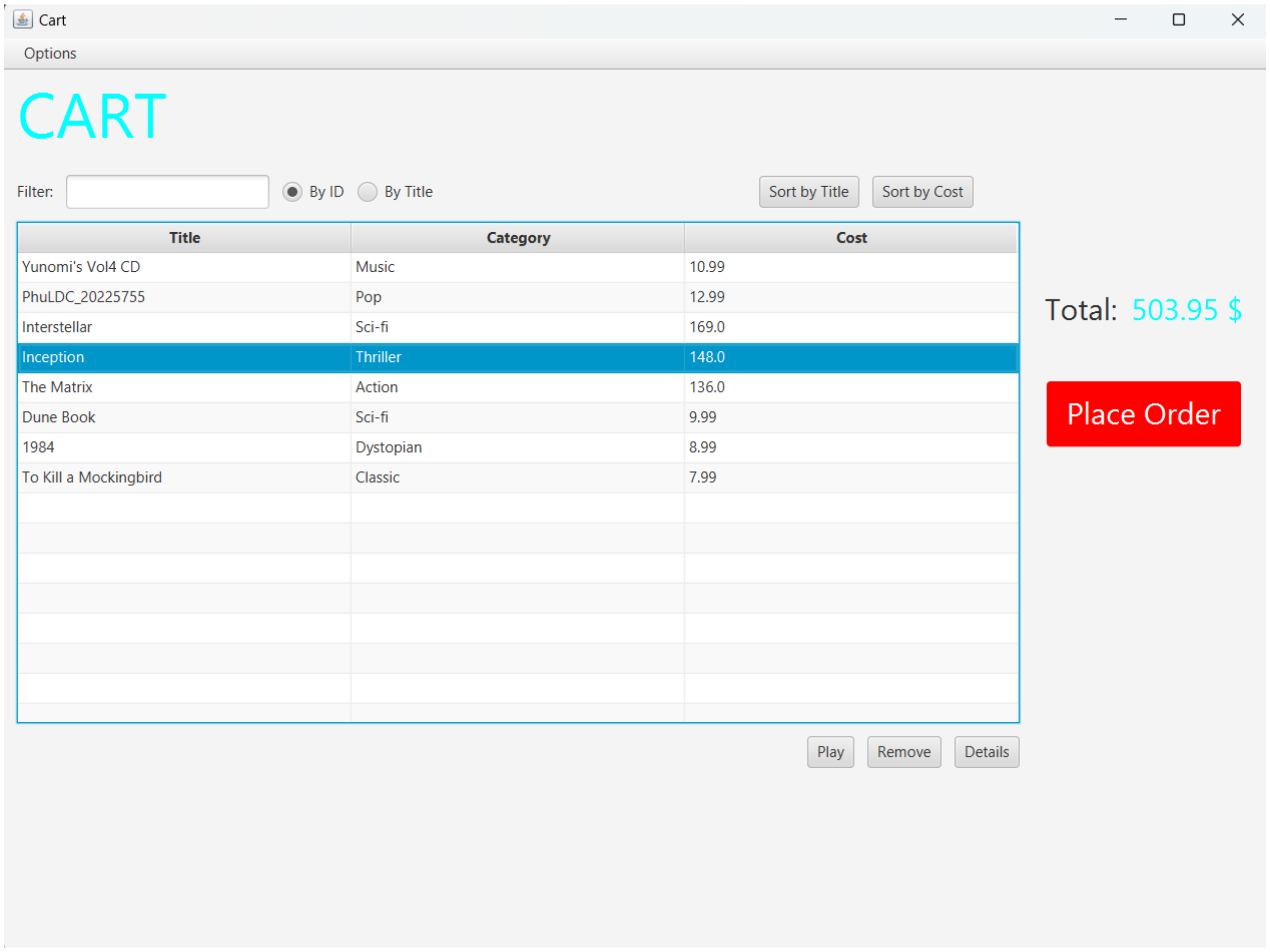
7.1 Code



```
//PhuLDC_20225755
@FXML
void btnRemovePressed(ActionEvent event) throws Exception {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    updateCost();
}
```

