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

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

Mã học phần: IT3103
Mã lớp: 744530

Giảng viên hướng dẫn: Cô Lê Thị Hoa
Sinh viên thực hiện: Nguyễn Mạnh Tùng
MSSV: 20225682

Hà Nội, tháng 12 năm 2024

Contents

1. Swing components

1.1 AWTAccumulator

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

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

public class AWTAccumulatorTungNM extends Frame{
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;          // Tổng tích lũy, khởi tạo bằng 0

    // Constructor để thiết lập GUI và các bộ xử lý sự kiện
    public AWTAccumulatorTungNM() {
        setLayout(new GridLayout(rows:2, cols:2));

        // Nhãn và ô nhập dữ liệu
        add(new Label(text:"Enter an Integer: "));
        tfInput = new TextField(columns:10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        add(new Label(text:"The Accumulated Sum is: "));
        tfOutput = new TextField(columns:10);
        tfOutput.setEditable(b:false);
        add(tfOutput);

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

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

    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

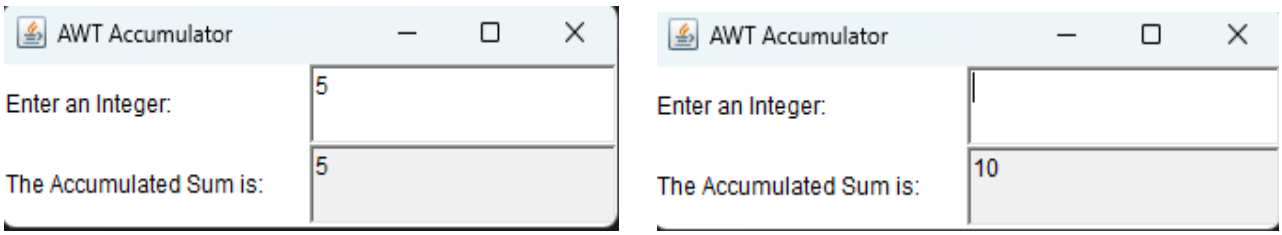


Figure 1.2: Demo of AWTAccumulator

1.2 SwingAccumulator

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

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

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

    // Constructor to setup the GUI components and event handlers
    public SwingAccumulatorTungNM() {
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(rows:2, cols:2));

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

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

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

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

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

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

    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

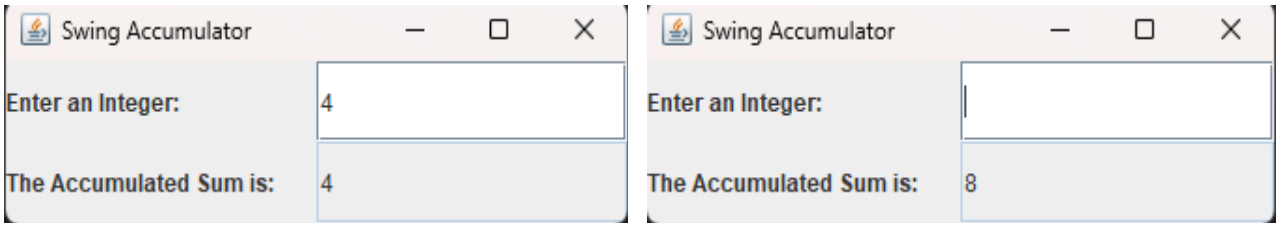


Figure 1.4: Demo of SwingAccumulator

2 Organizing Swing components with Layout Managers

2.1 Code

```java
package hust.soict.dsai.aims.GUIproject;

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

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

    public NumberGridTungNM() {
        tfDisplay = new JTextField();
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
        tfDisplay.setEditable(b:false);
        tfDisplay.setFont(new Font(name:"Arial", Font.BOLD, size:20));

        JPanel panelButtons = new JPanel(new GridLayout(rows:4, cols:3, hgap:5, vgap:5));
        addButtons(panelButtons);

        Container cp = getContentPane();
        cp.setLayout(new BorderLayout(hgap:5, vgap:5));
        cp.add(tfDisplay, BorderLayout.NORTH);
        cp.add(panelButtons, BorderLayout.CENTER);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle(title:"Number Grid - Nguyen Manh Tung 20225682");
        setSize(width:300, height:400);
        setLocationRelativeTo(c:null);
        setVisible(b:true);
    }

```

Figure 2.1: Source code of NumberGrid 1

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

Figure 2.2: Source code of NumberGrid 2

2.2 Demo

Figure 2.3: Demo buttons 0-9

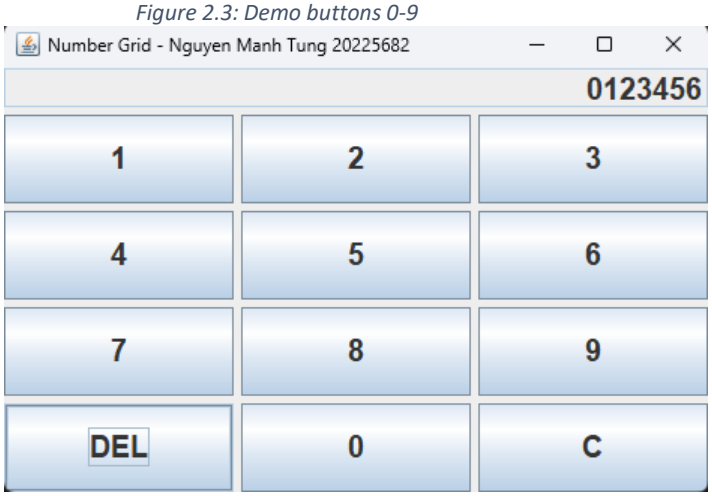


Figure 2.4: Demo DEL button

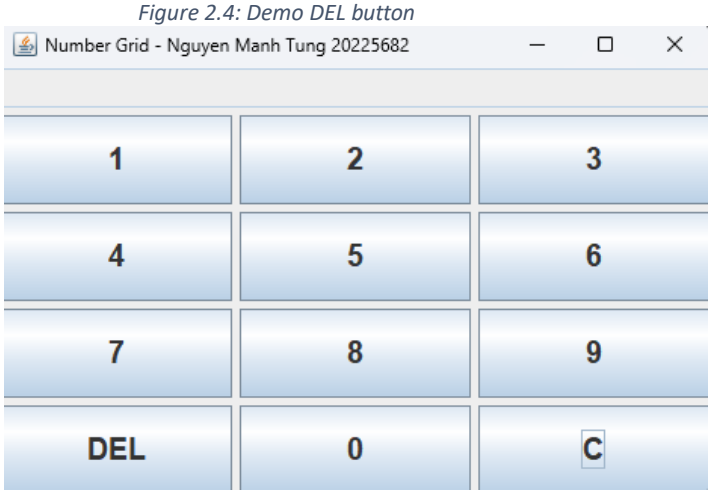


Figure 2.5: Demo C button

3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

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

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

import hust.soict.dsai.aims.media.MediaTungNM;
import hust.soict.dsai.aims.media.StoreTungNM;

public class StoreScreenTungNM extends JFrame {
    private StoreTungNM store;

    public StoreScreenTungNM(StoreTungNM store) {
        this.store = store;

        // Cài đặt Container chính của JFrame
        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(createNorth(), BorderLayout.NORTH);
        cp.add(createCenter(), BorderLayout.CENTER);

        // Cài đặt các thuộc tính JFrame
        setTitle(title:"Store Nguyen Manh Tung 20225682");
        setSize(width:1024, height:768);
        setVisible(b:true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    // Tạo phần phía trên (MenuBar + Header)
    JPanel createNorth() {
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }
```

Figure 3.1: Class StoreScreen 1

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

Figure 3.2: Class StoreScreen 2

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

Figure 3.3: Class StoreScreen 3

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

Figure 3.4: Class StoreScreen 4

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

Figure 3.5: Class TestStoreScreen 5

3.2 Create class MediaStore

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

import javax.swing.*;
import hust.soict.dsai.aims.media.MediaTungNM;
import hust.soict.dsai.aims.media.PlayableTungNM;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MediaStoreTungNM extends JPanel {
    private MediaTungNM media;

    public MediaStoreTungNM(MediaTungNM media) {
        this.media = media;
        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("" + media.getPrice() + " $");
        cost.setAlignmentX(CENTER_ALIGNMENT);

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

Figure 3.7: Class MediaStore 1

