ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

--- o0o -- -

BÁO CÁO THỰC HÀNH

IT3103-744527-2024.1

BÀI THỰC HÀNH – LAB05

Họ và tên: Lê Quang Khải

MSSV: 20225638

Lớp: VN03-K67

GVHD: Lê Thị Hoa

HTGD: Đặng Mạnh Cường

BÁO CÁO THỰC HÀNH LAB 5 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. Swing components

1.1 AWTAccumulator

```java
public class AWTAccumulator extends Frame {

    private TextField tfInput;  5 usages
    private TextField tfOutput;  4 usages
    private int sum = 0;  2 usages

    public AWTAccumulator() {  1 usage
        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(false);
        add(tfOutput);

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

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

    private class TFInputListener implements ActionListener {  1 usage
        @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

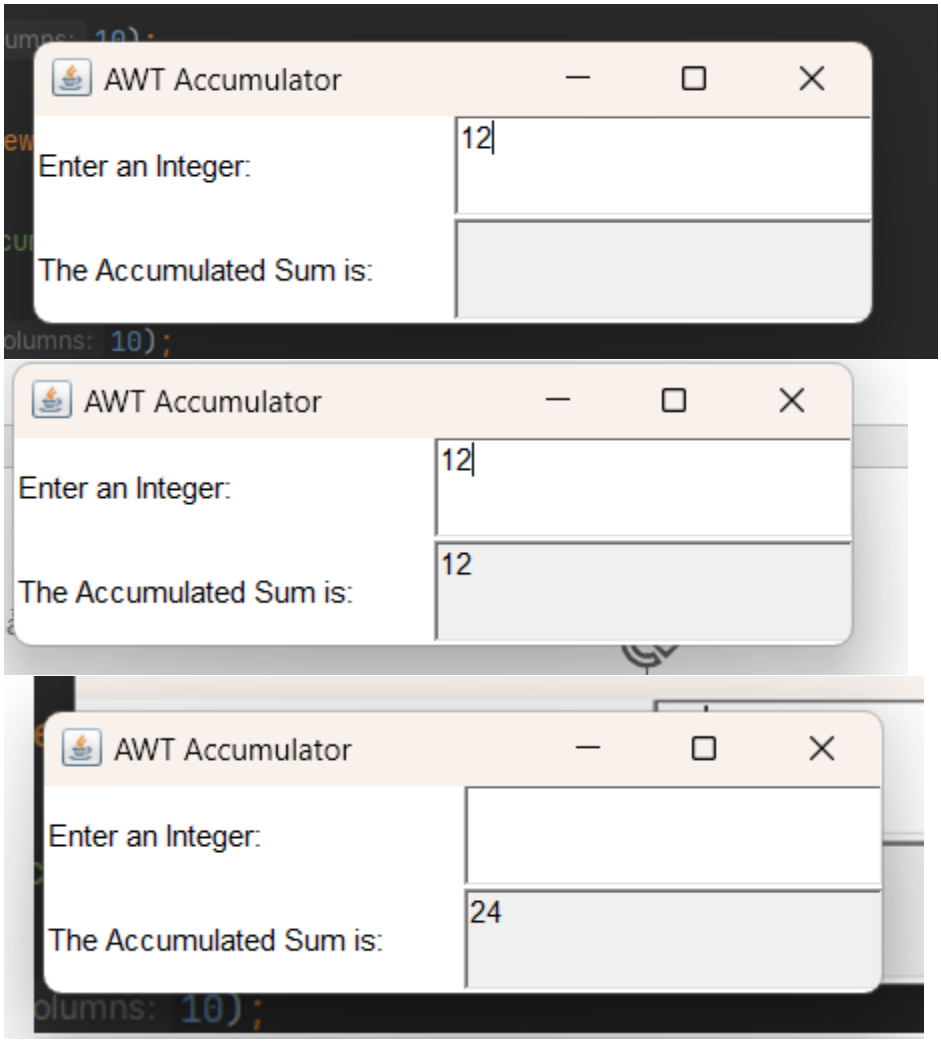


Figure 1.2: Demo of AWTAccumulator

1.2 Swing Accumulator

```java
public class SwingAccumulator extends JFrame {
    private JTextField tfInput;  5 usages
    private JTextField tfOutput;  4 usages
    private int sum = 0;  2 usages

    public SwingAccumulator() {  1 usage
        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(false);
        cp.add(tfOutput);

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

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

    private class TFInputListener implements ActionListener {  1 usage
        @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

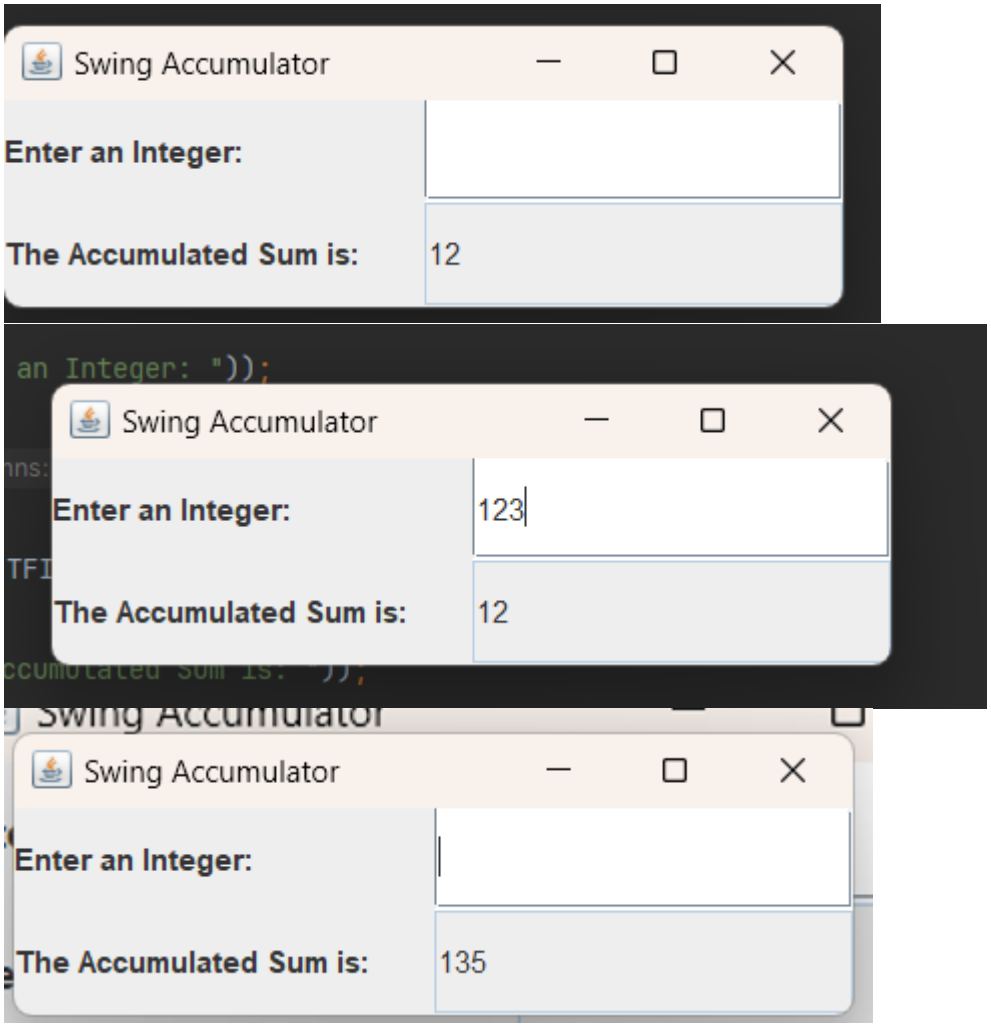


Figure 1.4: Demo of SwingAccumulator

2 Organizing Swing components with Layout Managers

2.1 Code

```java
public class NumberGrid extends JFrame {
    private JButton[] btnNumbers = new JButton[10];  6 usages
    private JButton btnDelete, btnReset;  3 usages
    private JTextField tfDisplay;  8 usages

    public NumberGrid() {  1 usage
        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("Number Grid");
        setSize( width: 200,  height: 200);
        setVisible(true);
    }

    void addButtons(JPanel panelButtons) {  1 usage
        ButtonListener btnListener = new ButtonListener();
        for (int i = 1; i <= 9; i++) {
            btnNumbers[i] = new JButton( text: "" + 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]);
```

Figure 2.1: Source code of NumberGrid 1

```java
    void addButtons(JPanel panelButtons) {  1 usage
        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 {  2 usages

        @Override
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
            if (button.charAt(0) >= '0' && button.charAt(0) <= '9') {
                tfDisplay.setText(tfDisplay.getText() + button);
            } else if (button.equals("DEL")) {
                String text = tfDisplay.getText();
                if (text.length() == 0)
                    return;
                tfDisplay.setText(text.substring(0, text.length() - 1));
            } else {
                tfDisplay.setText("");
            }
        }
    }

    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

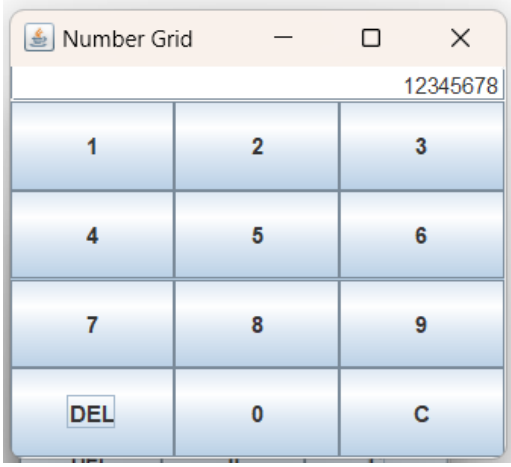


Figure 2.4: Demo DEL button

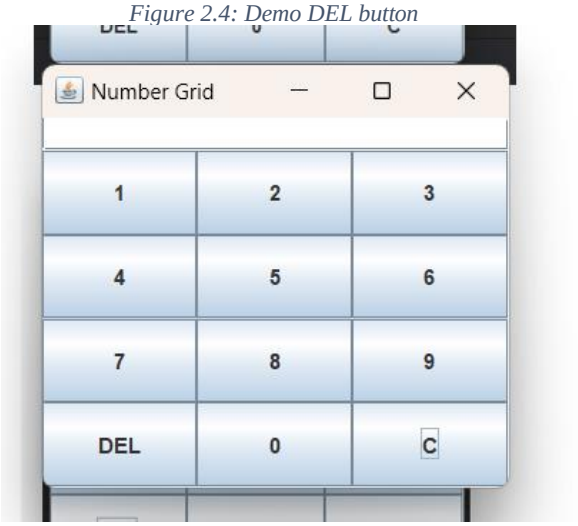


Figure 2.5: Demo C button

3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

```java
public class StoreScreen extends JFrame {  6 usages
    private Store store;  4 usages
    private JPanel center;  3 usages
     // Lê Quang Khải 20225638
    public StoreScreen(Store store) {  1 usage
        this.store = store;

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

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

        setVisible(true);
        setTitle("Store");
        setBounds( x: 100,  y: 0,  width: 1024,  height: 768);
    }
     // Lê Quang Khải 20225638
    JPanel createNorth() {  1 usage
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }
     // Lê Quang Khải 20225638
    JMenuBar createMenuBar() {  1 usage
        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

```java
public class StoreScreen extends JFrame {  6 usages
    JMenuBar createMenuBar() {  1 usage
        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;
    }
     // Lê Quang Khải 20225638
    JPanel createHeader() {  1 usage
        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);
         // Lê Quang Khải 20225638
        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

```java
public class StoreScreen extends JFrame {  6 usages
    JPanel createSearchBar() {  1 usage
        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  ");
```

Figure 3.3: Class StoreScreen 3

```java
public class StoreScreen extends JFrame {  6 usages
    JPanel createSearchBar() {  1 usage
        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(false);
// Lê Quang Khải 20225638
        ActionListener actionListener = new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (btnByTitle.isSelected() || btnByCategory.isSelected()) {
                    resetTextFields();
                    textField.setVisible(true);
                    panelCostFromTo.setVisible(false);
                } else {
                    resetTextFields();
                    textField.setVisible(false);
                    panelCostFromTo.setVisible(true);
                }
            }
            void resetTextFields() {  2 usages
                textField.setText("");
                tfFrom.setText("");
                tfTo.setText("");
            }
        };
```

Figure 3.4: Class StoreScreen 4