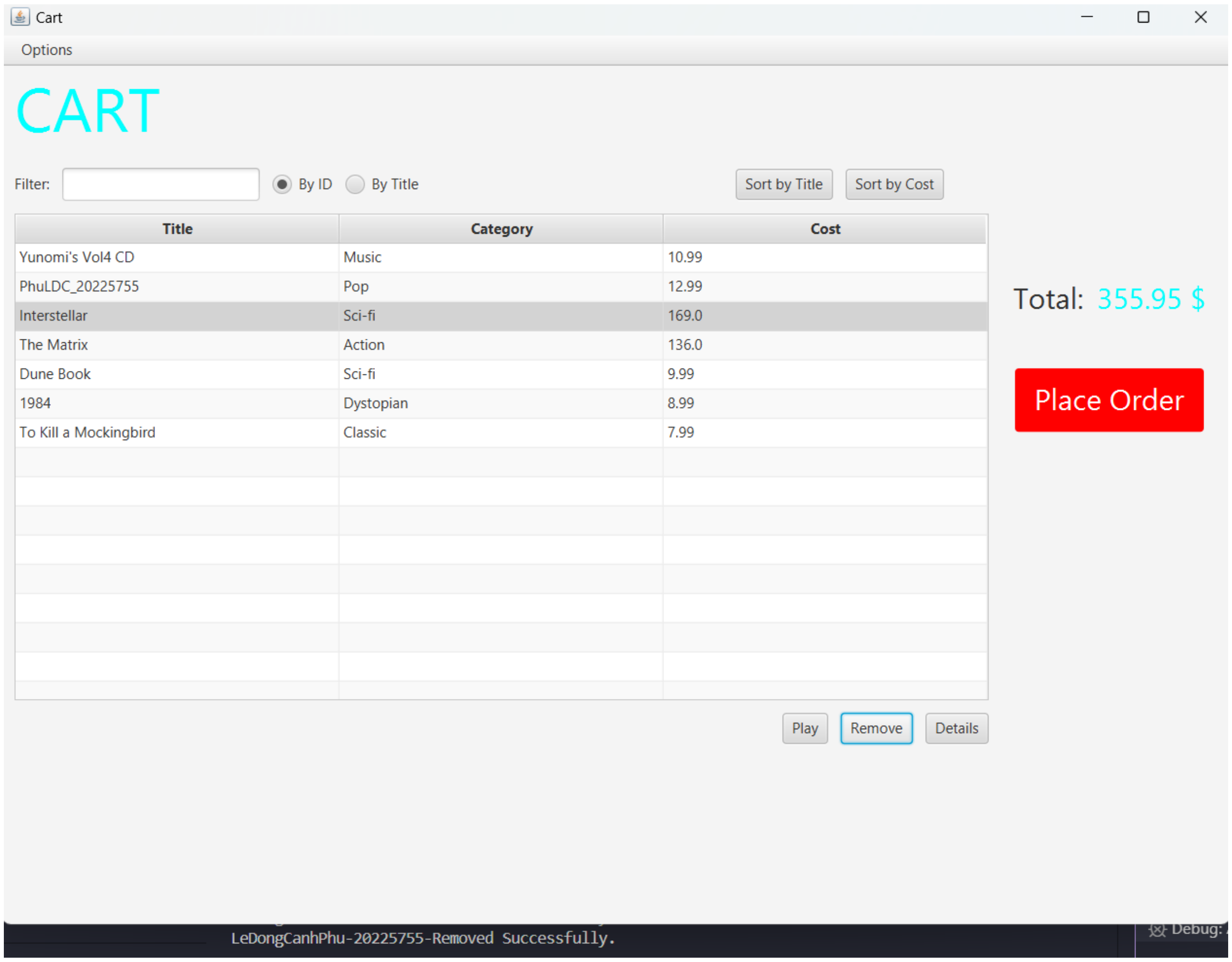
Figure 7.1: btnRemovePressed Method

7.2 Demo



Title	Category	Cost
Yunomi's Vol4 CD	Music	10.99
PhuLDC_20225755	Pop	12.99
Interstellar	Sci-fi	169.0
Inception	Thriller	148.0
The Matrix	Action	136.0
Dune Book	Sci-fi	9.99
1984	Dystopian	8.99
To Kill a Mockingbird	Classic	7.99

Figure 7.2: button Remove



Title	Category	Cost
Yunomi's Vol4 CD	Music	10.99
PhuLDC_20225755	Pop	12.99
Interstellar	Sci-fi	169.0
The Matrix	Action	136.0
Dune Book	Sci-fi	9.99
1984	Dystopian	8.99
To Kill a Mockingbird	Classic	7.99

```
LeDongCanhPhu-20225755-Removed Successfully.
```