```java
        // Add "Add to Cart" Button
        JButton addToCartButton = new JButton(text:"Add to cart");
        addToCartButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Show success message
                JOptionPane.showMessageDialog(
                    MediaStoreTungNM.this,
                    "Added " + media.getTitle() + " to the cart successfully!",
                    title:"Add to Cart",
                    JOptionPane.INFORMATION_MESSAGE
                );
            }
        });
        container.add(addToCartButton);
```

Figure 3.8: Class MediaStore 2

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

Figure 3.9: Class MediaStore 3

3.3 Demo

Figure 3.10: StoreScreen

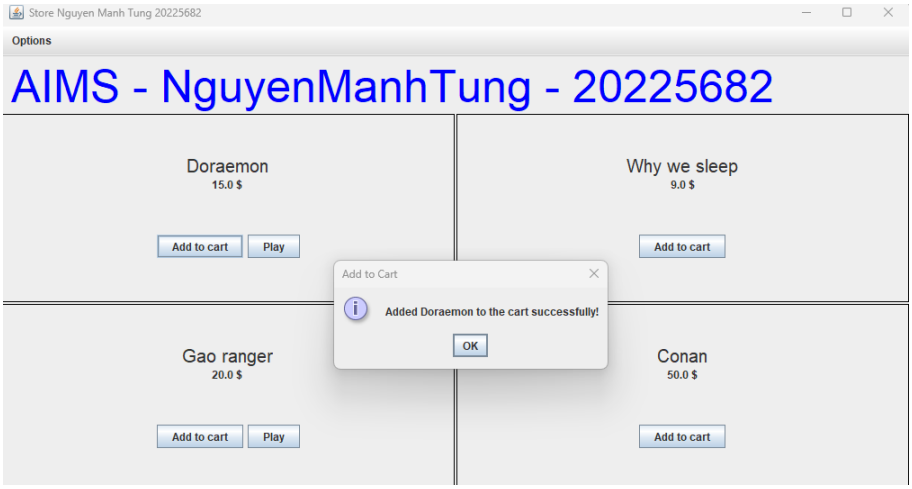


Figure 3.11 Demo Add to cart button

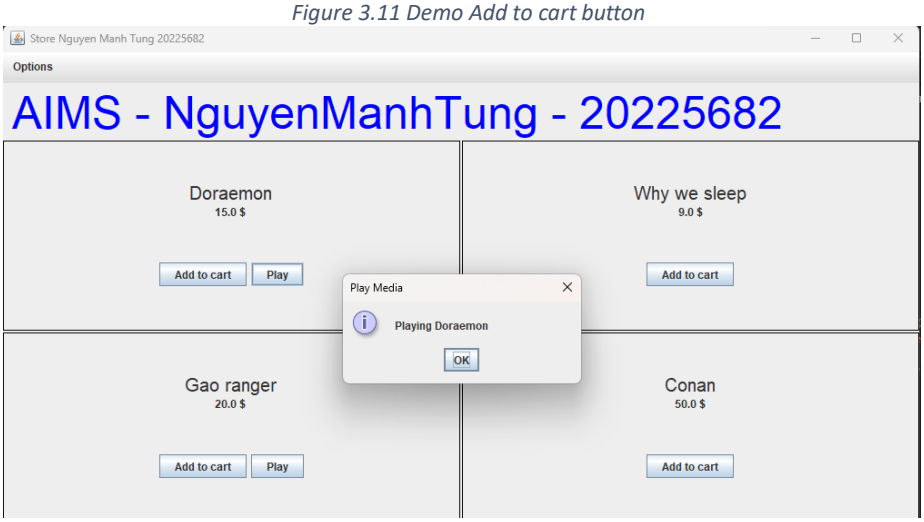


Figure 3.12 Demo Play button

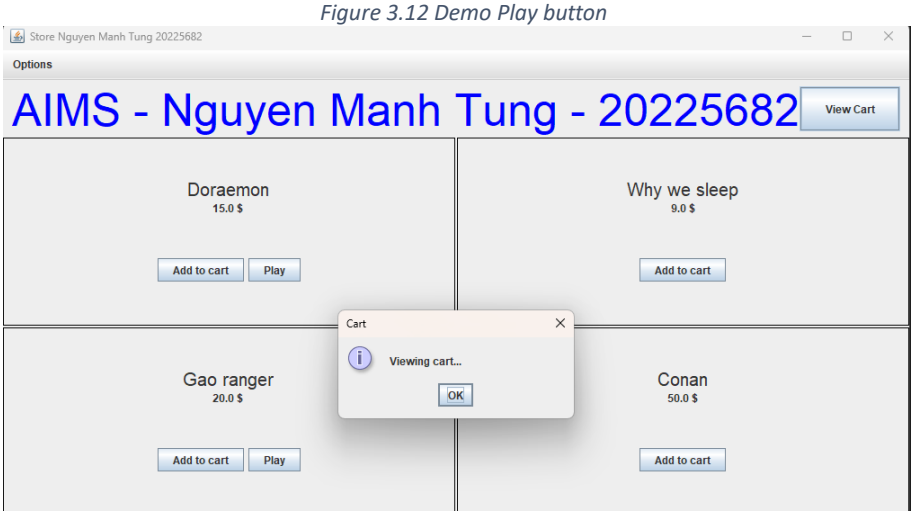


Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

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

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

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

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

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

```

Figure 4.1: Class Painter

4.2 Create Painter.fxml

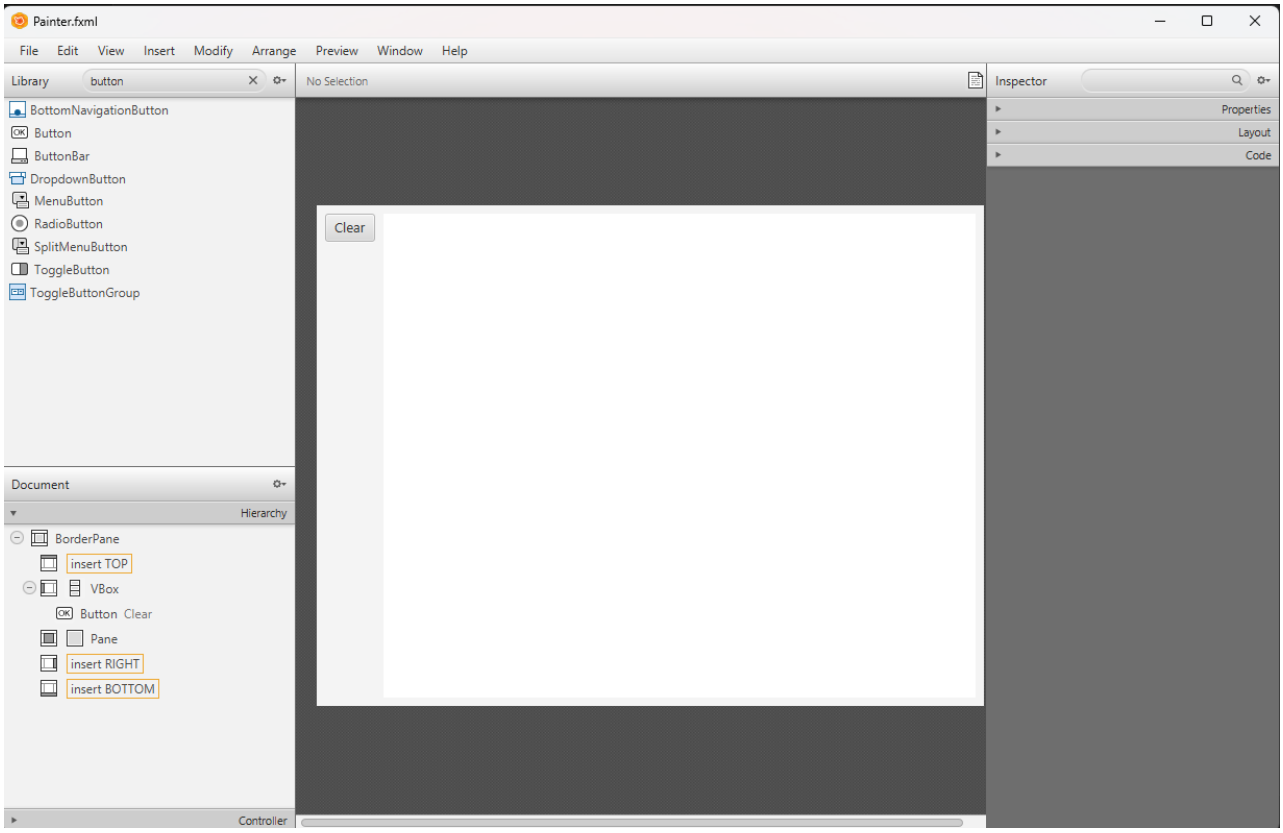


Figure 4.2: Painter.fxml 1

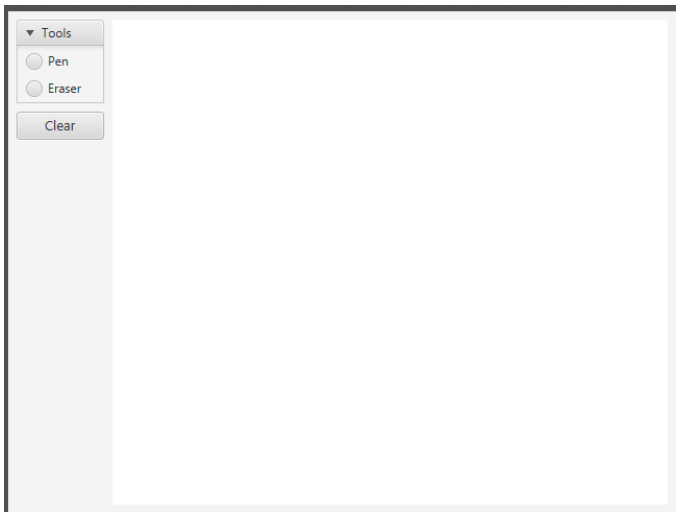


Figure 4.3: Painter.fxml 2

4.3 Create class PainterController

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

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

public class PainterControllerTungNM {

    @FXML
    private Pane drawingAreaPane;

    @FXML
    private RadioButton eraser;

    @FXML
    private RadioButton pen;

    private Color currentColor = Color.BLACK;

    @FXML
    void toolSelection(ActionEvent event) {
        if (pen.isSelected()) {
            currentColor = Color.BLACK;
        } else if (eraser.isSelected()) {
            currentColor = Color.WHITE;
        }
    }