```java
public class StoreScreen extends JFrame {  6 usages
    JPanel createSearchBar() {  1 usage

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

        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("")) {
                        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()));
                    }
```

Figure 3.5: Class StoreScreen 5

```java
public class StoreScreen extends JFrame {  6 usages
    JPanel createSearchBar() {  1 usage
        DocumentListener documentListener = new DocumentListener() {
            public void changedUpdate(DocumentEvent e) {
                } else {
                    if (tfFrom.getText().equals("") && tfTo.getText().equals("")) {
                        loadItemsToStore();
                        return;
                    }

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

    }
     // Lê Quang Khải 20225638
    JPanel createCenter(List<Media> itemList) {  2 usages
        JPanel center = new JPanel();
```

Figure 3.6: Class StoreScreen 6

```java
    JPanel createSearchBar() {  1 usage
    }
     // Lê Quang Khải 20225638
    JPanel createCenter(List<Media> itemList) {  2 usages
        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));

            center.add(Box.createRigidArea(new Dimension( width: 10,  height: 200)));
            center.add(lblStoreEmpty);
            return center;
        }

        center.setLayout(new GridLayout( rows: 0,  cols: 3,  hgap: 2,  vgap: 2));

        for (int i = 0; i < itemsToShow; i++) {
            MediaStore cell = new MediaStore(itemList.get(i),  storeScreen: this);
            center.add(cell);
        }

        return center;
    }

    public void loadItemsToStore(List<Media> itemList) {  3 usages
        remove(center);
        add(center = createCenter(itemList), BorderLayout.CENTER);
        repaint();
        revalidate();
```

```java
    }

    public void loadItemsToStore(List<Media> itemList) {  3 usages
        remove(center);
        add(center = createCenter(itemList), BorderLayout.CENTER);
        repaint();
        revalidate();
    }

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

    private class MenuListener implements ActionListener {  5 usages

        @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 3.7: Class StoreScreen 7 và 8

3.2 Create class MediaStore

```java
public class MediaStore extends JPanel {  2 usages
    private Media media;  5 usages
    private StoreScreen storeScreen;  1 usage
    public MediaStore(Media media, StoreScreen storeScreen) {  1 usage
        this.media = media;
        this.storeScreen = storeScreen;
        this.setLayout(new BoxLayout( target: 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("%.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);
        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }

    private class ButtonListener implements ActionListener {  3 usages
        @Override
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
```

Figure 3.7: Class MediaStore 1

```java
    }

    private class ButtonListener implements ActionListener {  3 usages
        @Override
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
            switch (button) {
                //Lê Chí Dũng 20210216
            case "Add to cart":
                try {
                    Aims.getCart().addMedia(media);
                    JOptionPane.showMessageDialog( parentComponent: null,
                            String.format("Added %s to cart.\nCurrent number of items in cart: %d", media.toString(),
                                    Aims.getCart().getItemsOrdered().size()));
                } catch (LimitExceededException | DuplicatedItemException e1) {
                    JOptionPane.showMessageDialog( parentComponent: null, e1.getMessage(),  title: "Error", JOptionPane.ERROR_MESSAGE);
                }
                break;
            case "Play":
                try {
                    ((Playable) media).play();
                } catch (PlayerException e2) {
                    JOptionPane.showMessageDialog( parentComponent: null, e2.getMessage(),  title: "Error", JOptionPane.ERROR_MESSAGE);
                }
                break;
            case "Details":
                new DetailScreen(media);
                break;
            }
        }
    }
}
```