Figure 7.3: button Remove

8 Complete the Aims GUI application

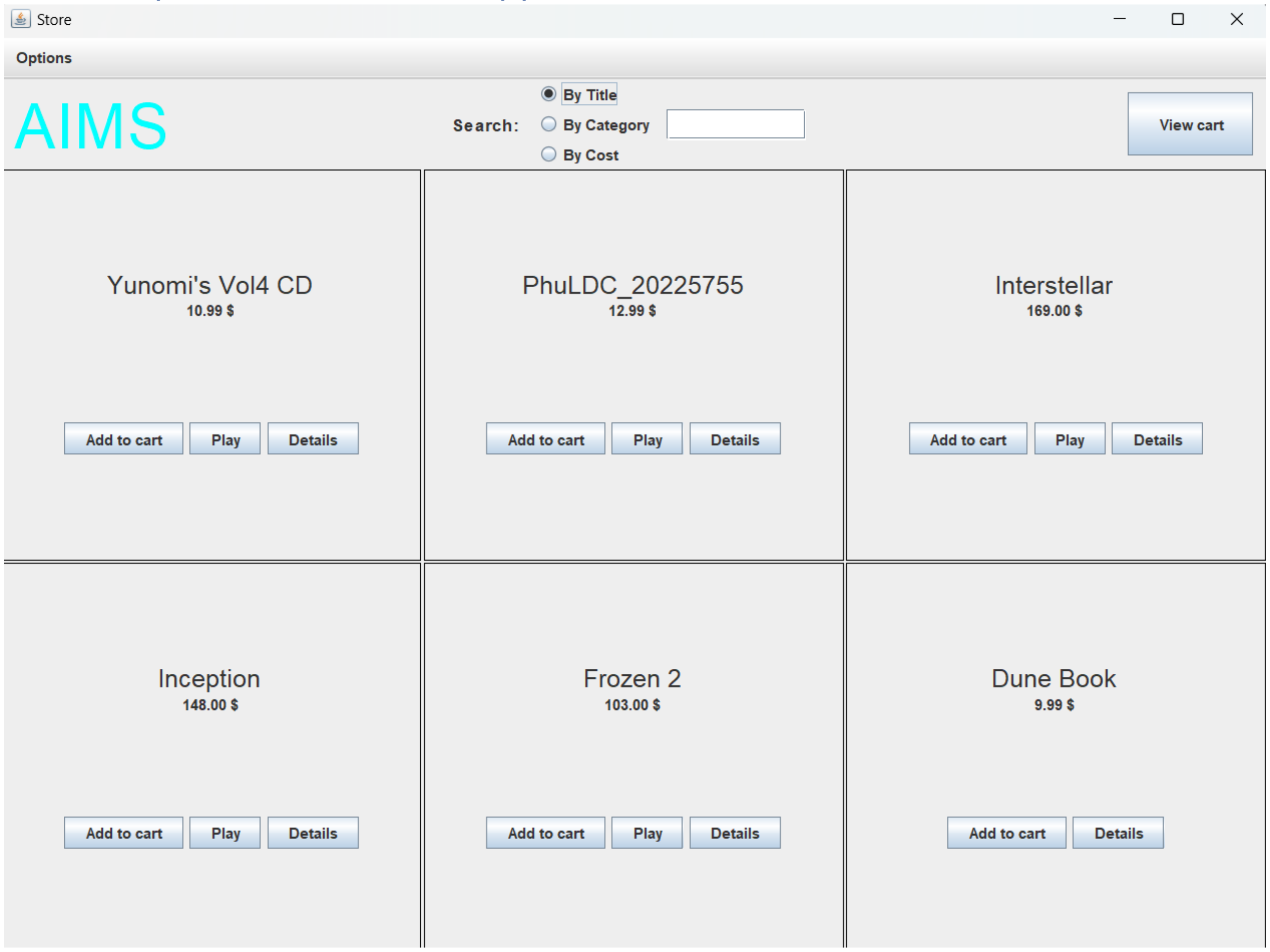
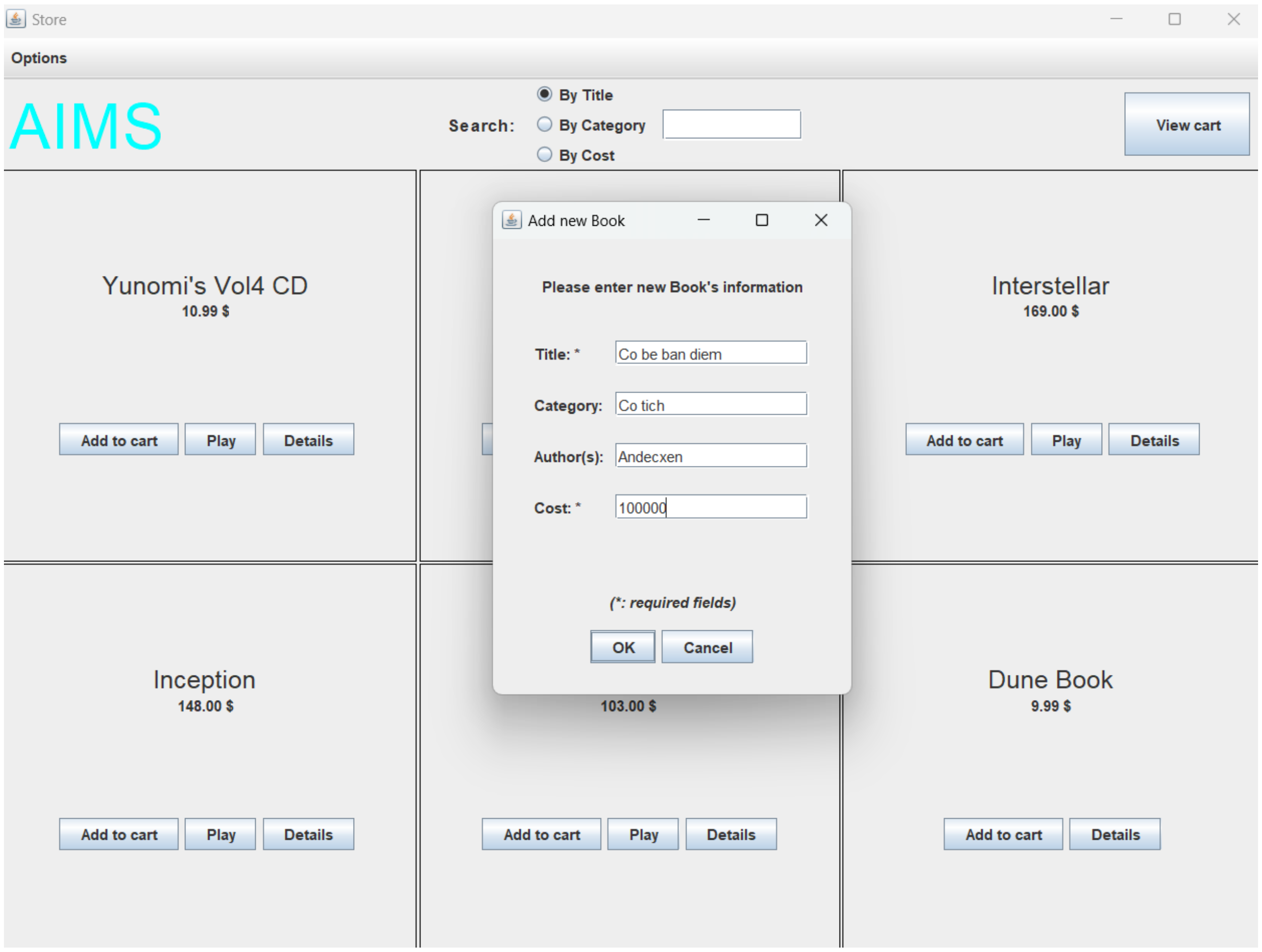