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

    @FXML
    void initialize() {
        drawingAreaPane.setOnMouseDragged(this::drawingAreaMouseDragged);
    }

    void drawingAreaMouseDragged(MouseEvent event) {
        Circle newCircle = new Circle(event.getX(), event.getY(), 4, currentColor);
        drawingAreaPane.getChildren().add(newCircle);
    }
}

```

Figure 4.4: PainterController

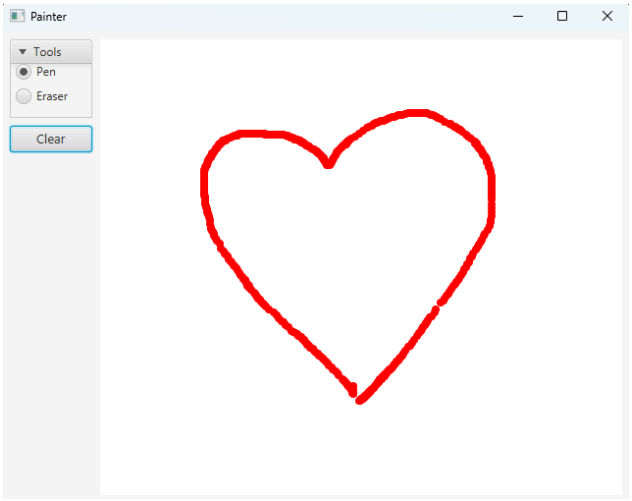


Figure 4.5: Use Pen

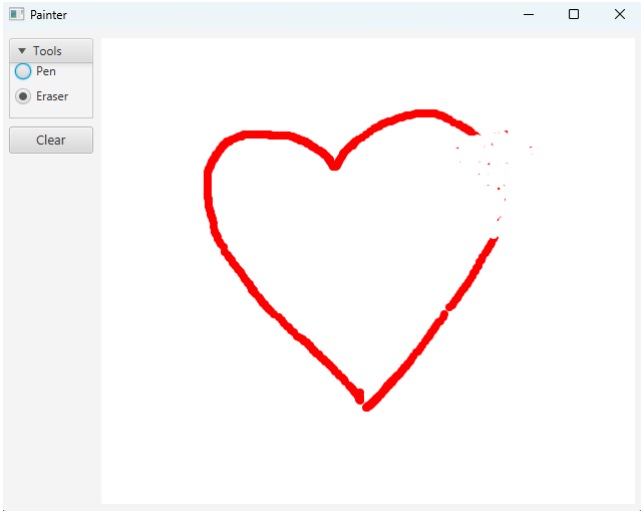


Figure 4.6: Use Eraser

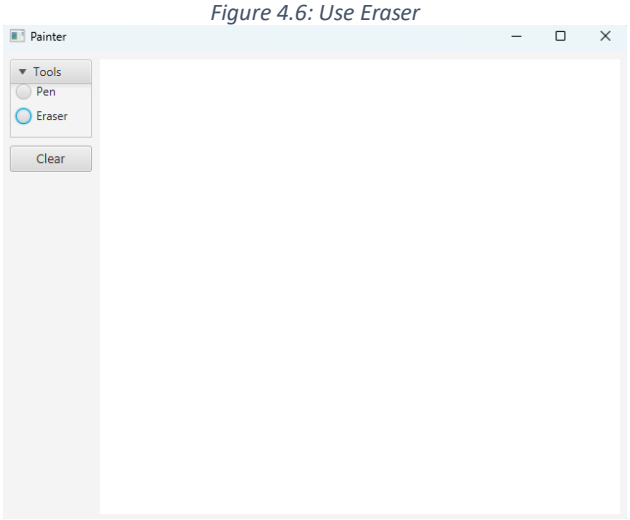


Figure 4.7: Clear button

5 View Cart Screen

5.1 Create cart.fxml

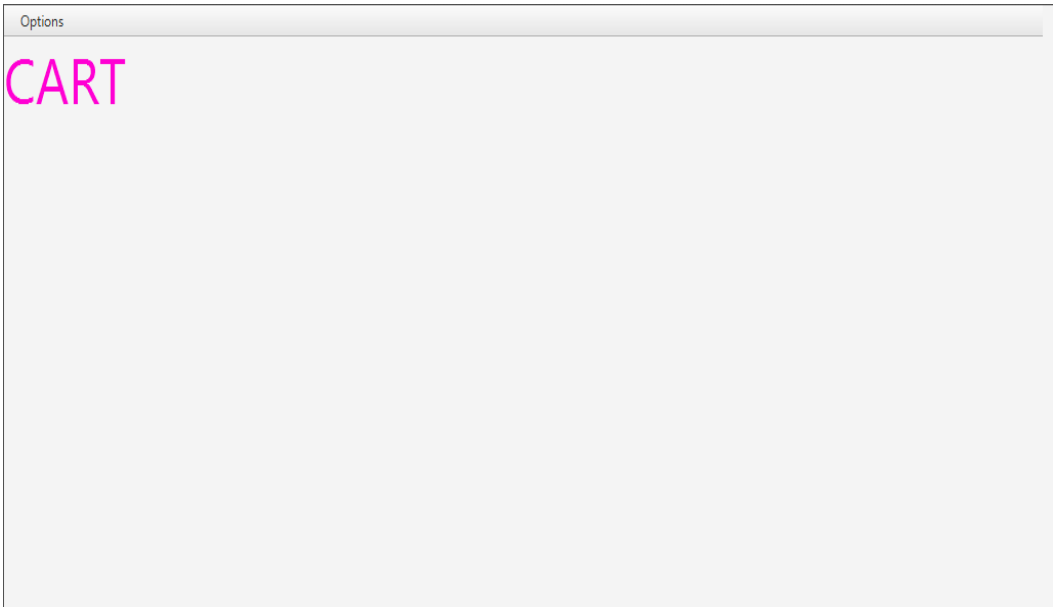


Figure 5.1: Cart.fxml 1

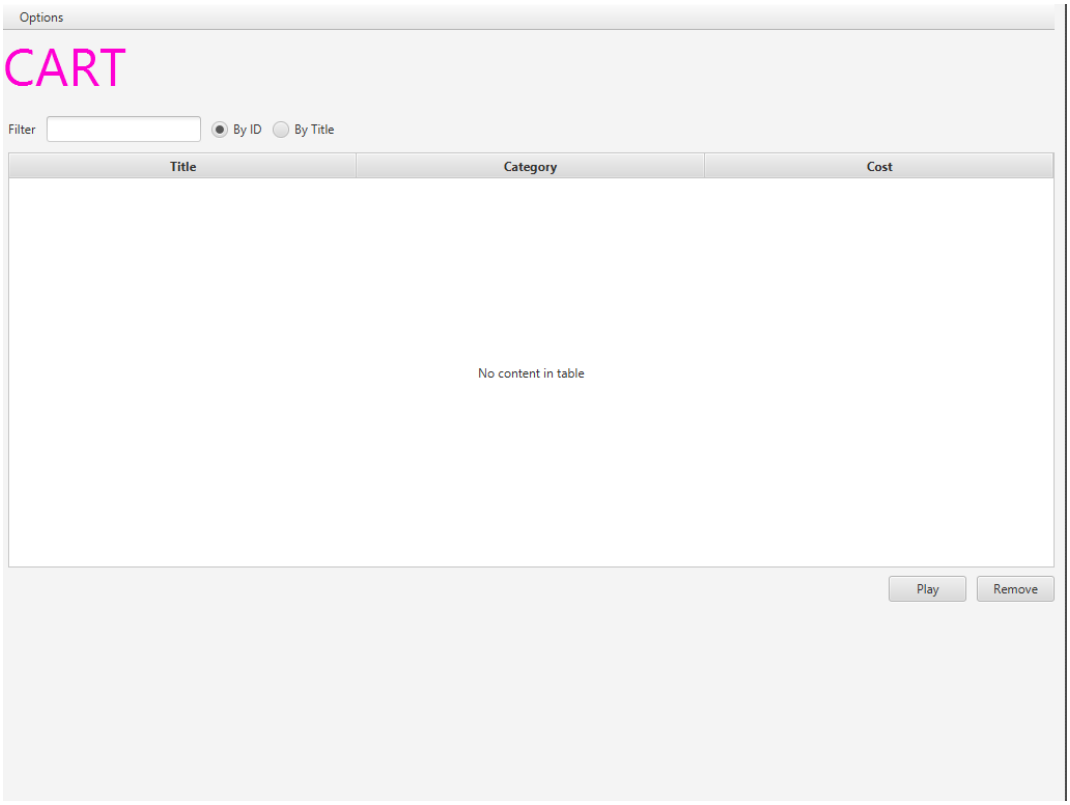


Figure 5.2: Cart.fxml 2

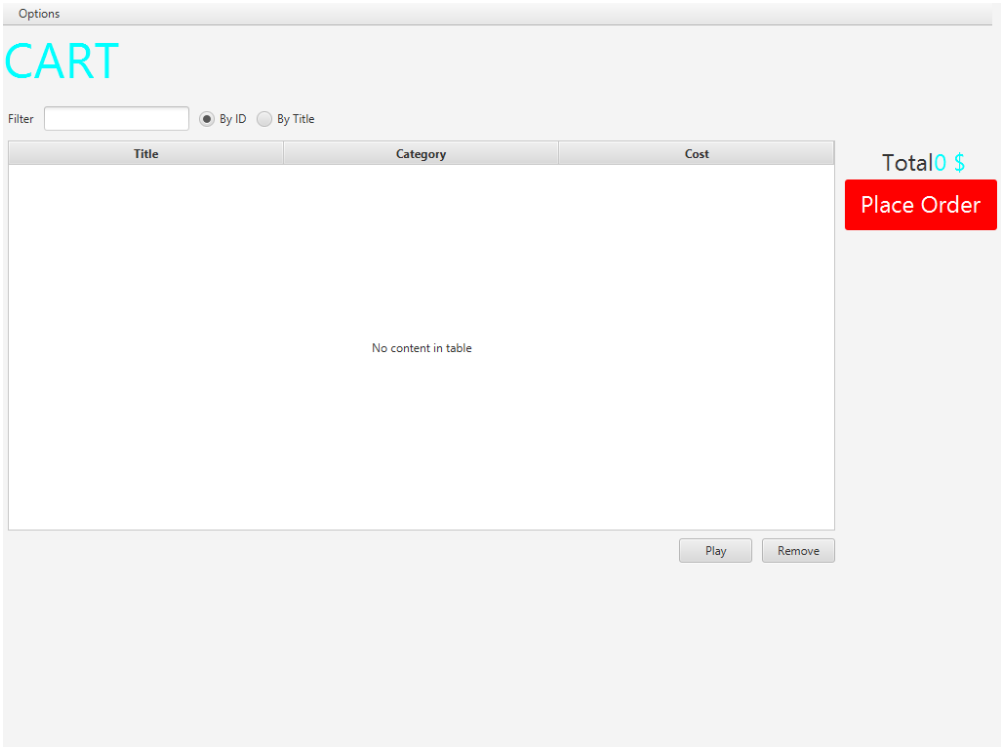


Figure 5.3: Cart.fxml 3

5.2 Create class CartScreen

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

import java.io.IOException;

import javax.swing.JFrame;

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

public class CartScreenTungNM extends JFrame{
    private CartTungNM cart;
    public CartScreenTungNM(CartTungNM cart) {
        super();
        this.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle(title:"Cart");
        this.setVisible(b:true);

        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"/hust/soict/dsai/aims/cart/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
package hust.soict.dsai.aims.cart;

import hust.soict.dsai.aims.media.CartTungNM;
import hust.soict.dsai.aims.media.MediaTungNM;
import javafx.fxml.FXML;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.cell.PropertyValueFactory;

public class CartScreenController {
    private CartTungNM cart;

    @FXML
    private TableColumn<MediaTungNM, String> colMediaCategory;

    @FXML
    private TableColumn<MediaTungNM, Float> colMediaCost;

    @FXML
    private TableColumn<MediaTungNM, String> colMediaTitle;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private TableView<MediaTungNM> tblMedia;

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

Figure 5.5: CartScreenController 1