Figure 3.8: Class MediaStore 2

3.3 Demo

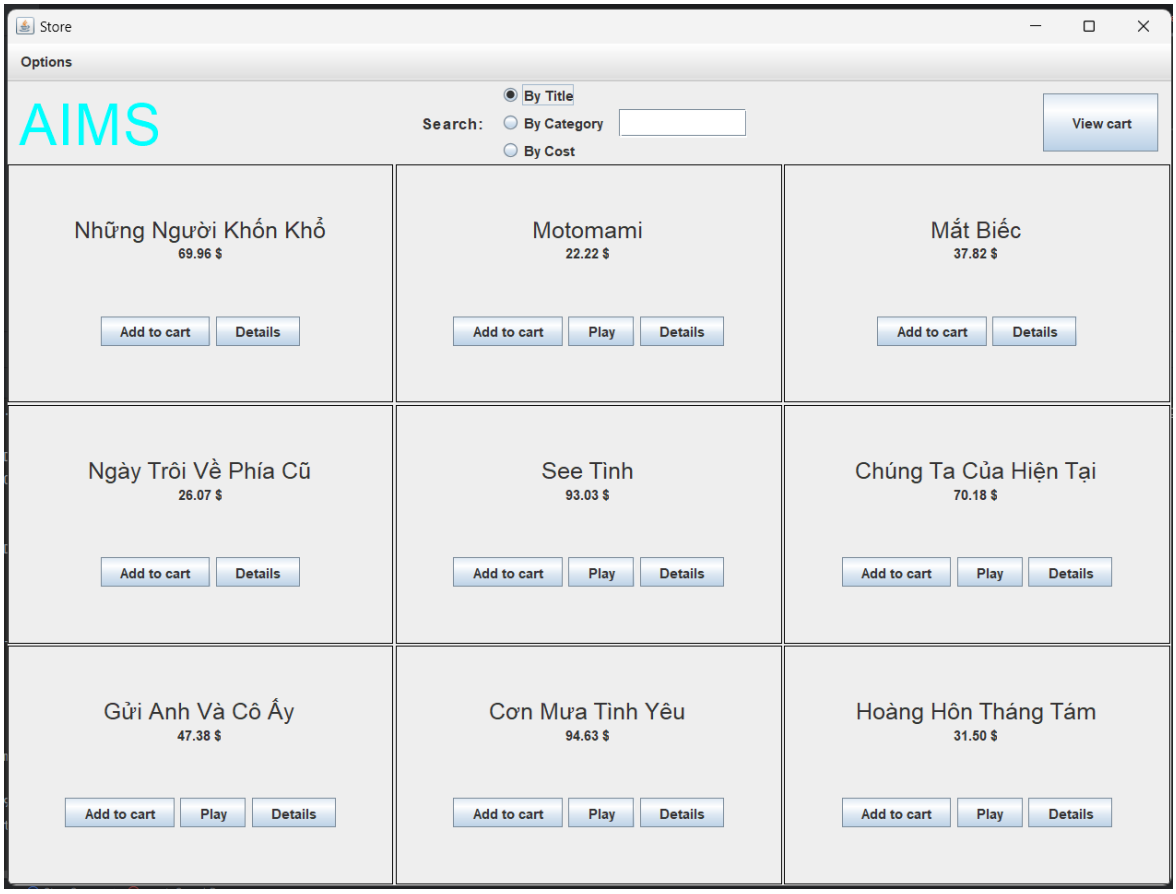


Figure 3.10: StoreScreen

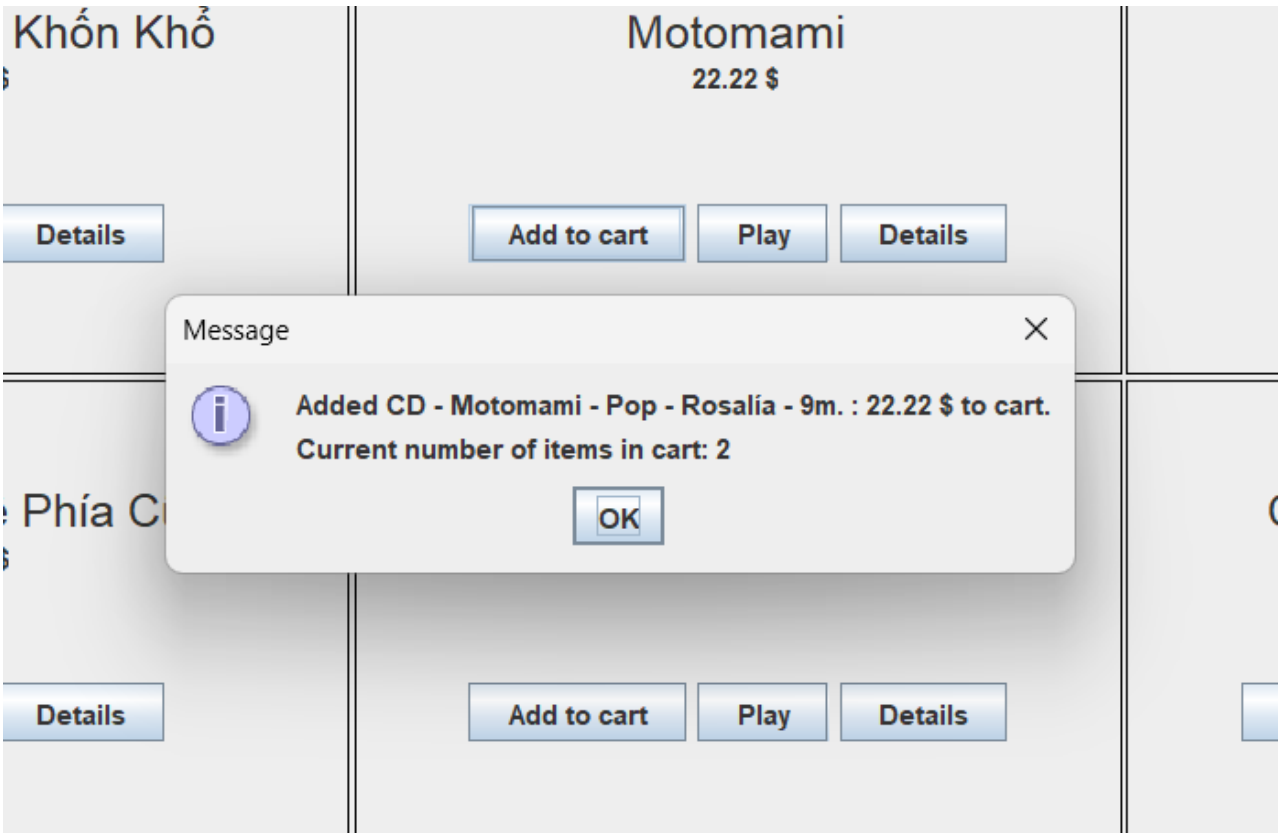


Figure 3.11 Demo Add to cart button

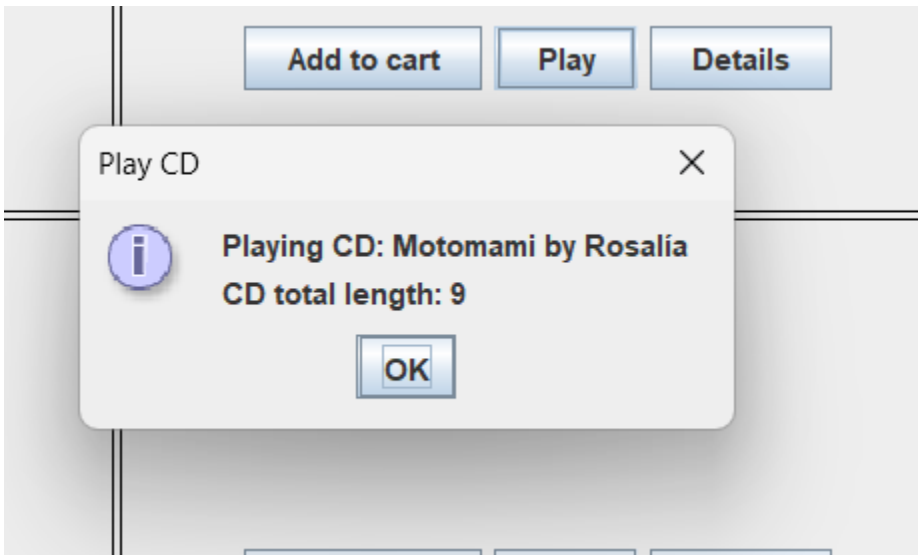


Figure 3.12 Demo Play button

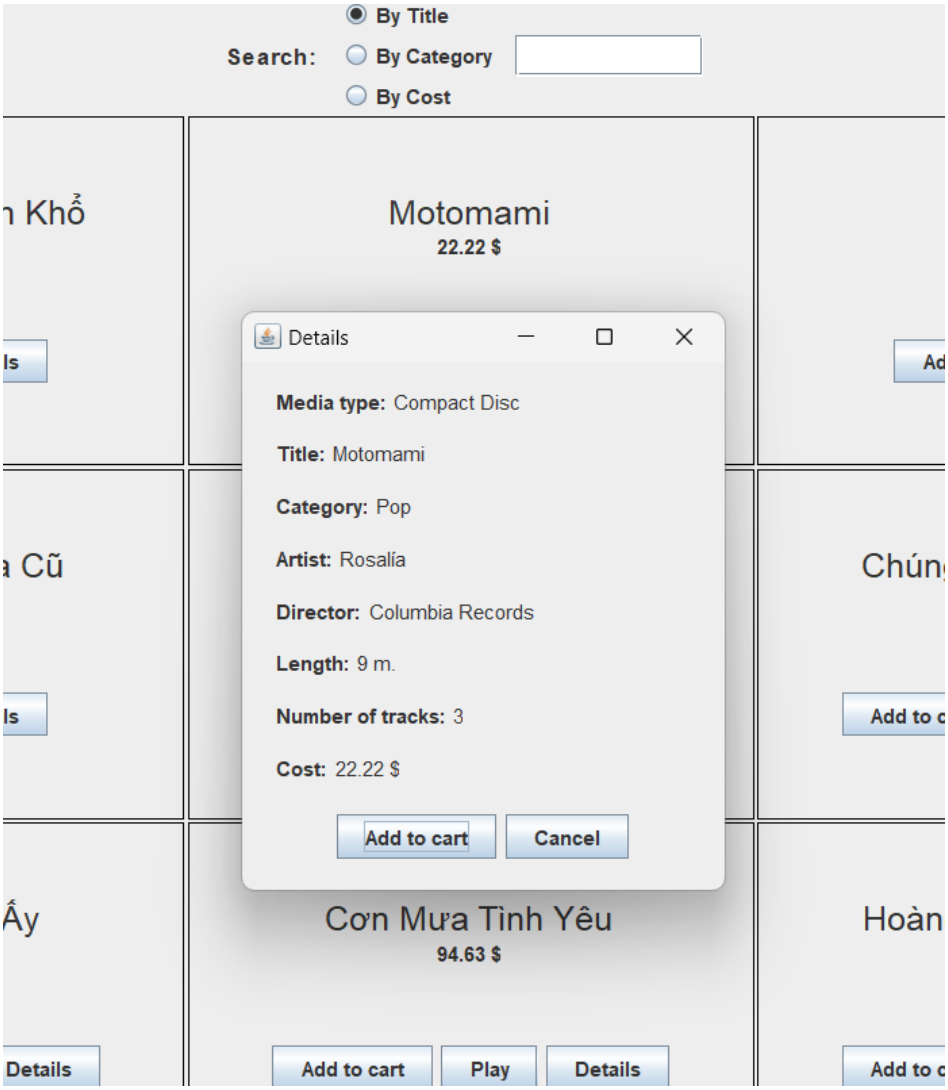


Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

```java
package hust.soict.hedspi.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 {

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

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

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

}
```

Figure 4.1: Class Painter

4.2 Create Painter.fxml

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

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

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="h
    <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" maxWidth="1.7976931348623157E308" text="Tools">
                    <content>
                        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="80.0" prefWidth="80.0">
                            <children>
                                <RadioButton fx:id="penRadioButton" layoutX="11.0" layoutY="14.0" mnemonicParsing="false" selected="true" text="Pen">
                                    <font>
                                        <Font size="13.0" />
                                    </font>
                                    <toggleGroup>
                                        <ToggleGroup fx:id="radioGroup" />
                                    </toggleGroup>
                                </RadioButton>
                                <RadioButton fx:id="eraserRadioButton" layoutX="11.0" layoutY="40.0" mnemonicParsing="false" text="Eraser" toggleGroup="$radioGroup">
                                    <font>
                                        <Font size="13.0" />
```

Figure 4.2: Painter.fxml 1

```
14          <left>
15              <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
19                  <children>
20                      <TitledPane animated="false" maxWidth="1.7976931348623157E308" text="Tools">
30                                  </toggleGroup>
31                              </RadioButton>
32                              <RadioButton fx:id="eraserRadioButton" layoutX="11.0" layoutY="40.0" mnemonicParsing="false" text="Eraser" toggleGroup="$radioGroup
33                                  <font>
34                                      <Font size="13.0" />
35                                  </font>
36                              </RadioButton>
37                          </children>
38                      </AnchorPane>
39                  </content>
40                  <font>
41                      <Font size="13.0" />
42                  </font>
43              </TitledPane>
44              <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear">
45                  <font>
46                      <Font size="13.0" />
47                  </font>
48              </Button>
49          </children>
50      </VBox>
51  </left>
52  <center>
53      <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
54  </center>
55 </BorderPane>
56
```

Figure 4.3: Painter.fxml 2

4.3 Create class PainterController

```
package hust.soict.hedspi.javafx;
D:\Workspace\Forlearning\Java\IT3103.744527.2024.1.20225638.LeQuangKhai\GUIProject\src\hust\soict\hedspi\javafx\Painter.fxml
import ...
// Lê Quang Khải 20225638
public class PainterController {  1 usage

    @FXML
    private RadioButton penRadioButton;

    @FXML
    private Pane drawingAreaPane;

    @FXML
    private RadioButton eraserRadioButton;

    @FXML
    private ToggleGroup radioGroup;

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Color color = penRadioButton.isSelected() ? Color.BLACK : Color.WHITE;
        Circle newCircle = new Circle(event.getX(), event.getY(),  v2: 4, color);
        drawingAreaPane.getChildren().add(newCircle);
    }

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

}
```

Figure 4.4: PainterController

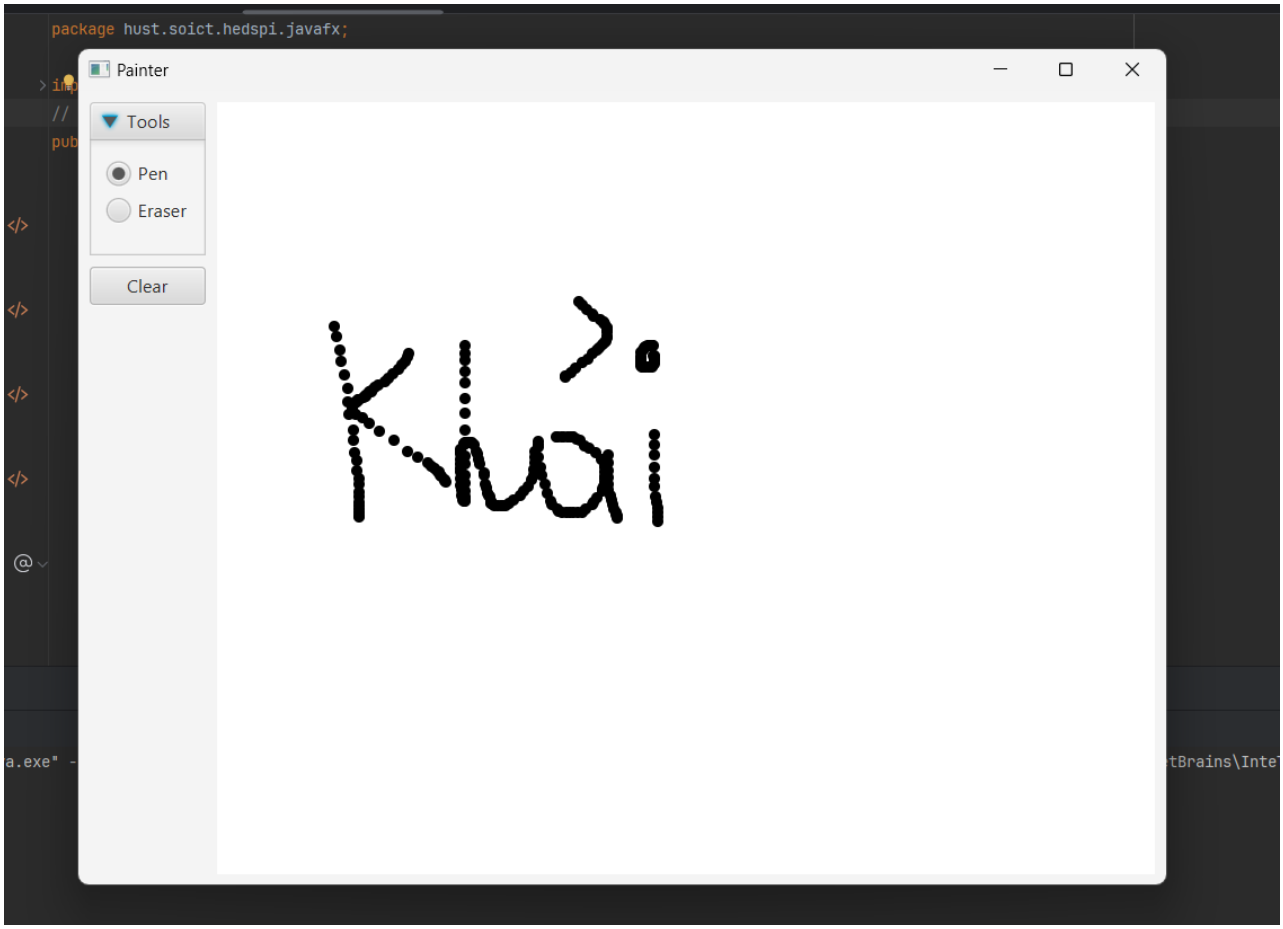


Figure 4.5: Use Pen

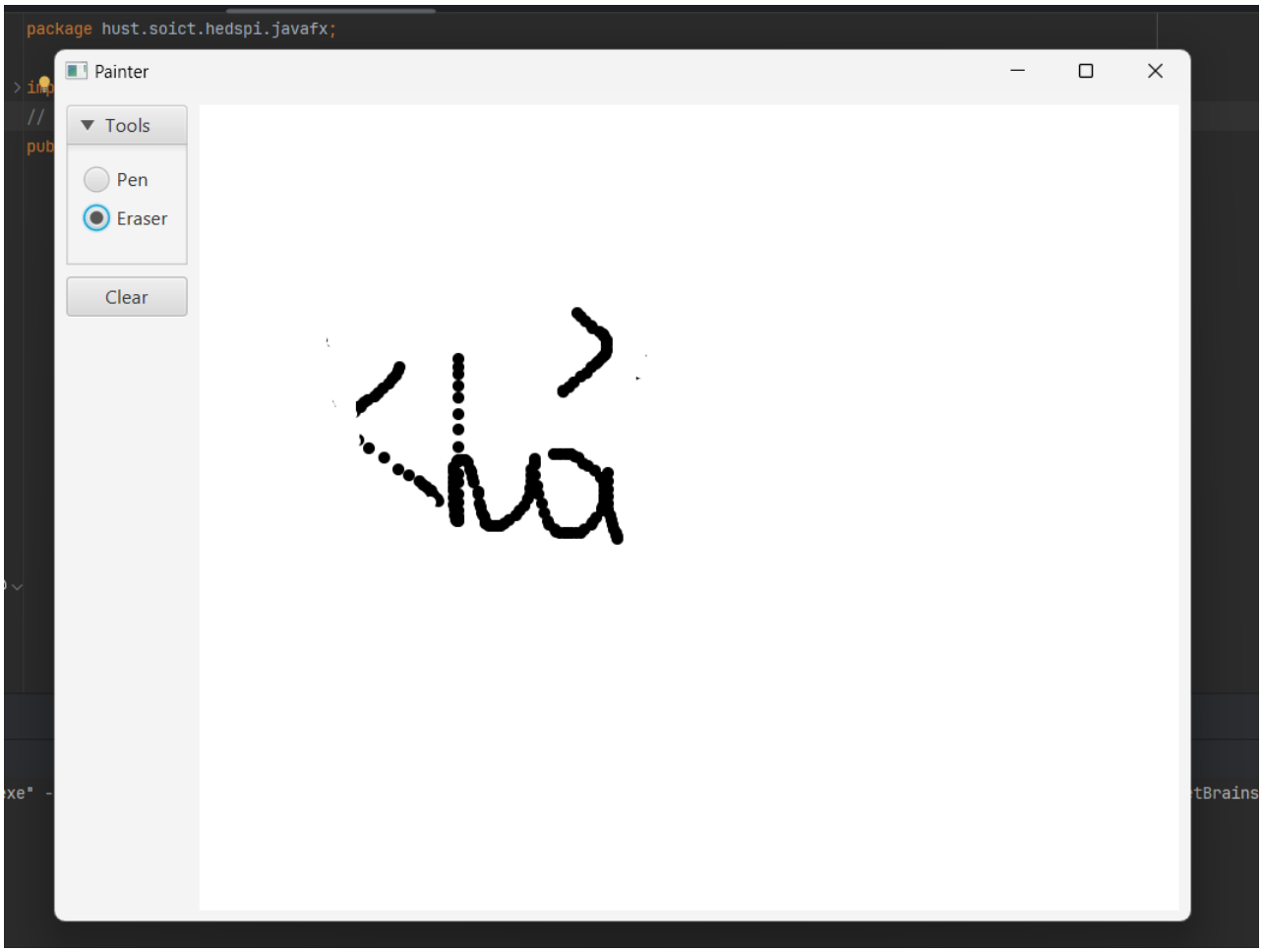


Figure 4.6: Use Eraser

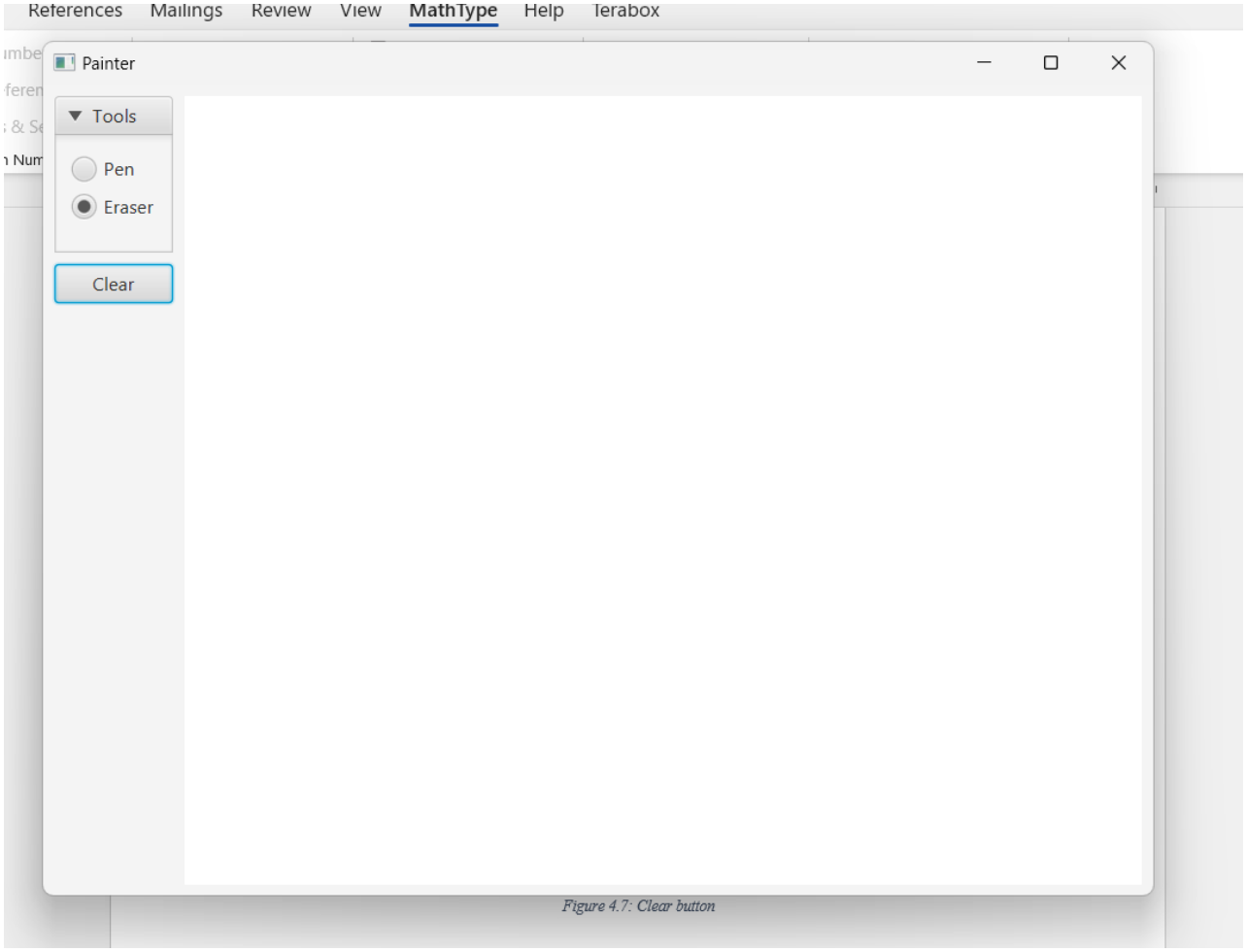


Figure 4.7: Clear button

5 View Cart Screen

5.1 Create cart.fxml