Figure 8.1: Store before add book



Please enter new Book's information

Title: * Co be ban diem

Category: Co tich

Author(s): Andecxen

Cost: * 100000

(: required fields)*

OK Cancel

Figure 8.2: Add book

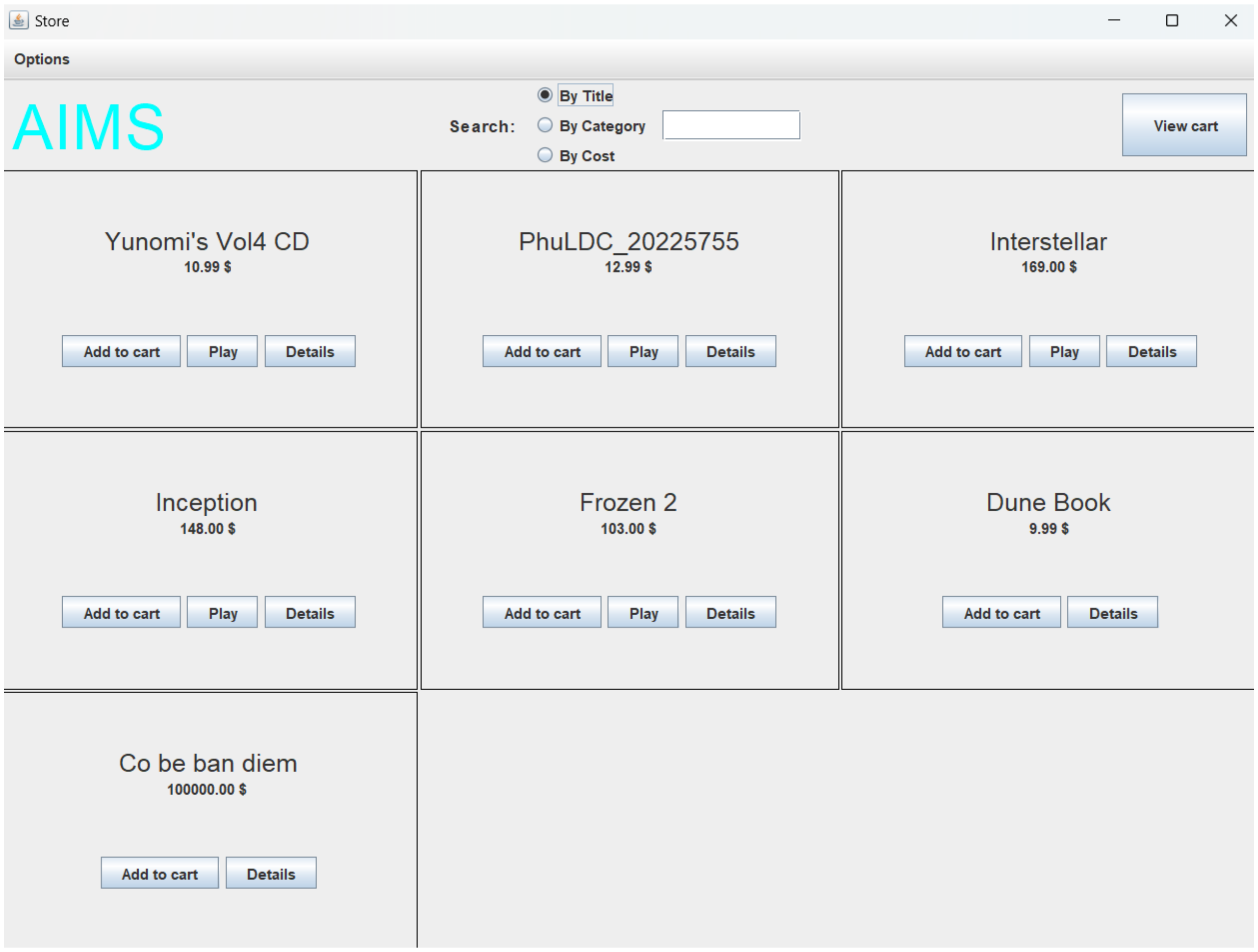
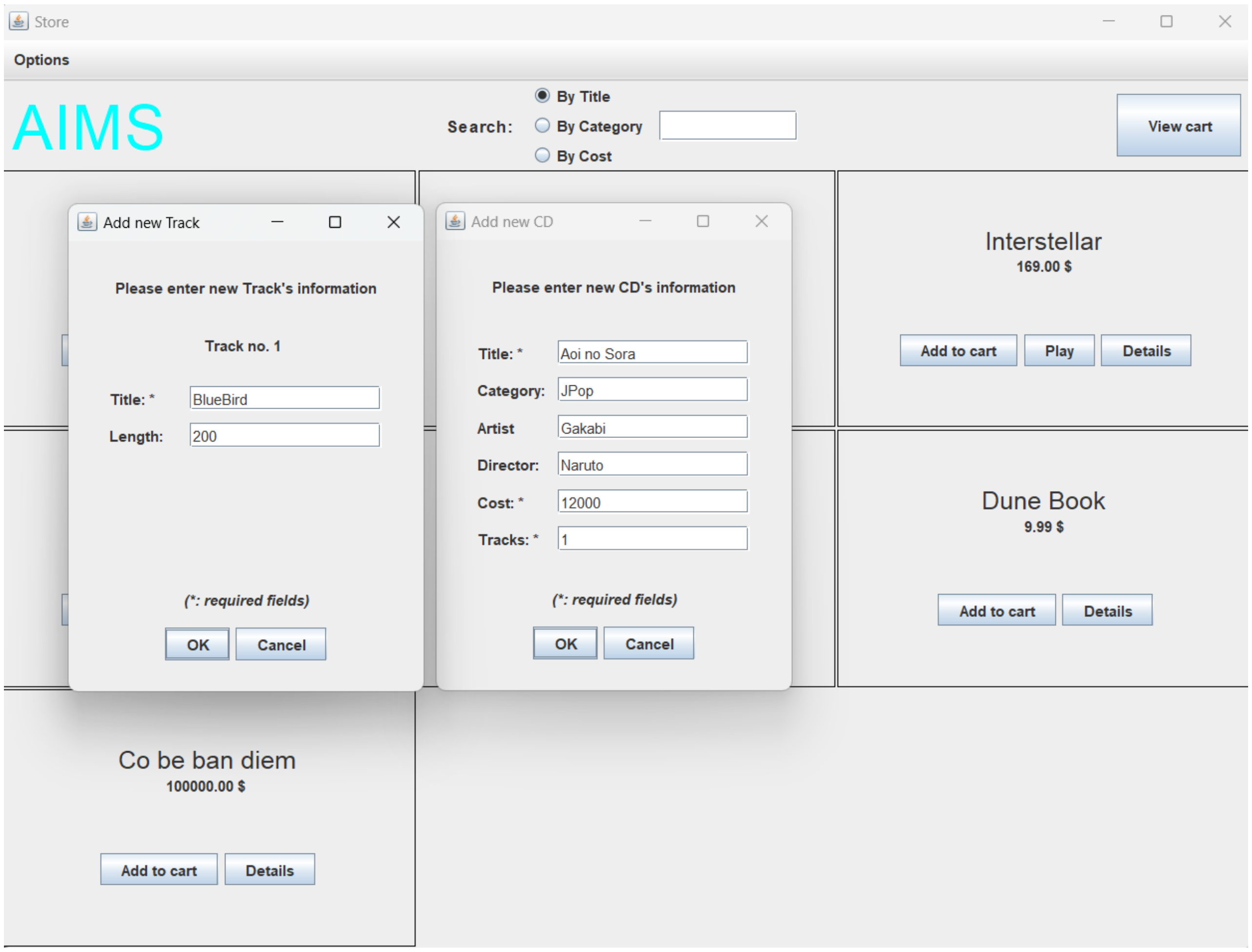