```java
    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Title"));
        colMediaCategory.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<MediaTungNM, Float>("Cost"));
        tblMedia.setItems(this.cart.getItemsOrdered());
    }

}
```

Figure 5.6: CartScreenController 2

5.4 Demo

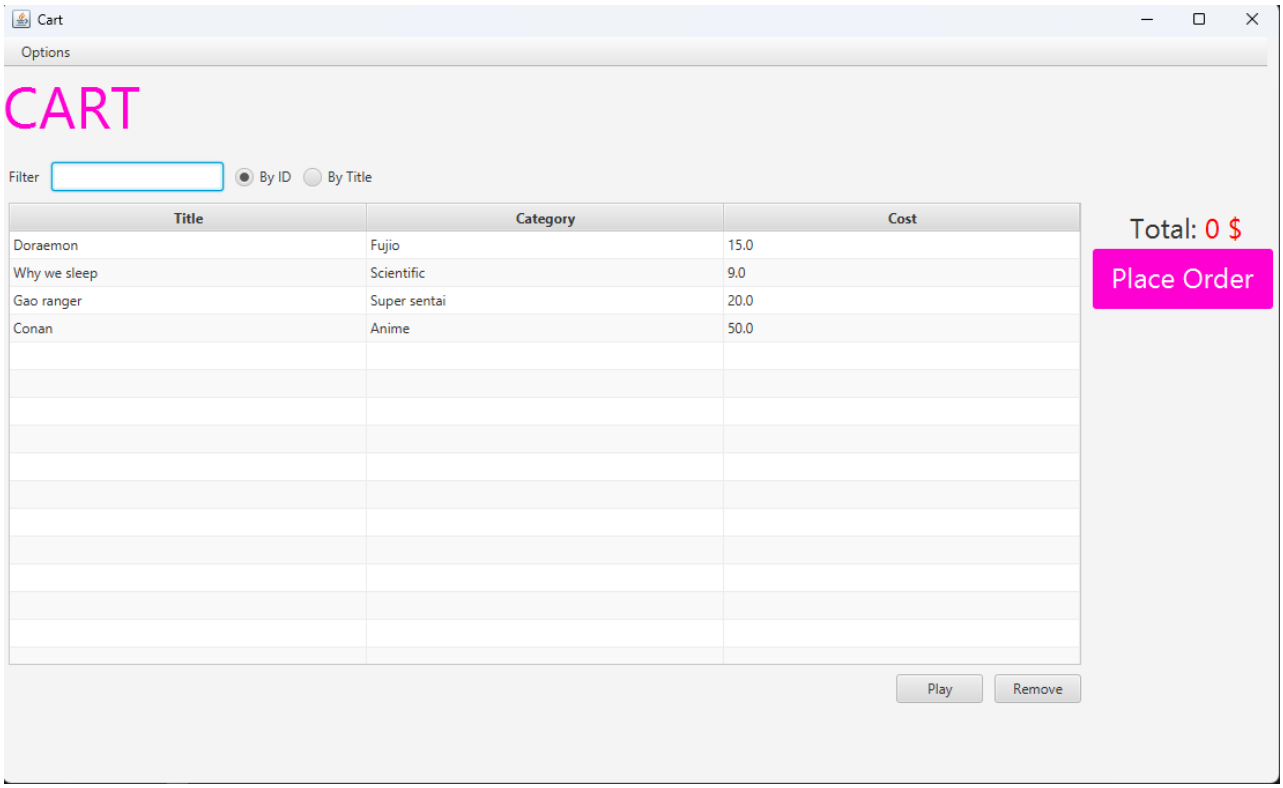


Figure 5.7: Demo CartScreen

6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController

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

import hust.soict.dsai.aims.media.CartTungNM;
import hust.soict.dsai.aims.media.MediaTungNM;
import hust.soict.dsai.aims.media.PlayableTungNM;
import javafx.beans.Observable;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.cell.PropertyValueFactory;

public class CartScreenController {
    private CartTungNM cart;

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;
```

Figure 6.1: CartScreenController 1

```java
    @FXML
    private TableColumn<MediaTungNM, String> colMediaCategory;

    @FXML
    private TableColumn<MediaTungNM, Float> colMediaCost;

    @FXML
    private TableColumn<MediaTungNM, String> colMediaTitle;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private TableView<MediaTungNM> tblMedia;

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

    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Title"));
        colMediaCategory.setCellValueFactory(new PropertyValueFactory<MediaTungNM, String>("Category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<MediaTungNM, Float>("Cost"));
        tblMedia.setItems(this.cart.getItemsOrdered());

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

        tblMedia.getSelectionModel().selectedItemProperty().addListener(
            new ChangeListener<MediaTungNM>() {
                @Override
                public void changed(ObservableValue<? extends MediaTungNM> observable, MediaTungNM oldValue, MediaTungNM newValue) {
                    if (newValue != null) {
                        updateButtonBar(newValue);
                    }
                }
            });
    }

    void updateButtonBar(MediaTungNM media) {
        btnRemove.setVisible(true);
        if(media instanceof PlayableTungNM) btnPlay.setVisible(true);
        else btnPlay.setVisible(false);
    }
}
```