```
<!--  Lê Quang Khải 20225638 -->
<BorderPane prefHeight="768.0" prefWidth="1024.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/jav
  <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>
                  <MenuItem mnemonicParsing="false" 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>
      </children>
    </VBox>
  </top>
  <center>
    <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
```

Figure 5.1: Cart.fxml 1

```
43        <center>
44           <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
46              <Insets left="10.0" />
47           </padding>
48           <children>
49              <HBox alignment="CENTER_LEFT" prefWidth="200.0" spacing="10.0">
50                 <opaqueInsets>
51                    <Insets />
52                 </opaqueInsets>
53                 <padding>
54                    <Insets bottom="10.0" top="10.0" />
55                 </padding>
56                 <children>
57                    <Label text="Filter:" />
58                    <TextField>
59                       <font>
60                          <Font size="13.0" />
61                       </font>
62                    </TextField>
63                    <RadioButton mnemonicParsing="false" selected="true" text="By ID">
64                       <toggleGroup>
65                          <ToggleGroup fx:id="filterCategory" />
66                       </toggleGroup>
67                    </RadioButton>
68                    <RadioButton mnemonicParsing="false" text="By Title" toggleGroup="$filterCategory" />
69                 </children>
70              </HBox>
71              <TableView>
72                 <columns>
73                    <TableColumn prefWidth="75.0" text="Title" />
74                    <TableColumn prefWidth="75.0" text="Category" />
75                    <TableColumn prefWidth="75.0" text="Cost" />
76                 </columns>
```

Figure 5.2: Cart.fxml 2

```
48              <children>
71                 <TableView>
78                    <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
79                 </columnResizePolicy>
80                 <opaqueInsets>
81                    <Insets />
82                 </opaqueInsets>
83                 <VBox.margin>
84                    <Insets top="10.0" />
85                 </VBox.margin>
86              </TableView>
87              <HBox prefHeight="100.0" prefWidth="200.0" spacing="10.0">
88                 <children>
89                    <Button mnemonicParsing="false" text="Play" />
90                    <Button mnemonicParsing="false" text="Remove" />
91                 </children>
92                 <padding>
93                    <Insets top="20.0" />
94                 </padding>
95              </HBox>
96           </children>
97        </VBox>
98     </center>
99     <right>
100       <VBox alignment="TOP_CENTER" prefHeight="200.0" BorderPane.alignment="CENTER">
101          <padding>
102             <Insets left="20.0" right="20.0" top="60.0" />
103          </padding>
104          <children>
105             <HBox alignment="CENTER" spacing="10.0">
106                <children>
BorderPane › center › VBox › children › TableView › opaqueInsets
```

```
            <insets left="20.0" right="20.0" top="60.0" />
        </padding>
        <children>
            <HBox alignment="CENTER" spacing="10.0">
                <children>
                    <Label 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>
                <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
package hust.soict.hedspi.aims.screen;
import ...
// Lê Quang Khải 20225638
public class CartScreen extends JFrame {  4 usages
    private Cart cart;  1 usage
    public CartScreen(Cart cart) {  1 usage
        super();
        this.cart = cart;
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle("Cart");
        this.setBounds( x: 100,  y: 0,  width: 1024,  height: 768);
        this.setVisible(true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/hust/soict/hedspi/aims/screen/fxml/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

```java
// Lê Quang Khải 20225638
public class CartScreenController {  3 usages
    private Cart cart;  7 usages
    @FXML  4 usages
    private TableColumn<Media, String> colMediaCost;
    @FXML  3 usages
    private TableColumn<Media, String> colMediaTitle;
    @FXML  1 usage
    private TableColumn<Media, String> colMediaCategory;
    @FXML  3 usages
    private Button btnPlay;
    @FXML  2 usages
    private Button btnRemove;
    @FXML  2 usages
    private Button btnDetails;
    @FXML  1 usage
    private RadioButton radioBtnFilterId;
    @FXML  no usages
    private RadioButton radioBtnFilterTitle;
    @FXML  1 usage
    private TextField tfFilter;
    @FXML  13 usages
    private TableView<Media> tblMedia;

    @FXML  no usages
    private ToggleGroup filterCategory;

    @FXML  no usages
    private Button btnPlaceOrder;
    @FXML  1 usage
    private Label lblCost;
    public CartScreenController(Cart cart) {  1 usage
        super();
        this.cart = cart;
    }
    @FXML  no usages
    private void initialize() {
```

Figure 5.5: CartScreenController 1

```java
public class CartScreenController {  3 usages
    private void initialize() {
        colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "cost"));
        tblMedia.setItems(cart.getItemsOrdered());
        btnDetails.setVisible(false);
        // Lê Quang Khải 20225638
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);
        tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
            }
        });
        // Lê Quang Khải 20225638
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
                showFilteredMedia(newValue);
            }

        });
        updateCost();
    }
    // Lê Quang Khải 20225638
    void updateCost() { lblCost.setText(String.format("%.2f $", cart.totalCost())); }
    // Lê Quang Khải 20225638
    void updateButtonBar(Media media) {  1 usage
        btnRemove.setVisible(true);
        if (media instanceof Playable) {
            btnPlay.setVisible(true);
        } else {
            btnPlay.setVisible(false);
```

Figure 5.6: CartScreenController 2

```java
    void updateButtonBar(Media media) {  1 usage
            btnPlay.setVisible(false);
        }
        btnDetails.setVisible(true);
    }
    // Lê Quang Khải 20225638
    void showFilteredMedia(String input) {  1 usage
        if (input == "") {
            tblMedia.setItems(cart.getItemsOrdered());
            return;
        }

        FilteredList<Media> filteredList = new FilteredList<>(cart.getItemsOrdered());
        if (radioBtnFilterId.isSelected())
            filteredList.setPredicate((it) -> it.getId() == Integer.parseInt(input));
        else
            filteredList.setPredicate((it) -> it.isMatch(input));

        tblMedia.setItems(filteredList);
    }

    @FXML  no usages
    void btnSortPressed(ActionEvent event) {
        tblMedia.getSortOrder().clear();

        colMediaCost.setSortType(SortType.DESCENDING);

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

        tblMedia.sort();
```