Figure 8.3: Store after add book



Please enter new Track's information

Track no. 1

Title: * BlueBird
Length: 200

(*: required fields)

OK Cancel

Please enter new CD's information

Title: * Aoi no Sora
Category: JPop
Artist Gakabi
Director: Naruto
Cost: * 12000
Tracks: * 1

(*: required fields)

OK Cancel

Figure 8.4: Add CD

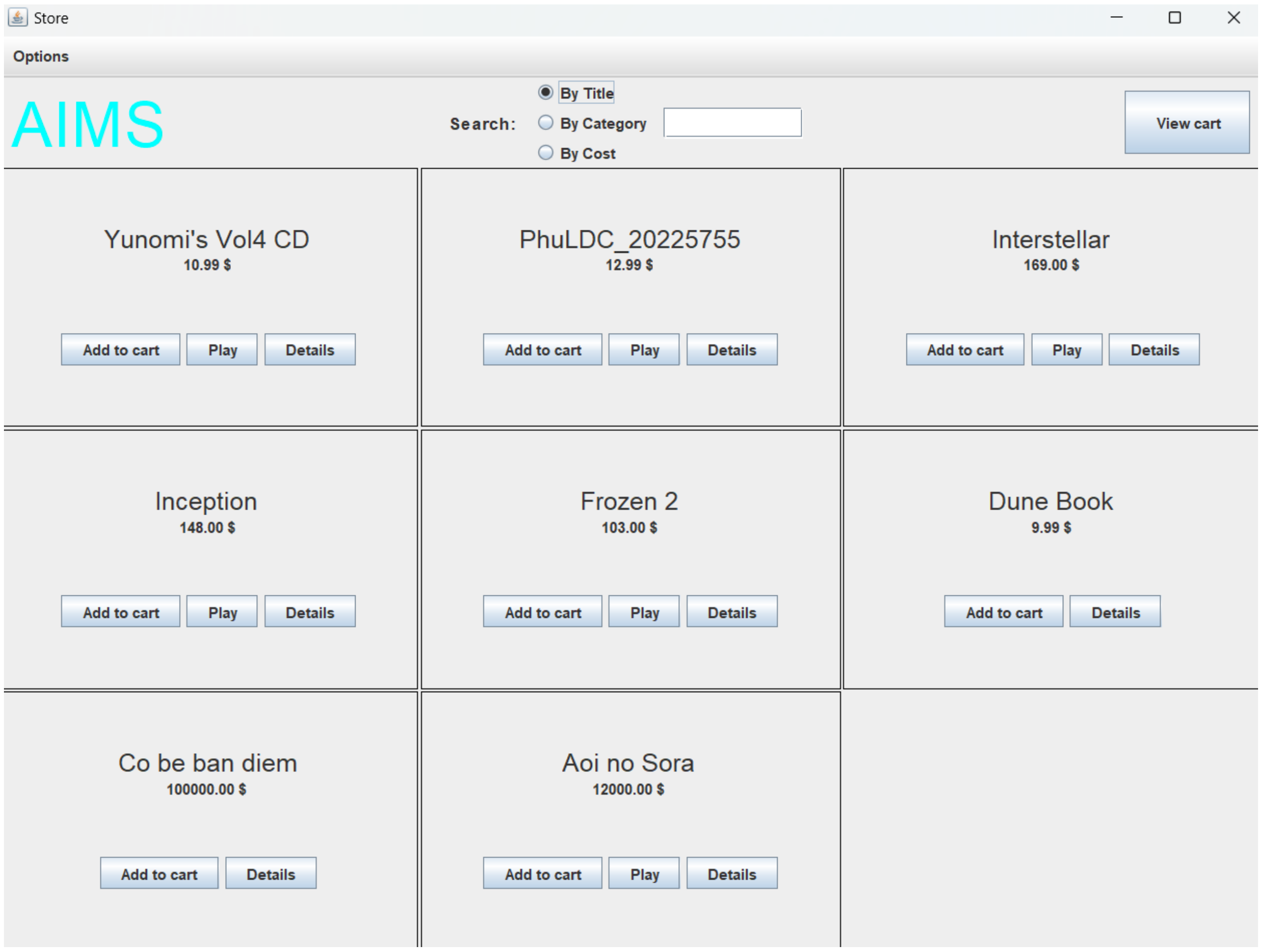
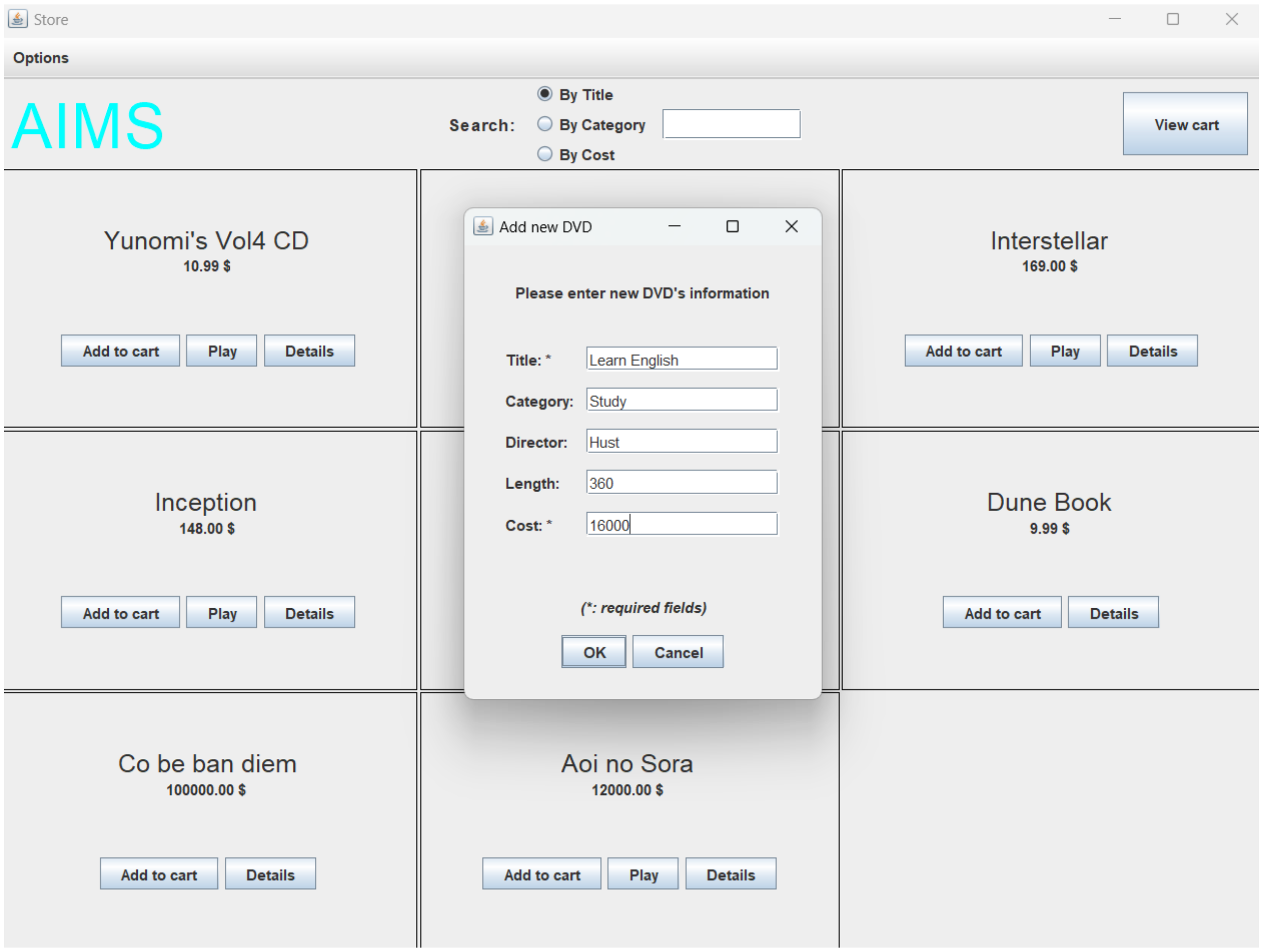