Figure 6.2: CartScreenController 2

6.2 Demo

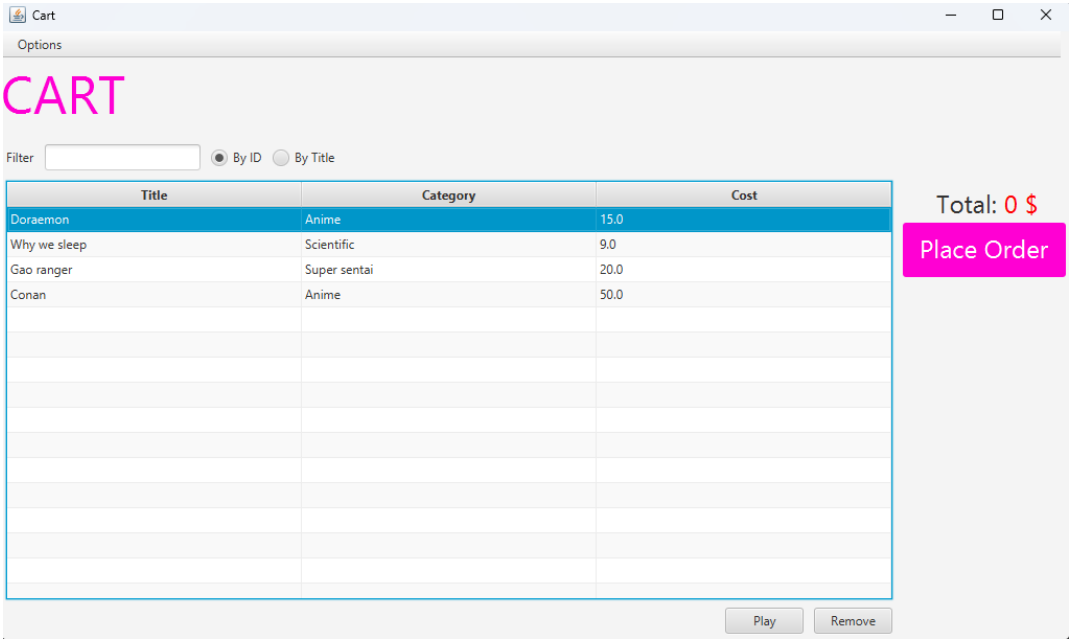


Figure 6.3: Demo media playable

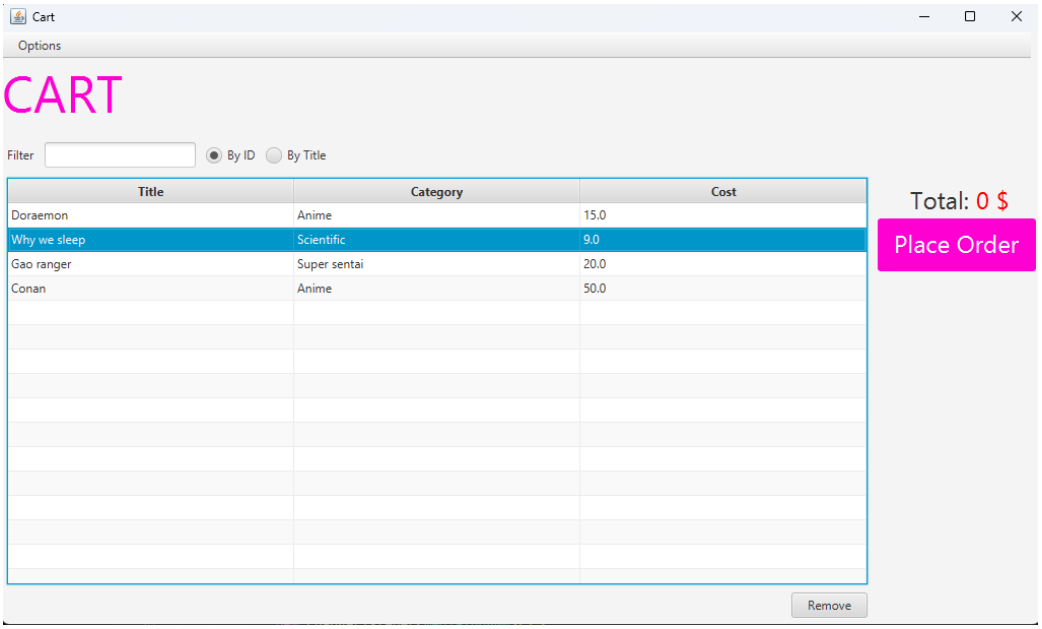


Figure 6.4: Demo media unplayable

7 Deleting a media

7.1 Code

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

Figure 7.1: btnRemovePressed Method

7.2 Demo

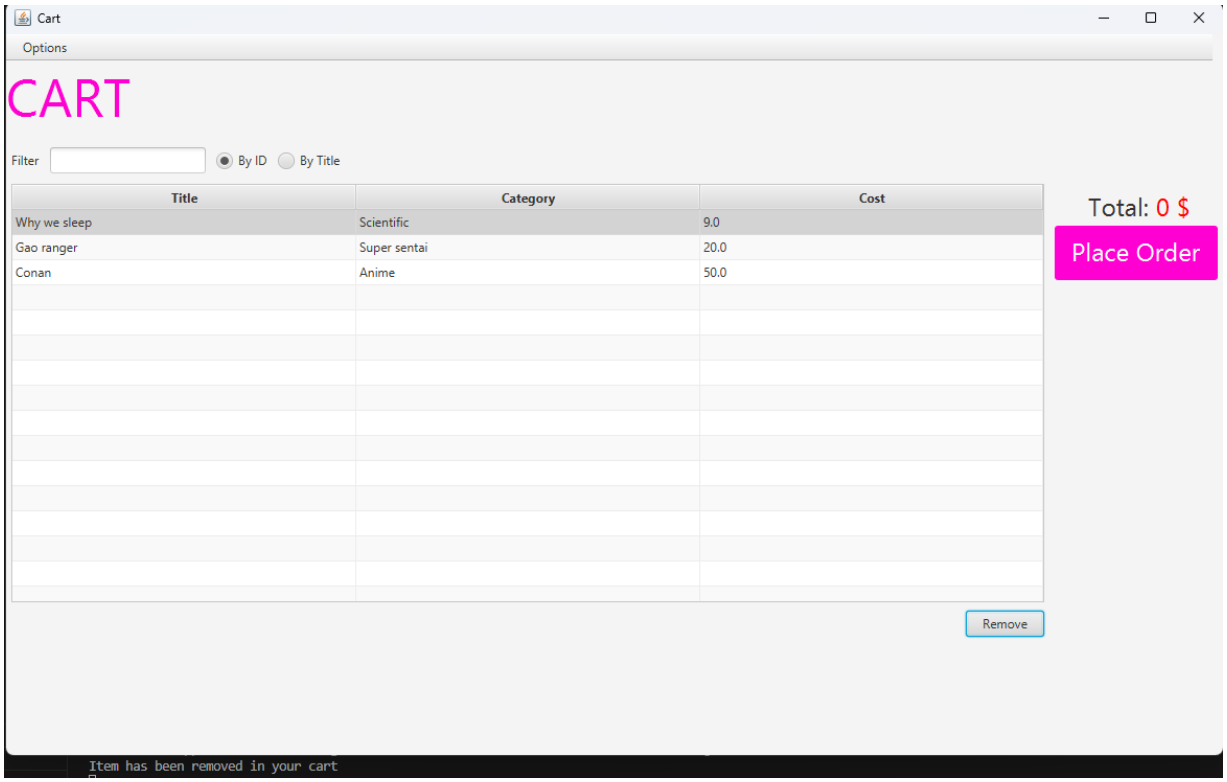


Figure 7.2: button Remove

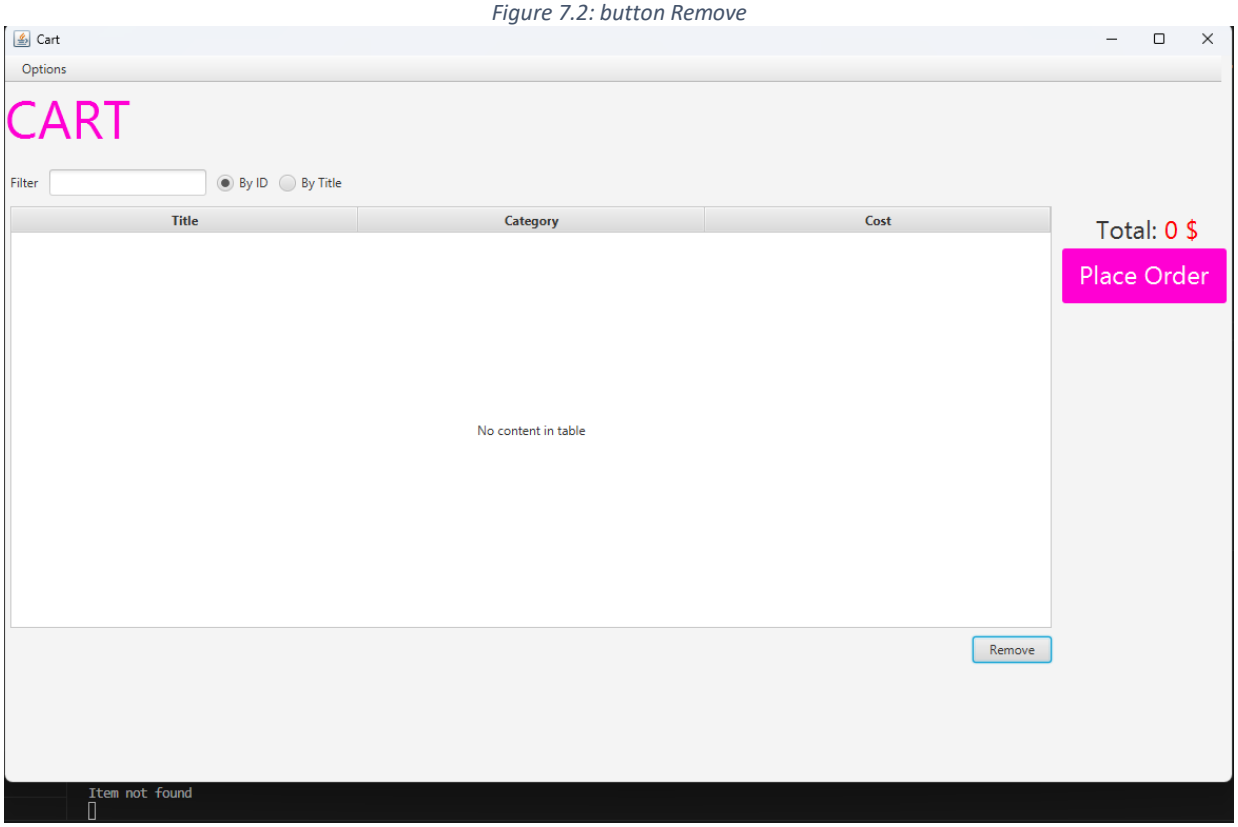


Figure 7.3: button Remove

8 Complete the Aims GUI application

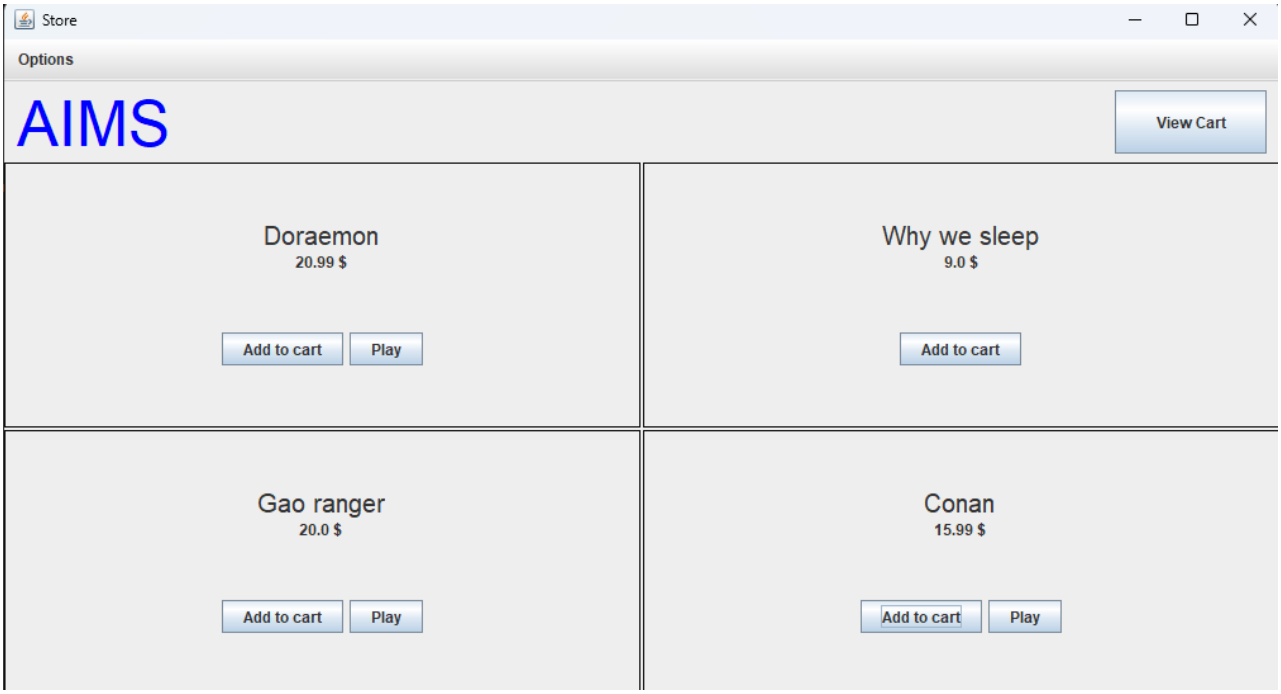