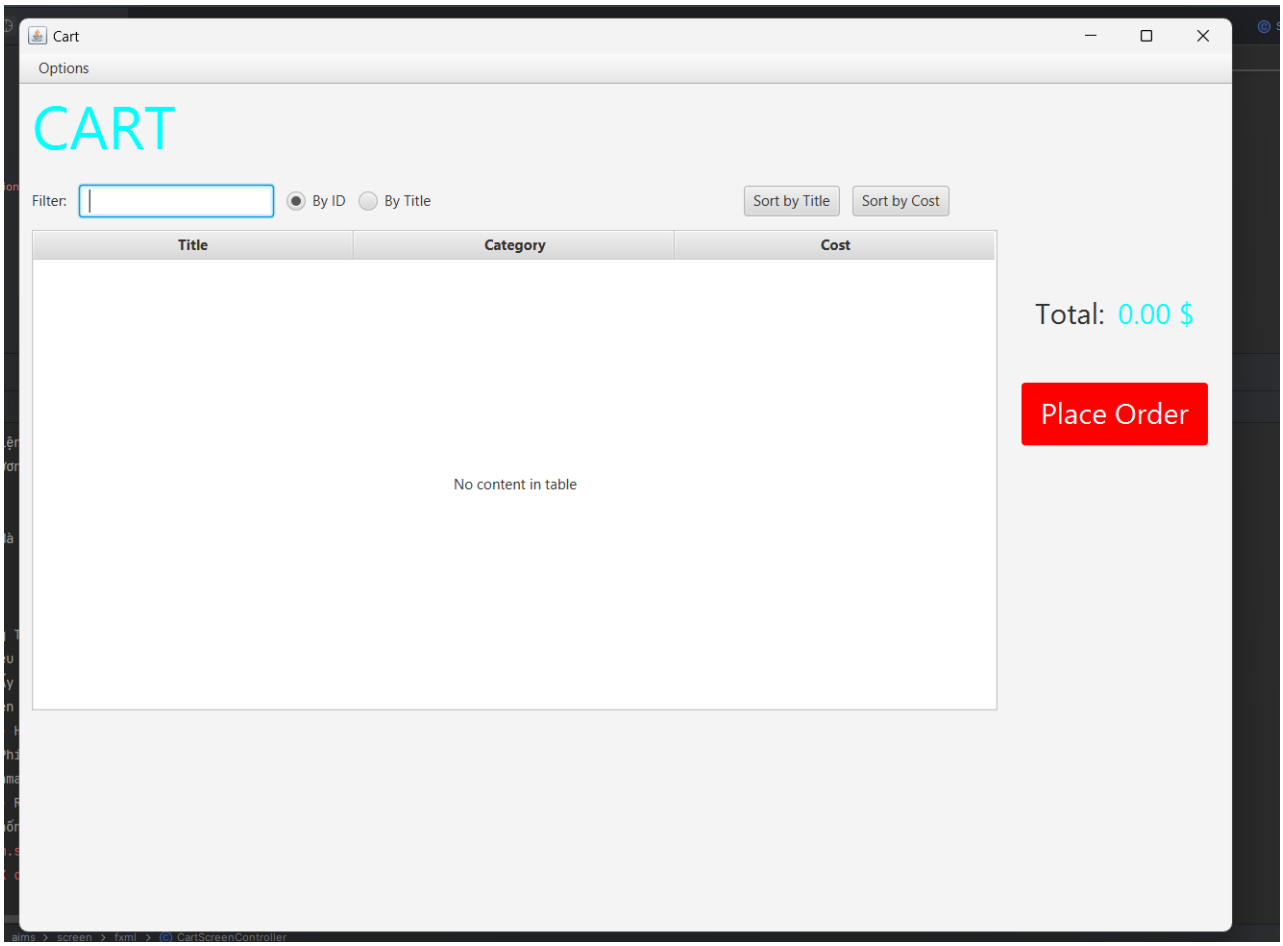
```java
public class CartScreenController { 3 usages
    @FXML no usages
    void btnDetailsPressed(ActionEvent event) { new DetailScreen(tblMedia.getSelectionModel().getSelectedItem()); }
    // Lê Quang Khải 20225638
    @FXML no usages
    void btnPlayPressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        try {
            ((Playable) media).play();
        } catch (PlayerException e) {
            JOptionPane.showMessageDialog( parentComponent: null, e.getMessage(), title: "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
    // Lê Quang Khải 20225638
    @FXML no usages
    void btnRemovePressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        cart.removeMedia(media);
        updateCost();
    }
    // Lê Quang Khải 20225638
    @FXML no usages
    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();
        }
    }

    @FXML no usages
    void menuAddBook(ActionEvent event) { new AddBookToStoreScreen(); }

    @FXML no usages
    void menuAddCd(ActionEvent event) { new AddCompactDiscToStoreScreen(); }

CartScreenController
```

5.4 Demo


Cart
Options
CART
Filter:
By ID
By Title
Sort by Title
Sort by Cost
Title
Category
Cost
No content in table
Total: 0.00 $
Place Order

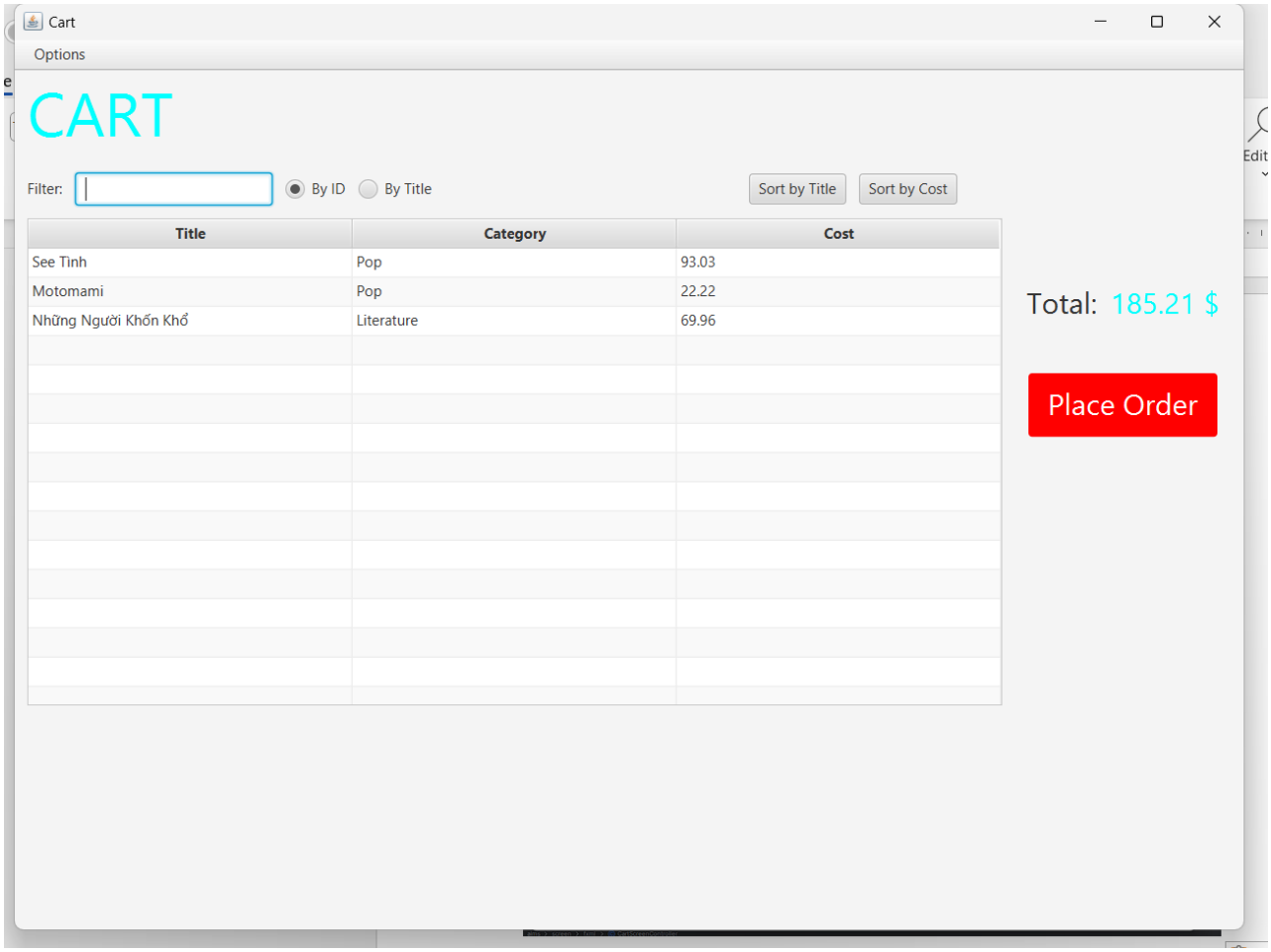


Figure 5.7: Demo CartScreen

6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController

```java
    }
    @FXML  no usages
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "title"));
        colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, String>( s: "cost"));
        tblMedia.setItems(cart.getItemsOrdered());
        btnDetails.setVisible(false);
        // Lê Quang Khải 20225638
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);
        tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
            }
        });
        // Lê Quang Khải 20225638
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
                showFilteredMedia(newValue);
            }

        });
        updateCost();
    }
    // Lê Quang Khải 20225638
```

Figure 6.1: CartScreenController 1

```
        updateCost();
    }
    // Lê Quang Khải 20225638
    void updateCost() { lblCost.setText(String.format("%.2f $", cart.totalCost())); }
    // Lê Quang Khải 20225638
    void updateButtonBar(Media media) {  1 usage
        btnRemove.setVisible(true);
        if (media instanceof Playable) {
            btnPlay.setVisible(true);
        } else {
            btnPlay.setVisible(false);
        }
        btnDetails.setVisible(true);
    }
```