Figure 8.5: Store after add CD



Add new DVD

Please enter new DVD's information

Title: * Learn English

Category: Study

Director: Hust

Length: 360

Cost: * 16000

(*: required fields)

Figure 8.6 Add DVD

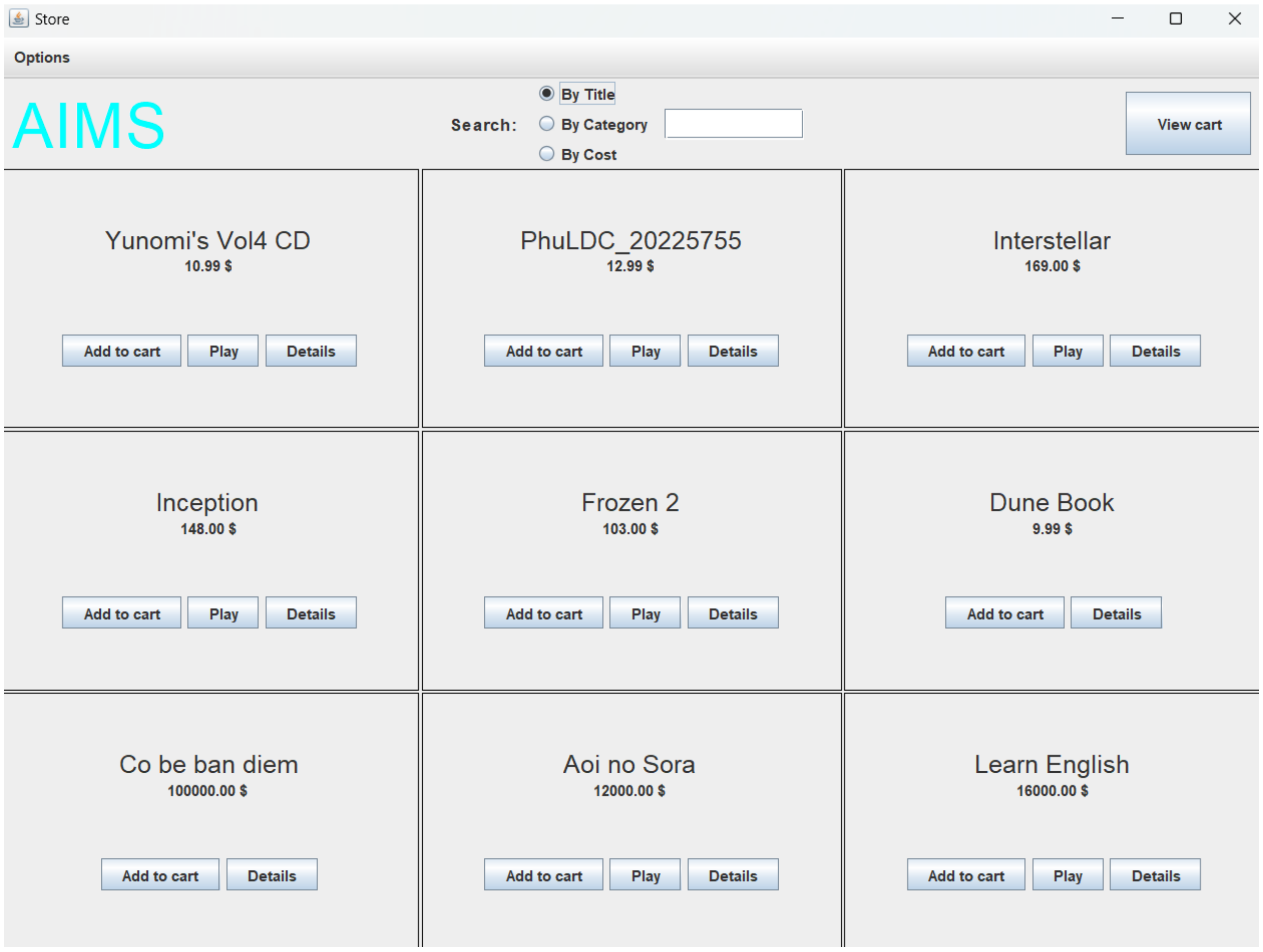
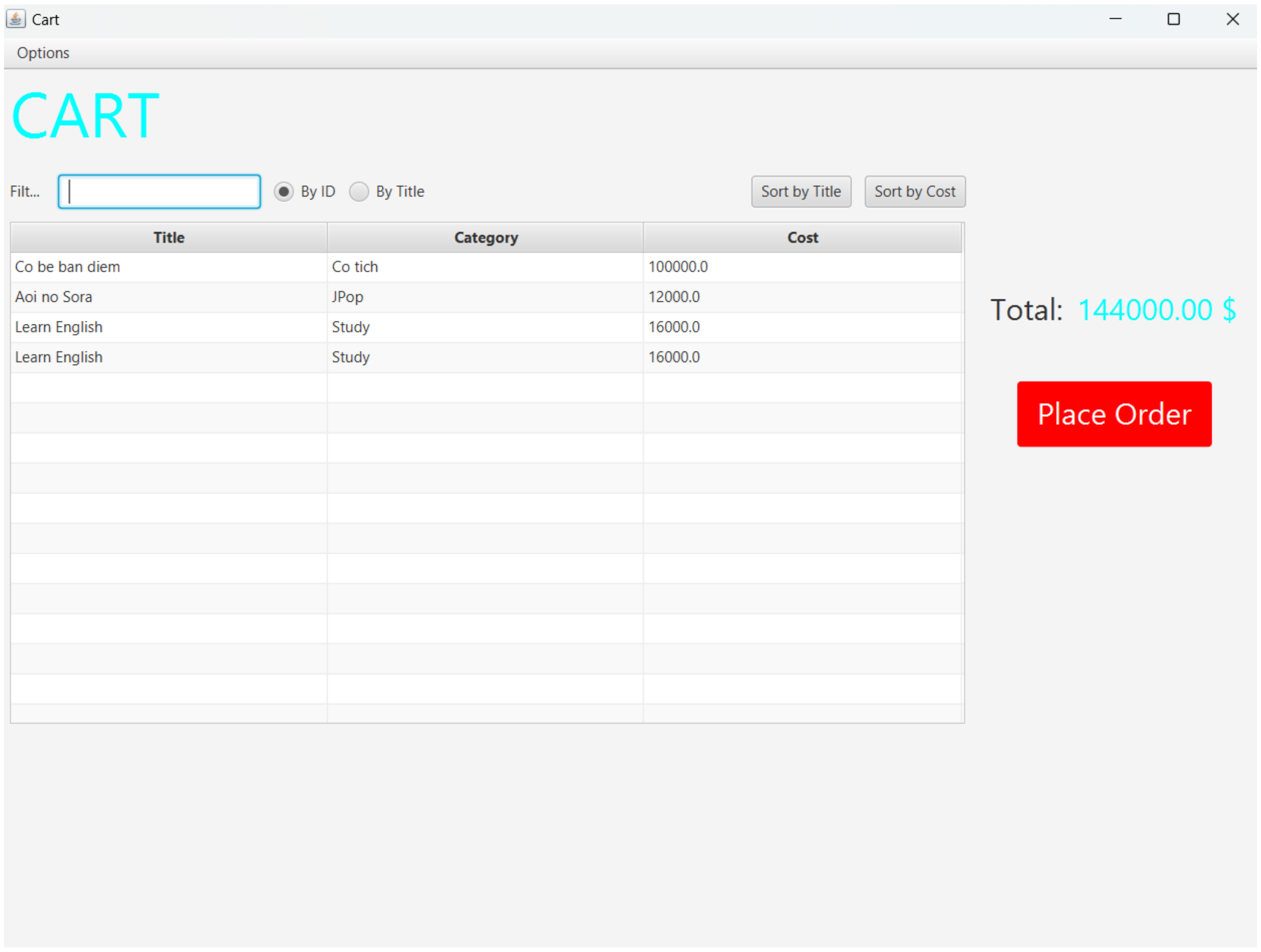
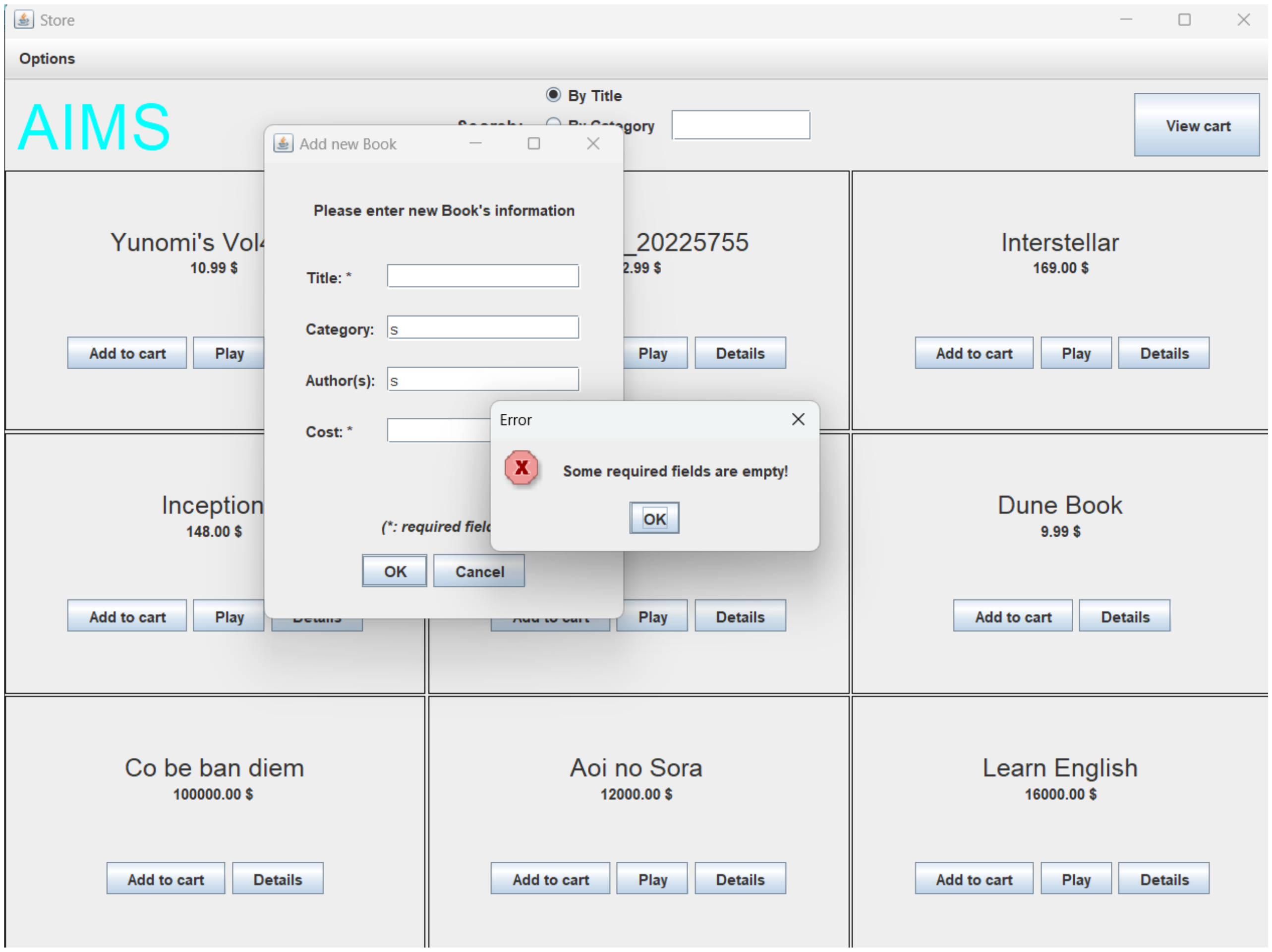