Figure 8.1: Store before add book

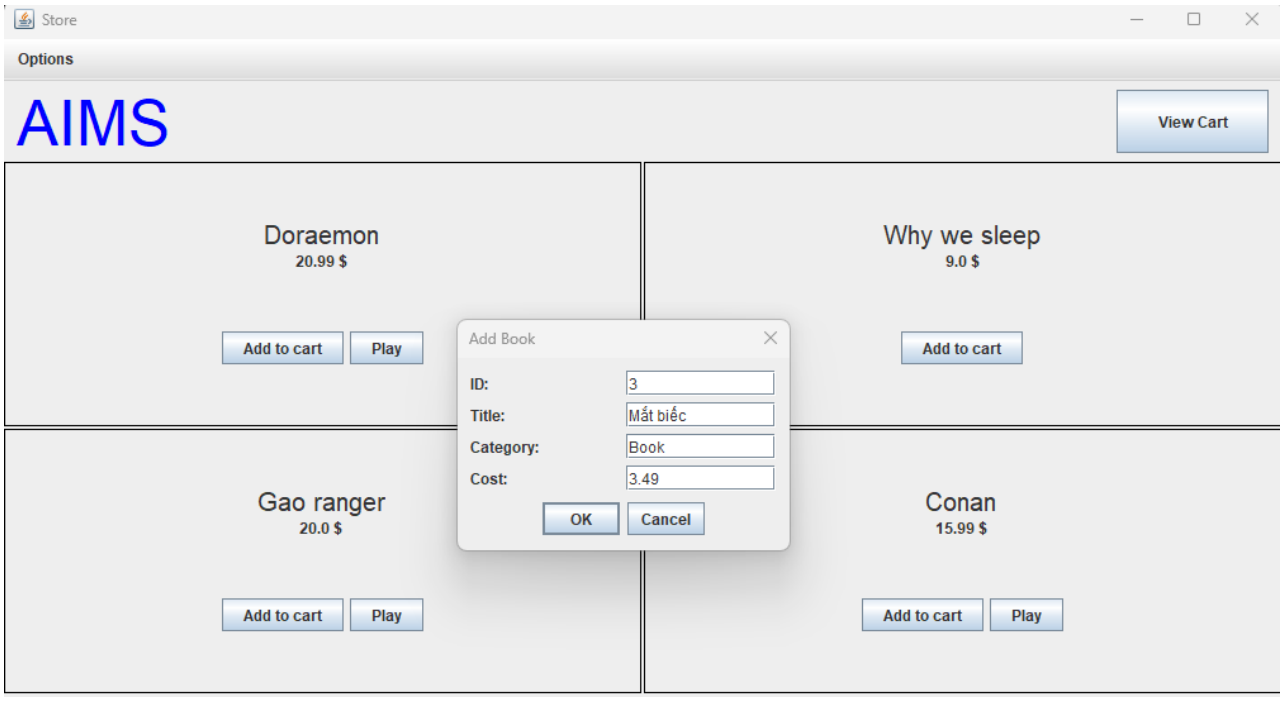


Figure 8.2: Add book

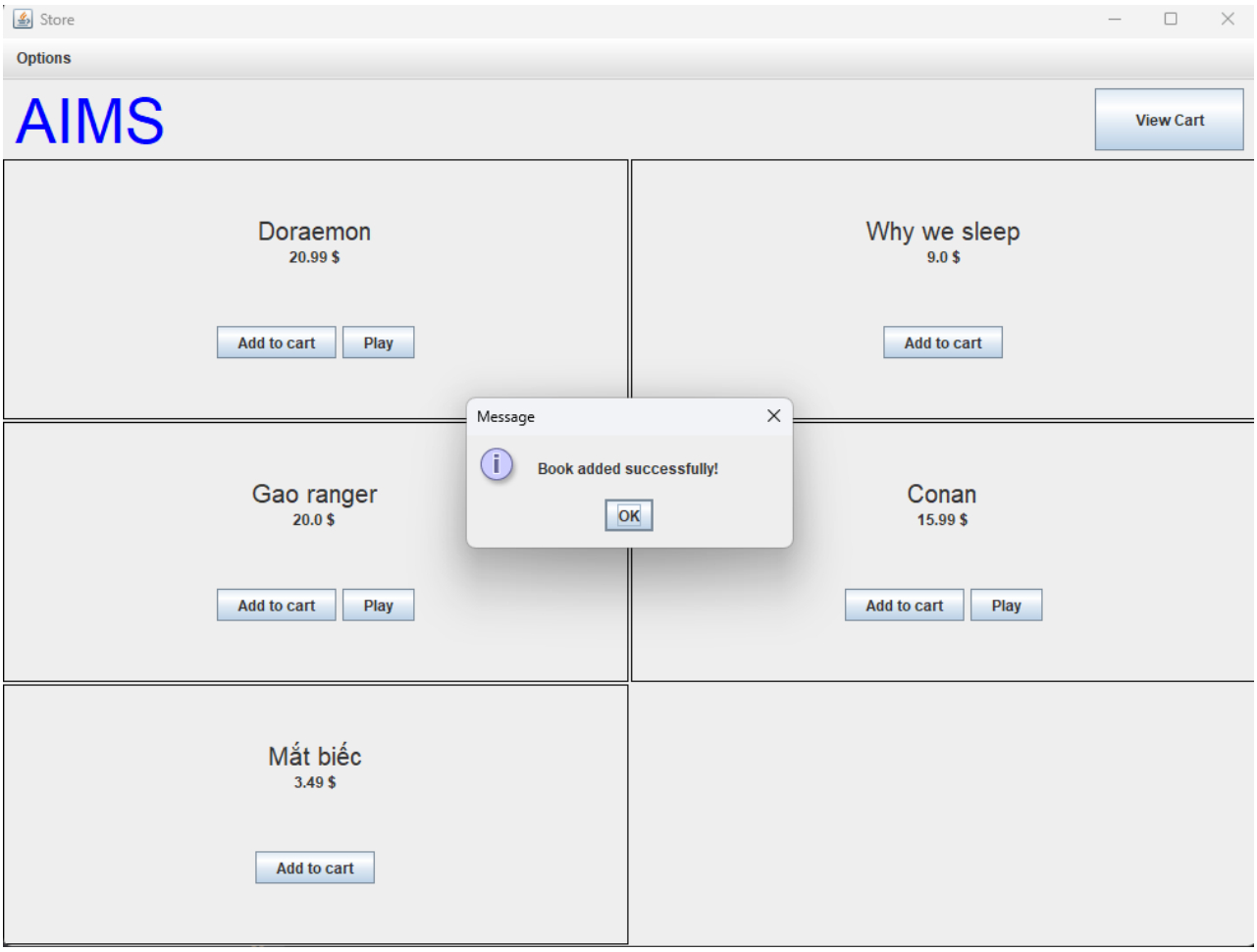


Figure 8.3: Store after add book

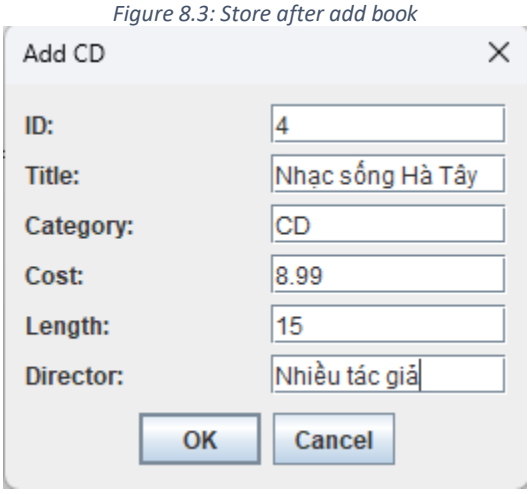


Figure 8.4: Add CD

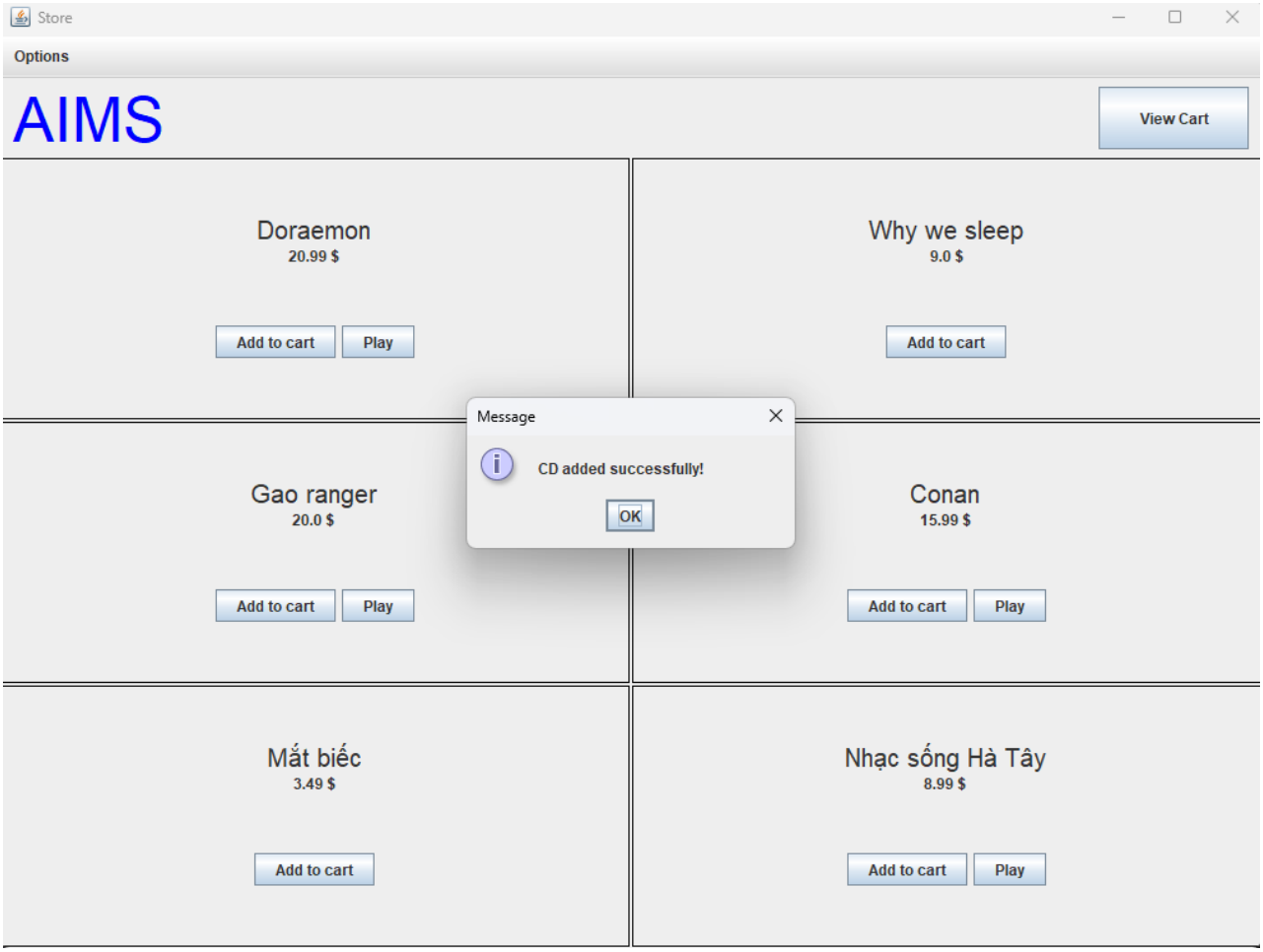


Figure 8.5: Store after add CD

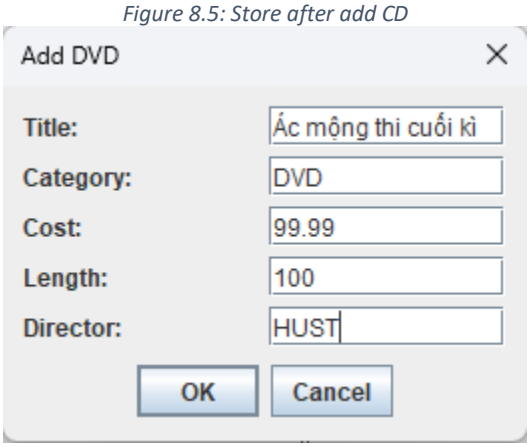


Figure 8.6 Add DVD

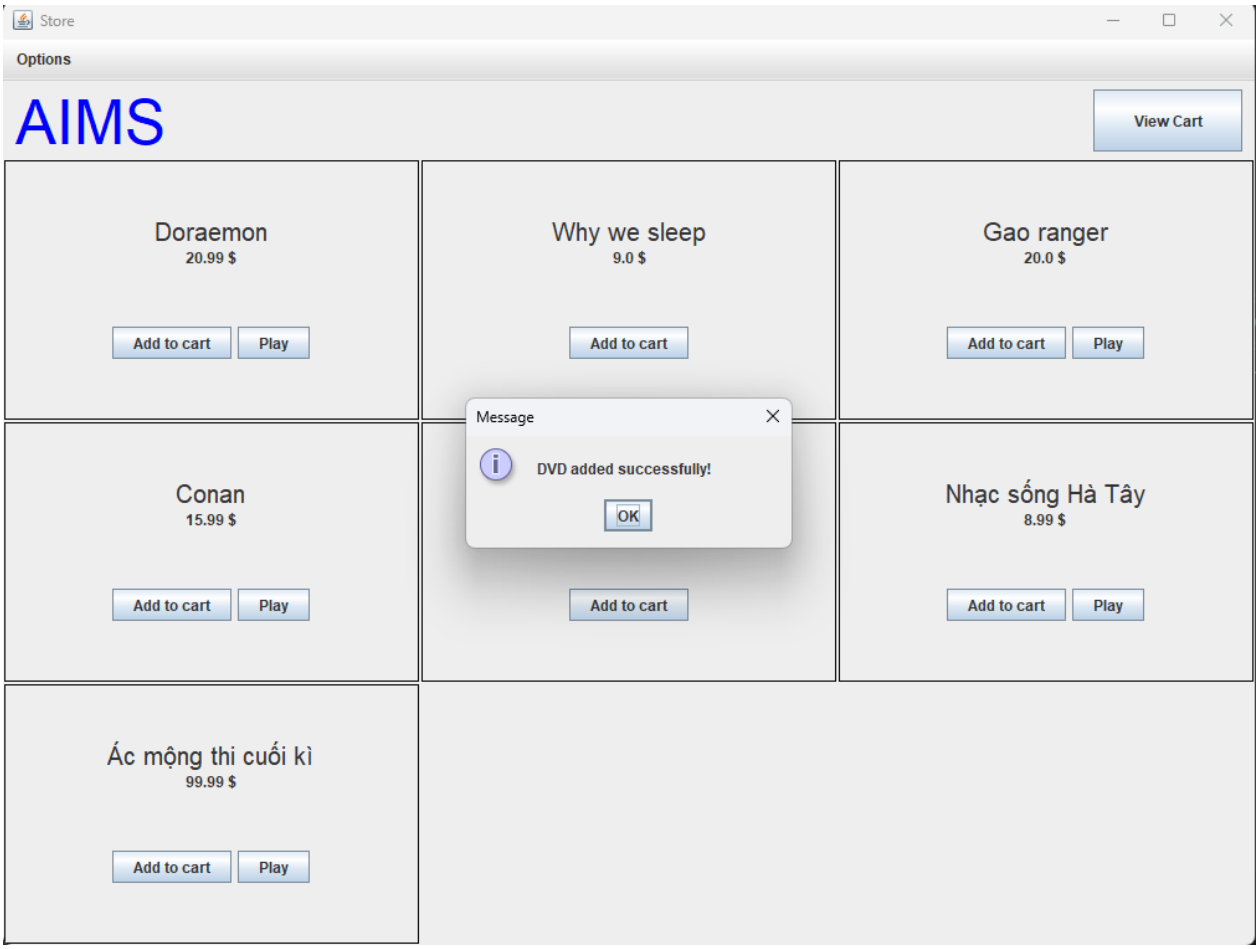


Figure 8.7: Store after add DVD

Figure 8.8: Cart

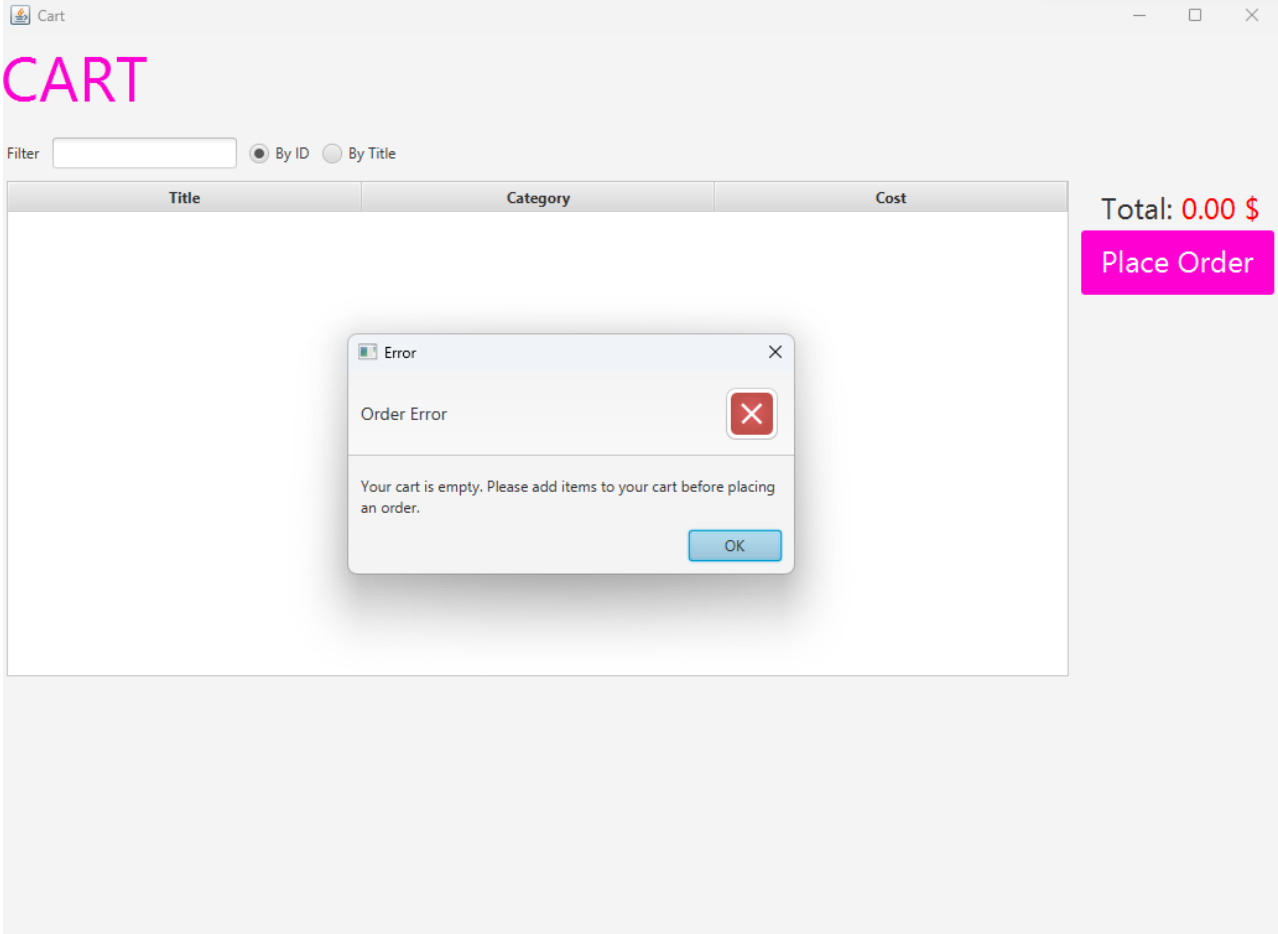
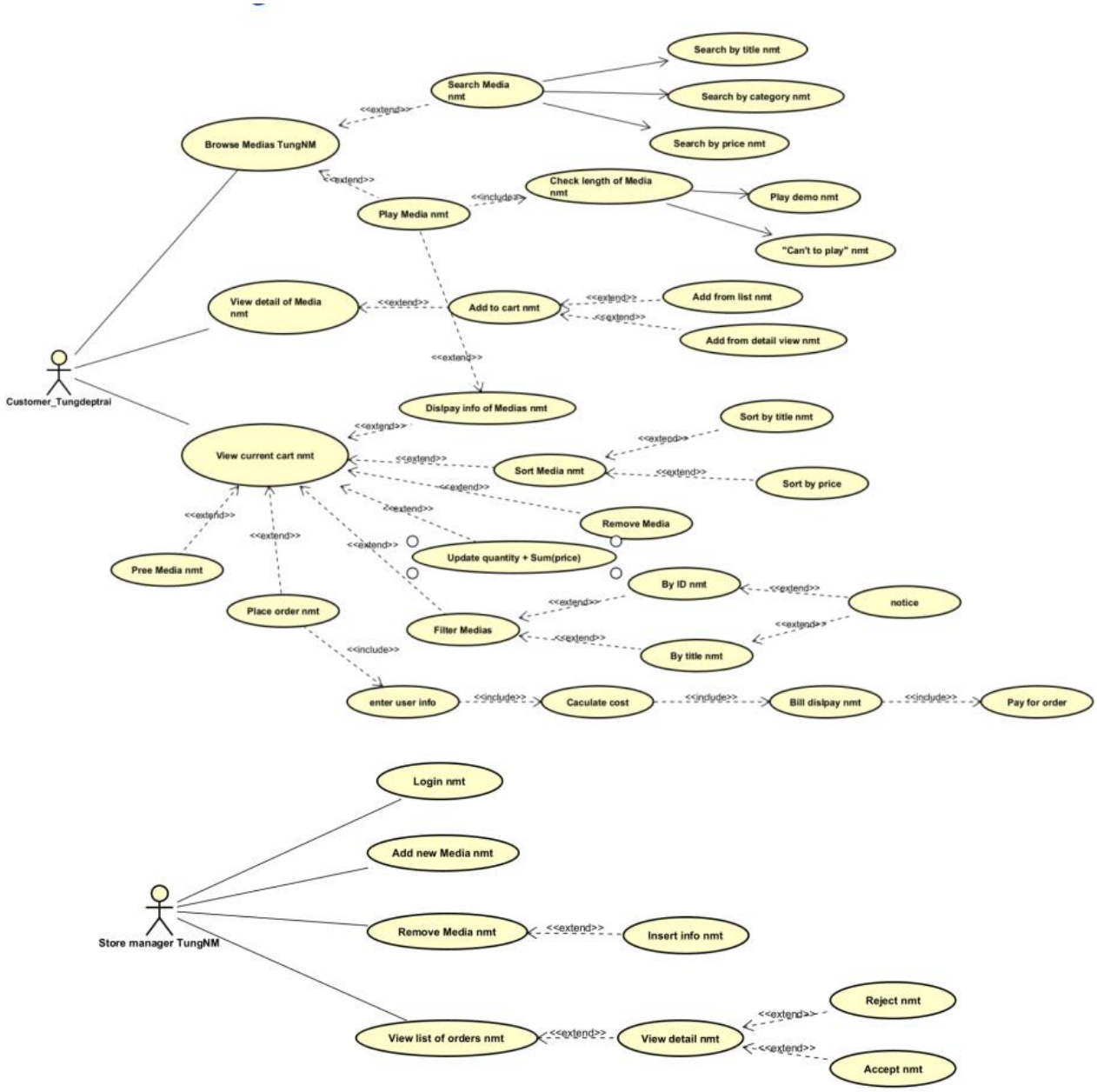