Figure 6.2: CartScreenController 2

6.2 Demo

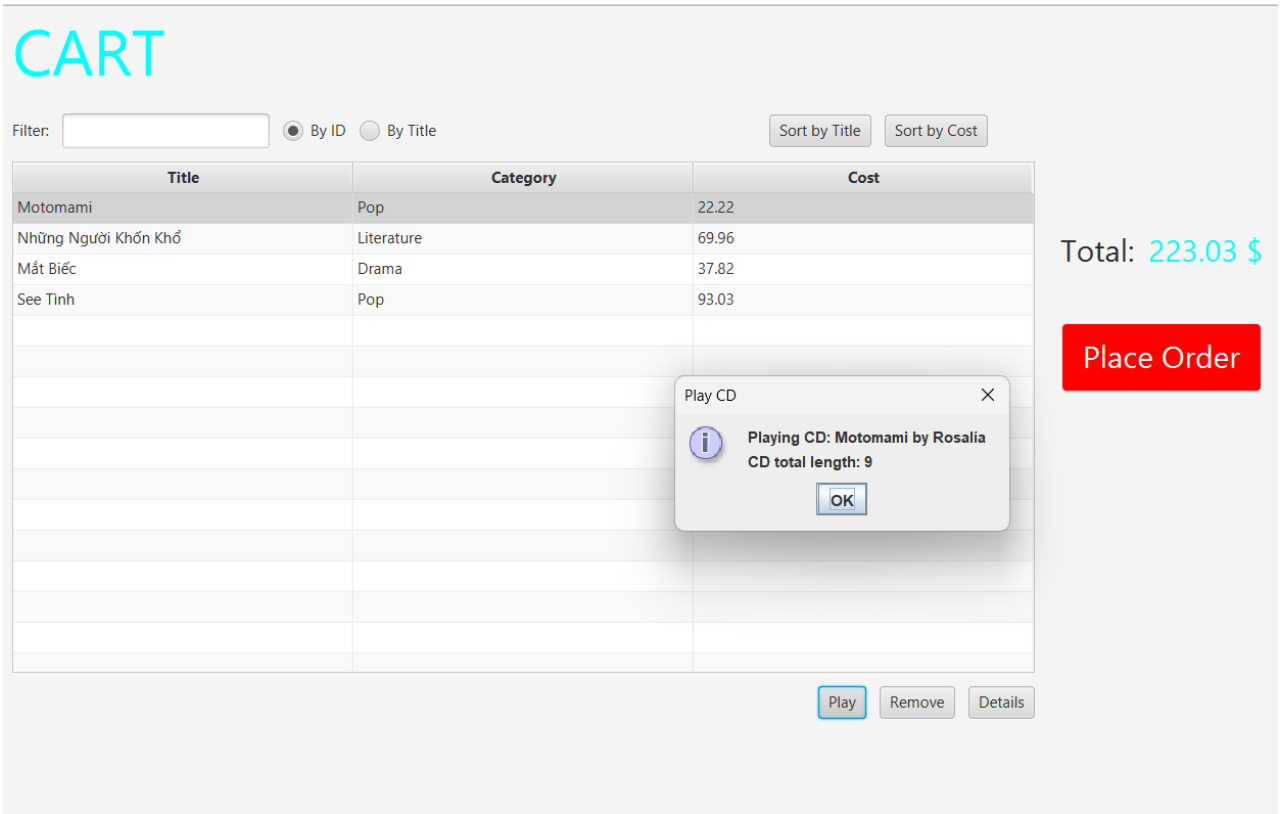


Figure 6.3: Demo media playable

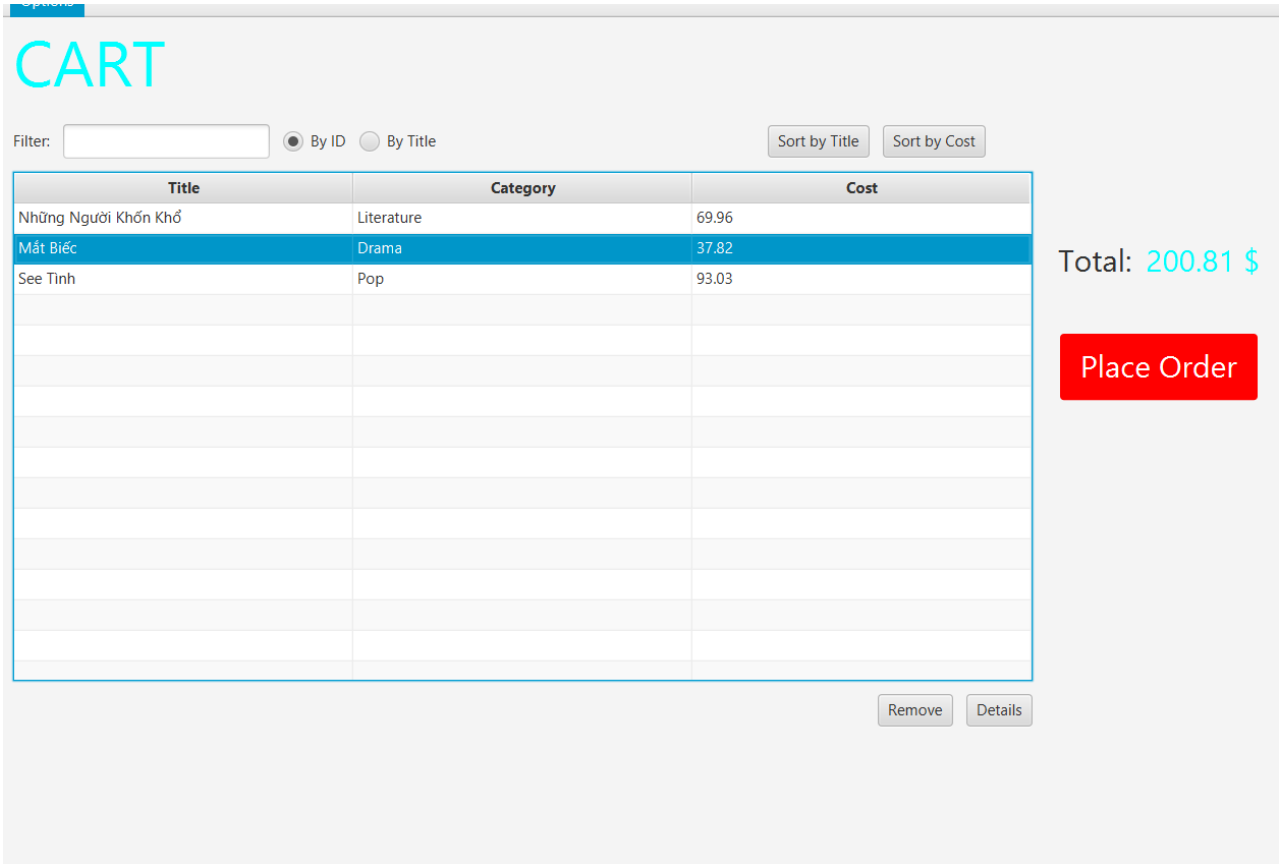


Figure 6.4: Demo media unplayable

7 Deleting a media

7.1 Code

```
}
// Lê Quang Khải 20225638
@FXML  no usages
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    updateCost();
}
```