Figure 8.7: Store after add DVD



Title	Category	Cost
Co be ban diem	Co tich	100000.0
Aoi no Sora	JPop	12000.0
Learn English	Study	16000.0
Learn English	Study	16000.0

Figure 8.8: Cart



Please enter new Book's information

Title: *

Category: s

Author(s): s

Cost: *

(*: required fiel

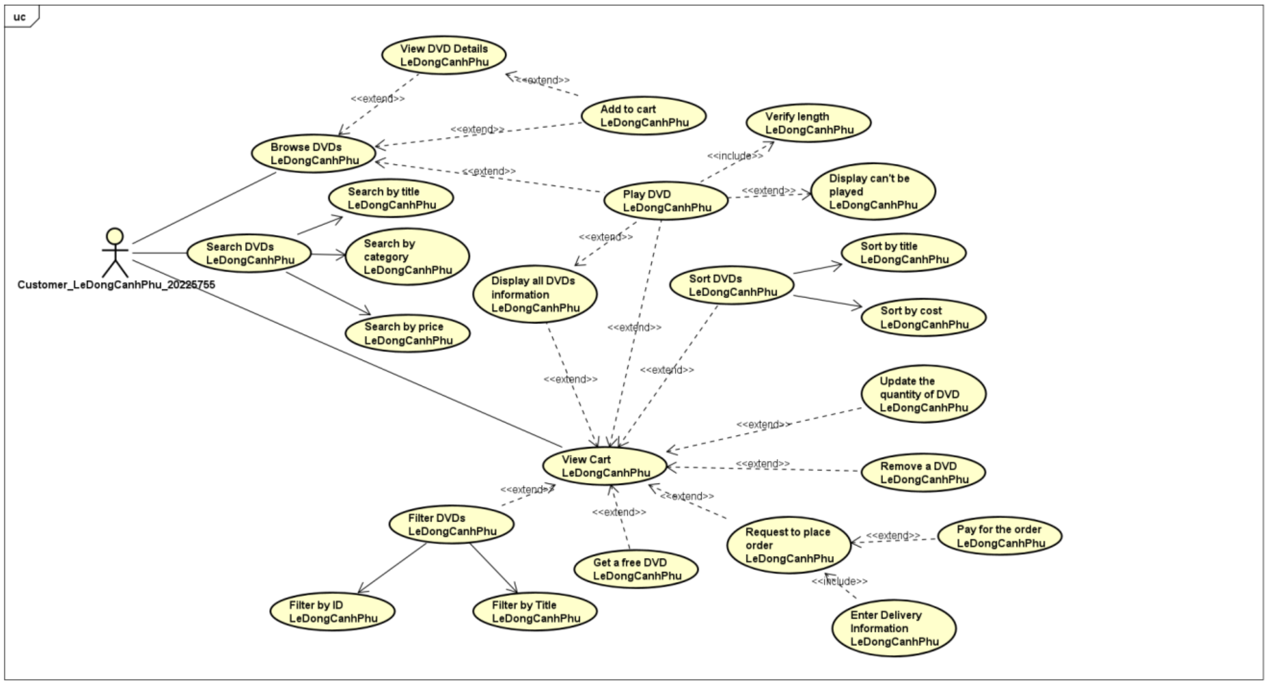
Error

Some required fields are empty!

OK

Figure 8.9: Exception

9 Use case Diagram


uc
Customer_LeDongCanhPhu_20225755
View DVD Details LeDongCanhPhu
Add to cart LeDongCanhPhu
Browse DVDs LeDongCanhPhu
Verify length LeDongCanhPhu
Play DVD LeDongCanhPhu
Display can't be played LeDongCanhPhu
Search by title LeDongCanhPhu
Search DVDs LeDongCanhPhu
Search by category LeDongCanhPhu
Search by price LeDongCanhPhu
Display all DVDs information LeDongCanhPhu
Sort DVDs LeDongCanhPhu
Sort by title LeDongCanhPhu
Sort by cost LeDongCanhPhu
Update the quantity of DVD LeDongCanhPhu
View Cart LeDongCanhPhu
Remove a DVD LeDongCanhPhu
Filter DVDs LeDongCanhPhu
Get a free DVD LeDongCanhPhu
Request to place order LeDongCanhPhu
Pay for the order LeDongCanhPhu
Filter by ID LeDongCanhPhu
Filter by Title LeDongCanhPhu
Enter Delivery Information LeDongCanhPhu
<<extend>>
<<include>>

10 Class Diagram