Figure 8.9: Exception

9 Use case Diagram


Search by title nmt
Search Media nmt
Search by category nmt
Search by price nmt
<<extend>>
Browse Medias TungNM
<<extend>>
Check length of Media nmt
<<include>>
Play demo nmt
Play Media nmt
"Can't to play" nmt
View detail of Media nmt
<<extend>>
Add to cart nmt
<<extend>>
Add from list nmt
<<extend>>
Add from detail view nmt
<<extend>>
Customer_Tungdeptrai
Dislpay info of Medias nmt
<<extend>>
Sort by title nmt
<<extend>>
View current cart nmt
<<extend>>
Sort Media nmt
<<extend>>
Sort by price
<<extend>>
<<extend>>
Remove Media
<<extend>>
<<extend>>
<<extend>>
Update quantity + Sum(price)
Pree Media nmt
By ID nmt
<<extend>>
notice
<<extend>>
Place order nmt
<<extend>>
Filter Medias
<<extend>>
By title nmt
<<include>>
enter user info
<<include>>
Caculate cost
<<include>>
Bill dislpay nmt
<<include>>
Pay for order
Login nmt
Add new Media nmt
Store manager TungNM
Remove Media nmt
<<extend>>
Insert info nmt
Reject nmt
<<extend>>
View list of orders nmt
<<extend>>
View detail nmt
<<extend>>
Accept nmt

10 Class Diagram