Figure 7.1: btnRemovePressed Method

7.2 Demo

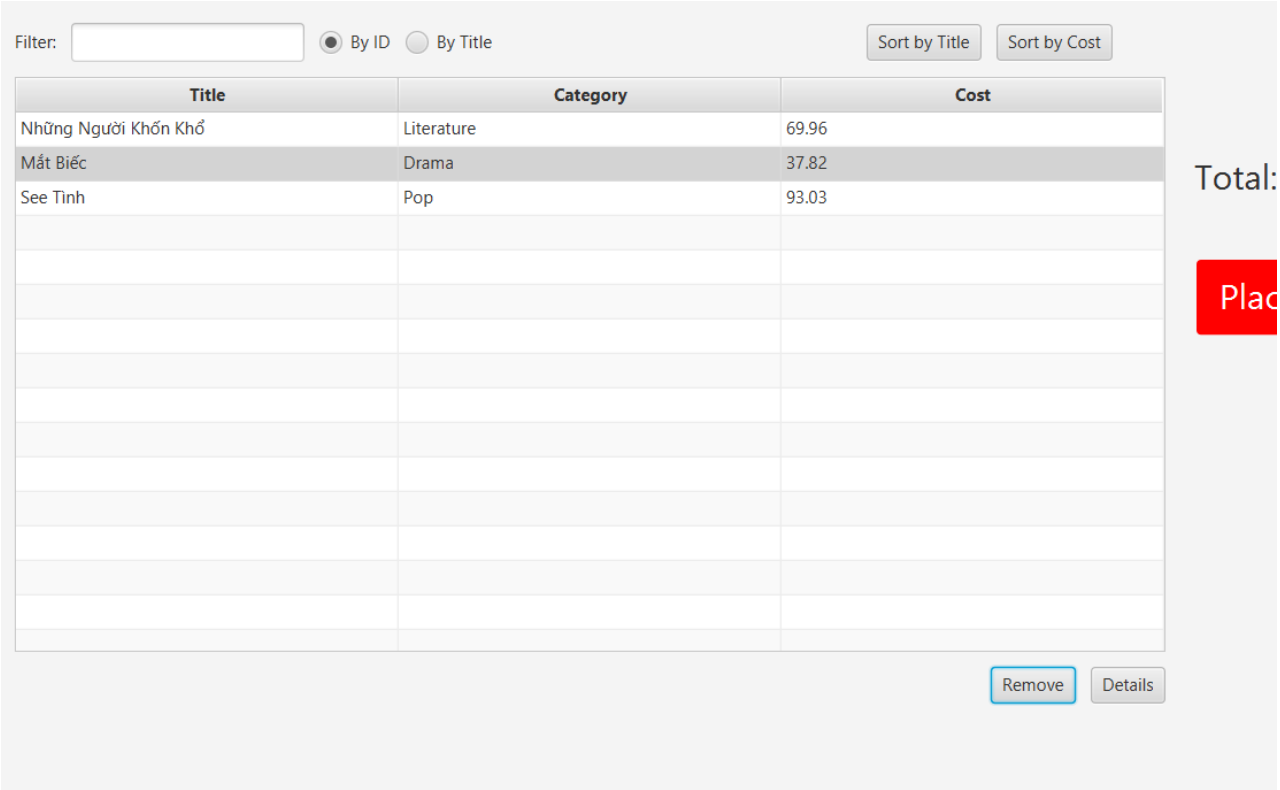


Figure 7.2: button Remove

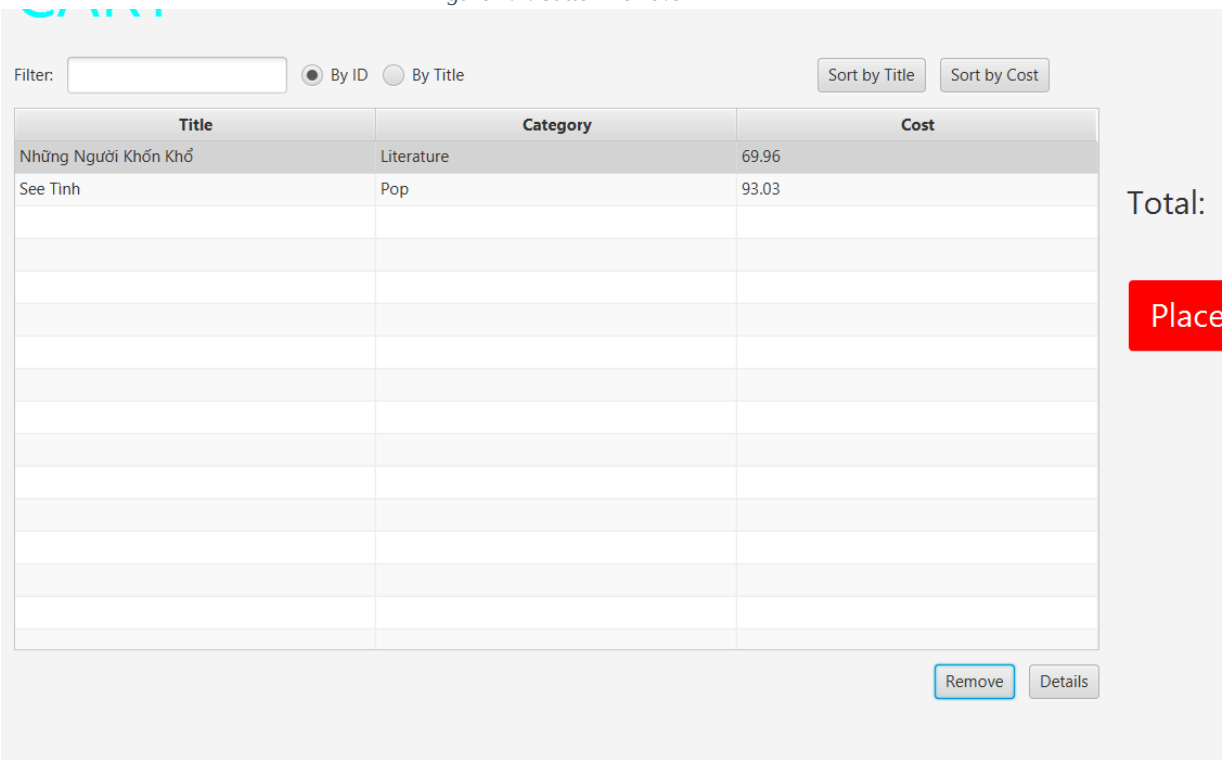


Figure 7.3: button Remove

8 Complete the Aims GUI application

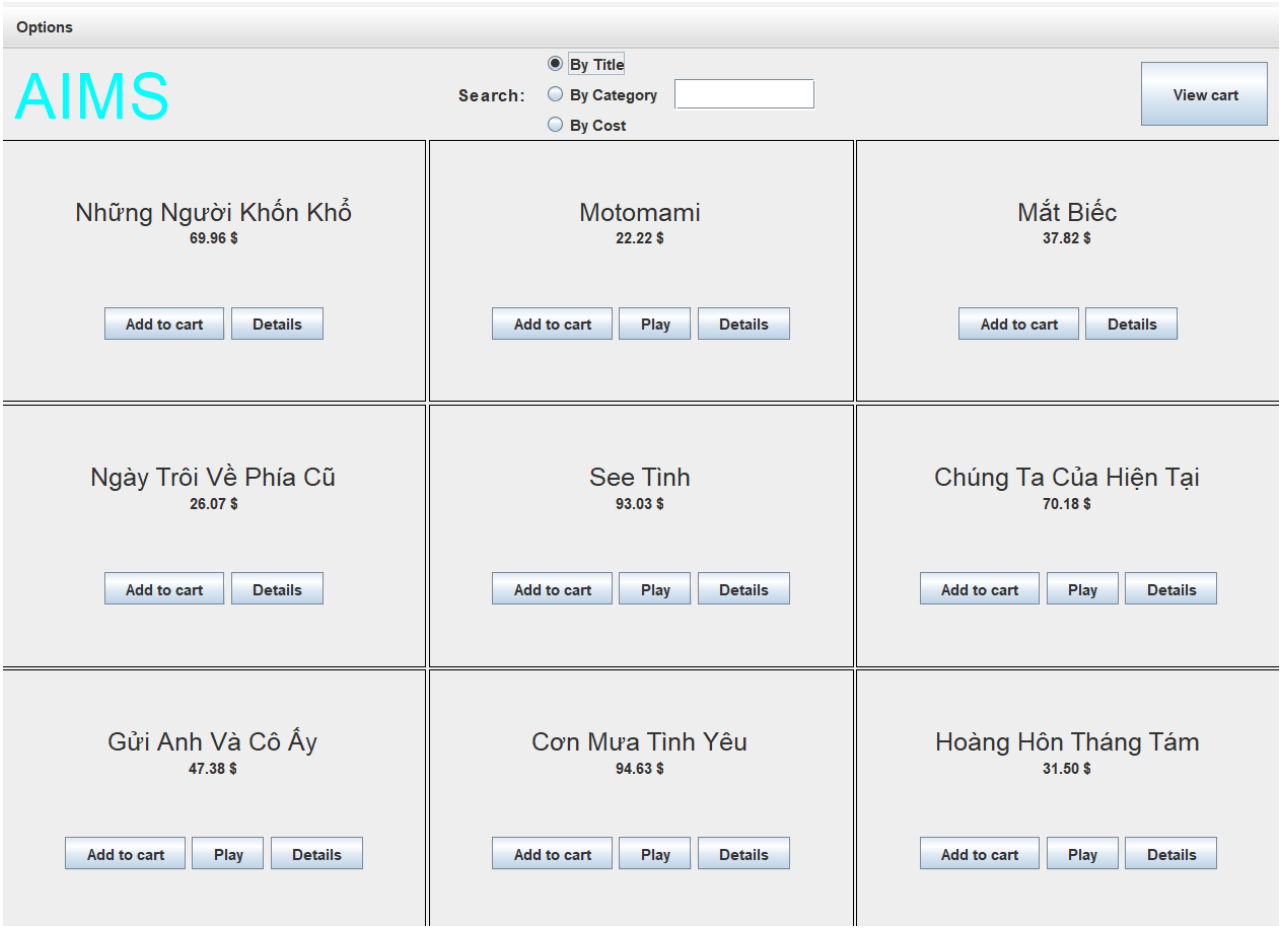


Figure 8.1: Store before add book

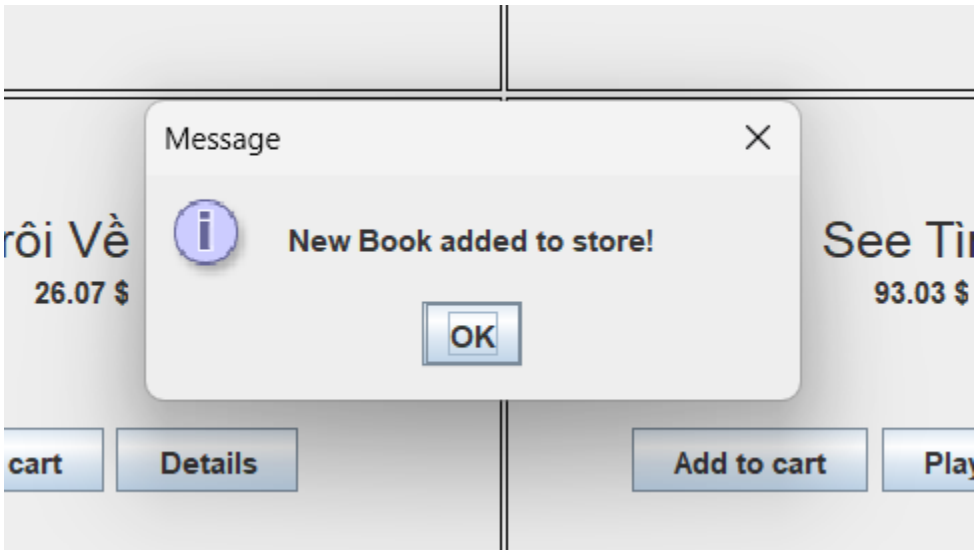


Figure 8.2: Add book

Figure 8.3: Store after add book

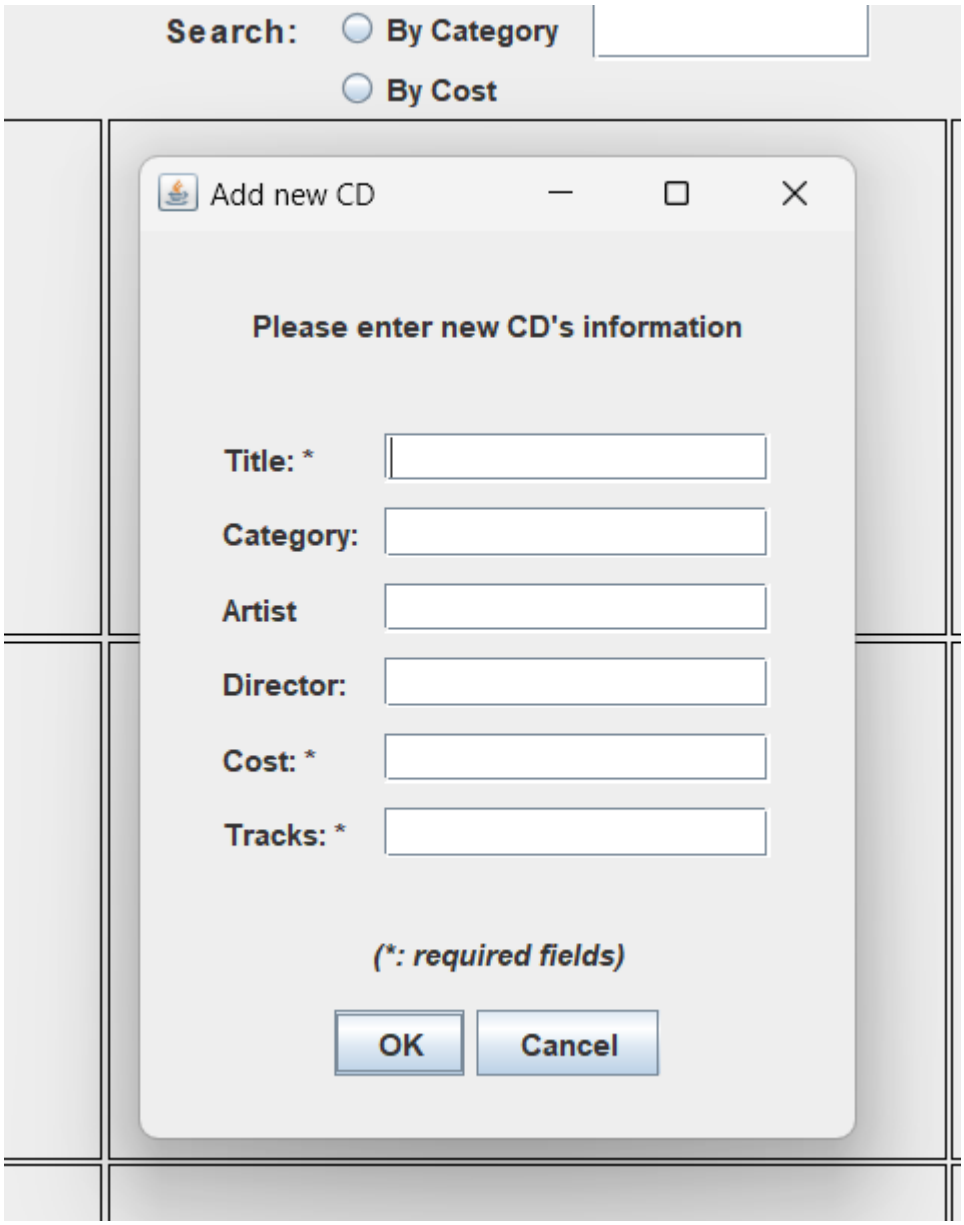


Figure 8.4: Add CD

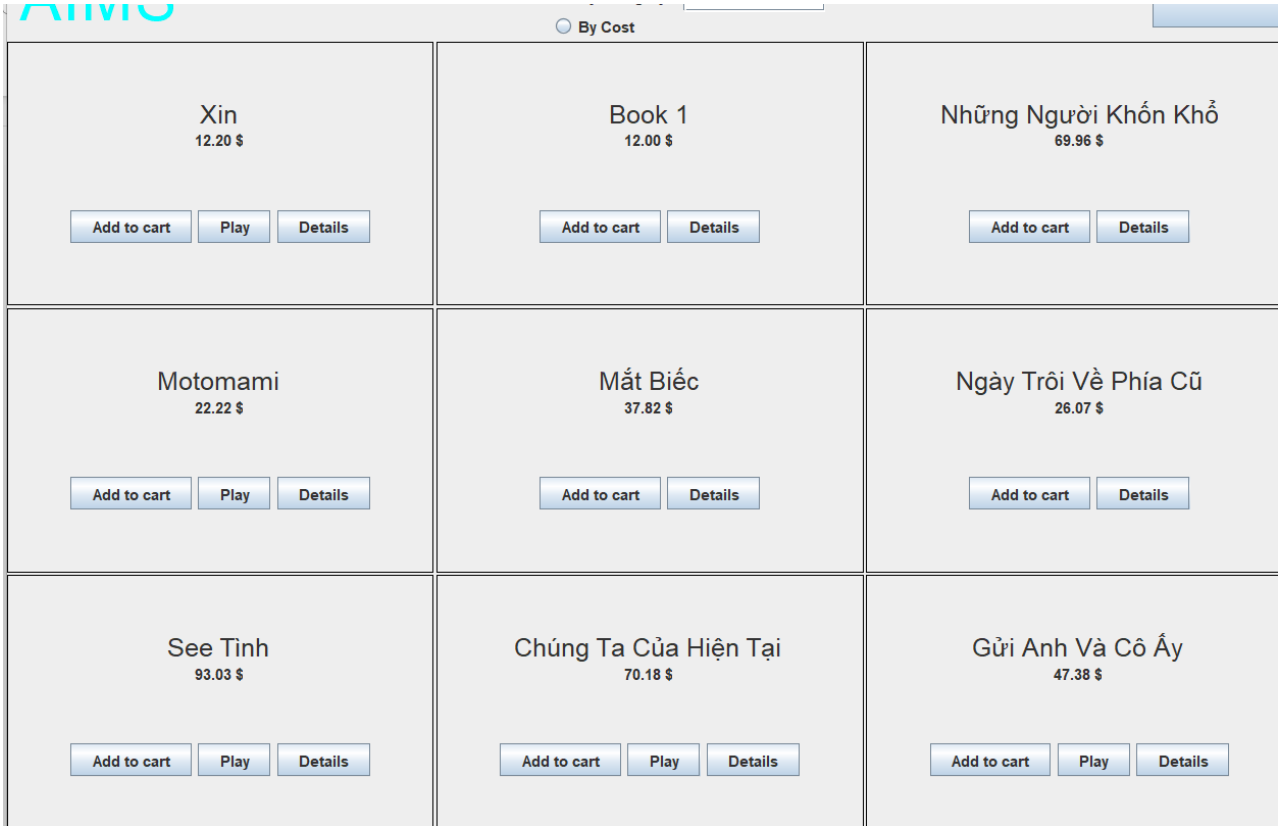


Figure 8.5: Store after add CD

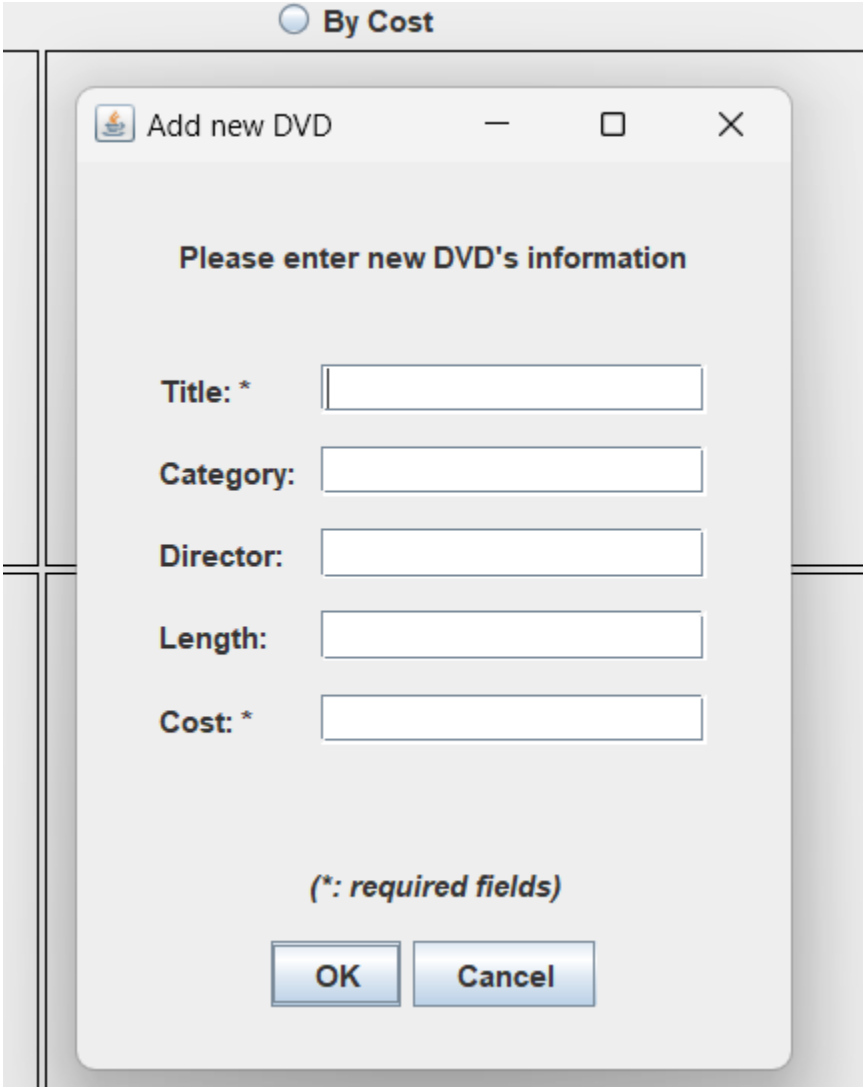


Figure 8.6 Add DVD

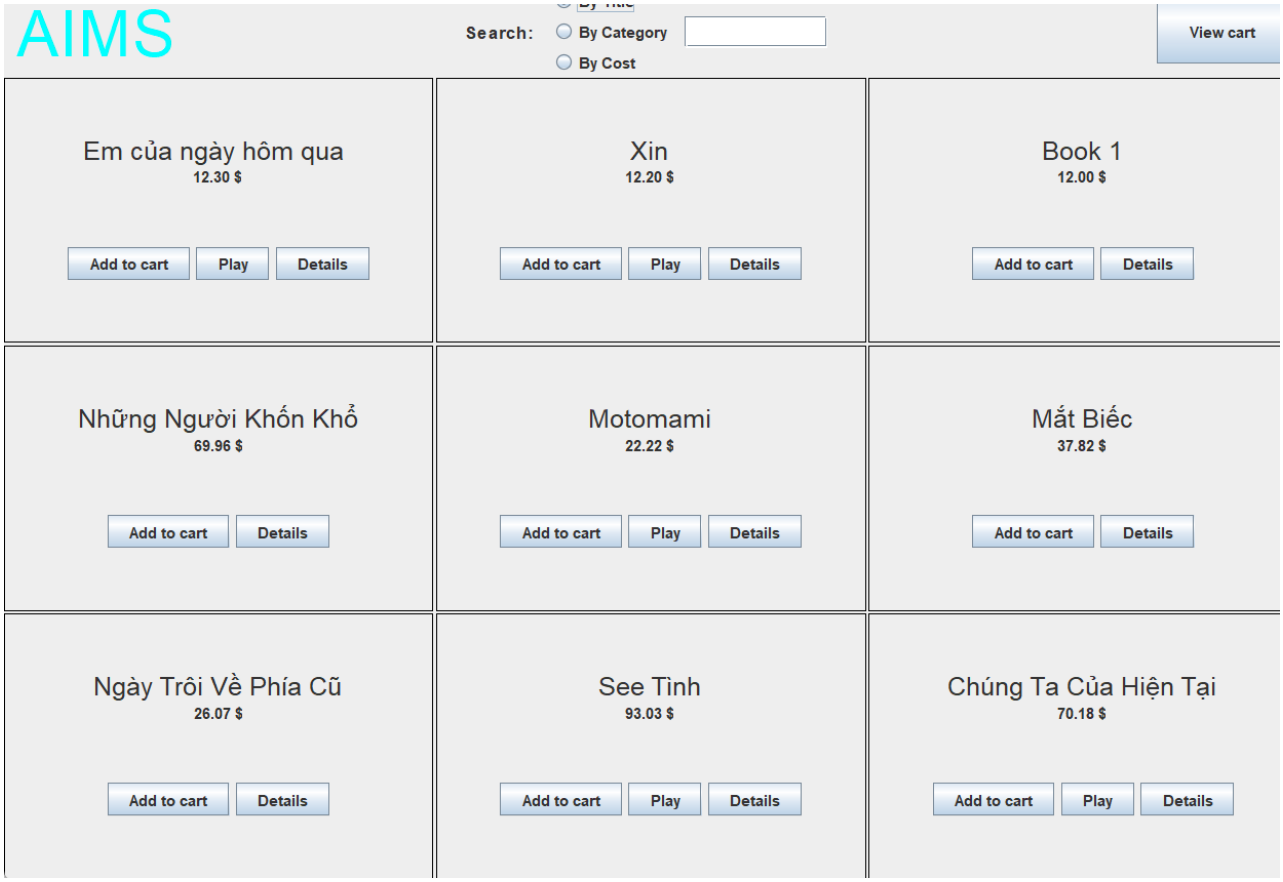


Figure 8.7: Store after add DVD

```java
package hust.soict.hedspi.aims.cart;

import ...

public class Cart {  11 usages  KhaiLe190904 *

    public static final int MAX_NUMBERS_ORDERED = 20;  1 usage
    // Lê Quang Khải 20225638
    private ObservableList<Media> itemsOrdered = FXCollections.observableArrayList();  14 usages

    public void addMedia(Media media) throws LimitExceededException, DuplicatedItemException {  2 usages  new *
        if (itemsOrdered.size() == MAX_NUMBERS_ORDERED) {
            throw new LimitExceededException("ERROR: The number of media has reached its limit.");
        }

        if (itemsOrdered.contains(media)) {
            throw new DuplicatedItemException("ERROR: Item already in cart.");
        }

        itemsOrdered.add(media);
    }

```

Figure 8.8: Cart

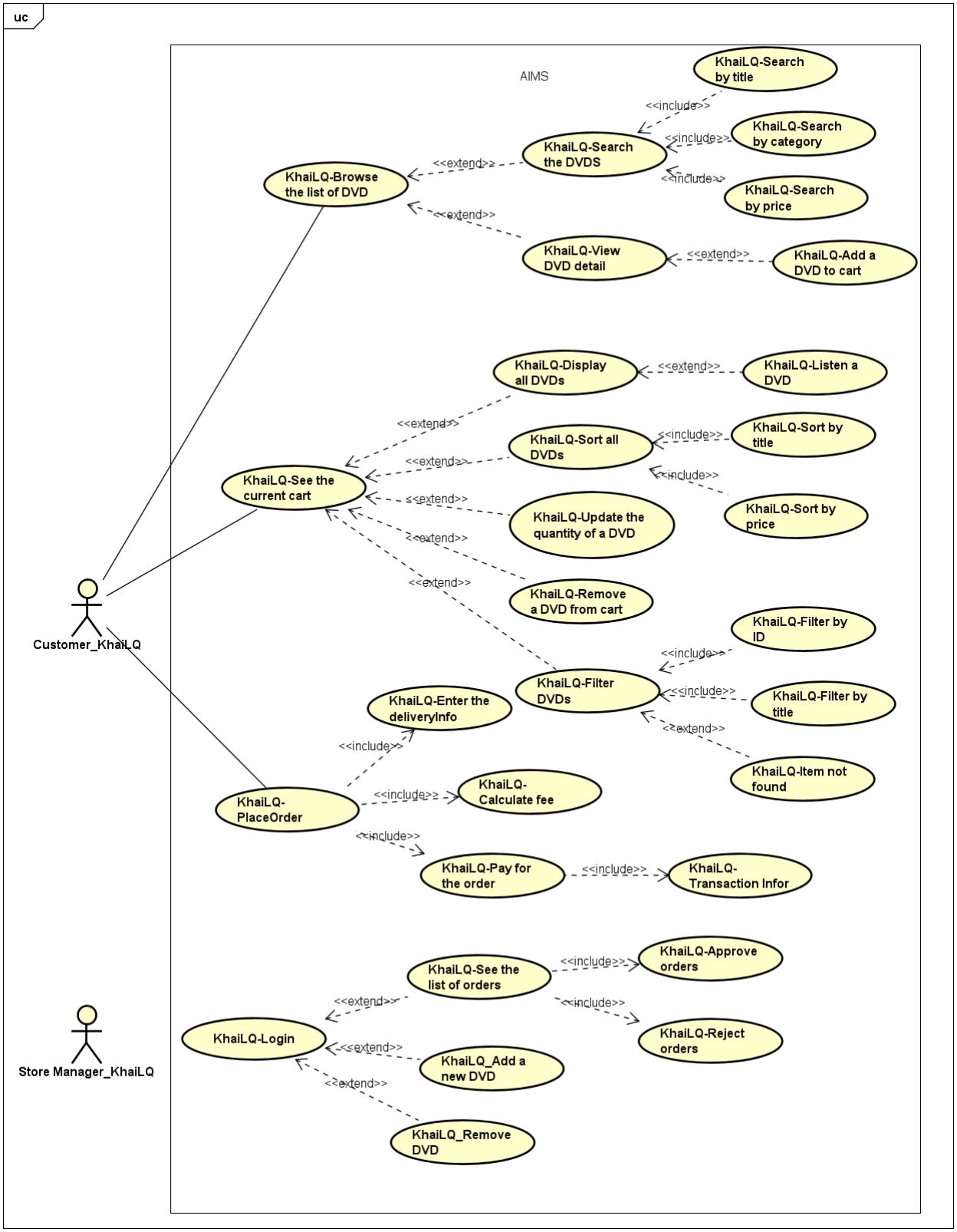
```java
package hust.soict.hedspi.aims.exception;
// Lê Quang Khải 20225638
public class DuplicatedItemException extends Exception { 18 usages
    public DuplicatedItemException() { no usages
    }
    public DuplicatedItemException(String message) { super(message); }
    public DuplicatedItemException(Throwable cause) { super(cause); }
    public DuplicatedItemException(String message, Throwable cause) { super(message, cause); }
    public DuplicatedItemException(String message, Throwable cause, boolean enableSuppression, no usages
            boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
    }
}
```

```java
package hust.soict.hedspi.aims.exception;
// Lê Quang Khải 20225638
public class PlayerException extends Exception { 16 usages

    public PlayerException() { super(); }
    public PlayerException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) { no usages
        super(message, cause, enableSuppression, writableStackTrace);
    }
    public PlayerException(String message, Throwable cause) { super(message, cause); }
    public PlayerException(String message) { super(message); }
    public PlayerException(Throwable cause) { super(cause); }
}
```

Figure 8.9: Exception

9 Use case Diagram


uc
AIMS
KhaiLQ-Search by title
KhaiLQ-Search by category
KhaiLQ-Search the DVDS
KhaiLQ-Search by price
KhaiLQ-Browse the list of DVD
KhaiLQ-View DVD detail
KhaiLQ-Add a DVD to cart
KhaiLQ-Display all DVDs
KhaiLQ-Listen a DVD
KhaiLQ-Sort all DVDs
KhaiLQ-Sort by title
KhaiLQ-Sort by price
KhaiLQ-See the current cart
KhaiLQ-Update the quantity of a DVD
KhaiLQ-Remove a DVD from cart
KhaiLQ-Filter by ID
KhaiLQ-Filter DVDs
KhaiLQ-Filter by title
KhaiLQ-Item not found
KhaiLQ-Enter the deliveryInfo
KhaiLQ-PlaceOrder
KhaiLQ-Calculate fee
KhaiLQ-Pay for the order
KhaiLQ-Transaction Infor
KhaiLQ-See the list of orders
KhaiLQ-Approve orders
KhaiLQ-Reject orders
KhaiLQ-Login
KhaiLQ_Add a new DVD
KhaiLQ_Remove DVD
Customer_KhaiLQ
Store Manager_KhaiLQ
<<include>>
<<extend>>

10 Class Diagram