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

BÁO CÁO THỰC HÀNH
IT3103-744528-2024.1

BÀI THỰC HÀNH 5

Họ và tên sv: Đỗ Hoàng Đông
MSSV: 20225807
Lớp: Việt Nhật 03 – K67
GVHD: Lê Thị Hoa
HTGD: Đặng Mạnh Cường

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 5
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. Swing components

1.1 AWTAccumulator

```java
//Do Hoang Dong 20225807
package hust.soict.hedspi.aims.swing;

import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class AWTAccumulator extends Frame {
    private TextField tfInput;    // Ô nhập dữ liệu
    private TextField tfOutput;   // Ô hiển thị kết quả
    private int sum = 0;          // Tổng tích lũy ban đầu là 0

    // Constructor để thiết lập giao diện và các sự kiện
    public AWTAccumulator() {
        setLayout(new GridLayout(rows: 2, cols: 2)); // Layout dạng lưới 2x2

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

        // Nhãn và ô hiển thị kết quả
```

Figure 1.1: Source code of AWTAccumulator

```java
        // Nhãn và ô hiển thị kết quả
        add(new Label(text: "The Accumulated Sum is: "));
        tfOutput = new TextField(columns:10);
        tfOutput.setEditable(b: false); // Không cho chỉnh sửa kết quả
        add(comp: tfOutput);

        // Cài đặt cửa sổ
        setTitle(title: "AWT Accumulator");
        setSize(width: 350, height: 120);
        setVisible(b: true);
    }

    // Hàm main để chạy chương trình
    public static void main(String[] args) {
        new AWTAccumulator();
    }

    // Lớp xử lý sự kiện khi nhấn Enter trong ô nhập
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(s: tfInput.getText()); // Lấy số nguy
            sum += numberIn; // Cộng dồn vào tổng
            tfInput.setText(t: ""); // Xóa ô nhập
            tfOutput.setText(sum + ""); // Hiển thị tổng tích lũy
        }
```

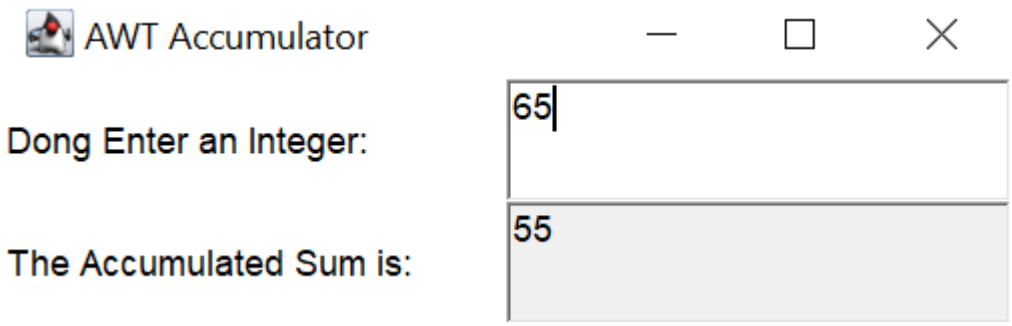


Figure 1.2: Demo of AWTAccumulator

1.2 SwingAccumulator

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

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

public class SwingAccumulator extends JFrame {
    private JTextField tfInput;     // Ô nhập số nguyên
    private JTextField tfOutput;    // Ô hiển thị kết quả
    private int sum = 0;            // Tổng tích lũy, khởi tạo bằng 0

    // Constructor để thiết lập giao diện và các xử lý sự kiện
    public SwingAccumulator() {
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(rows: 2, cols: 2)); // Layout dạng lưới 2x2

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

        // Nhãn và ô hiển thị kết quả
        cp.add(new JLabel(text: "The Accumulated Sum is: "));
        tfOutput = new JTextField(columns:10);
        tfOutput.setEditable(b: false); // Không cho phép chỉnh sửa ô kết quả
        cp.add(comp: tfOutput);
```

Figure 1.3: Source code of SwingAccumulator

```java
        tfOutput.setEditable(b: false); // Không cho phép chỉnh sửa ô kết quả
        cp.add(comp: tfOutput);

        // Cài đặt cửa sổ JFrame
        setTitle(title: "Swing Accumulator");
        setSize(width: 350, height: 120);
        setVisible(b: true);
        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE); // Đóng cửa sổ 
    }

    // Hàm main để chạy chương trình
    public static void main(String[] args) {
        new SwingAccumulator();
    }

    // Lớp xử lý sự kiện khi nhấn Enter trong ô nhập
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(s: tfInput.getText()); // Lấy số ngu
            sum += numberIn; // Cộng dồn vào tổng tích lũy
            tfInput.setText(t: ""); // Xóa nội dung ô nhập
            tfOutput.setText(sum + ""); // Hiển thị tổng tích lũy
        }
    }
}
```

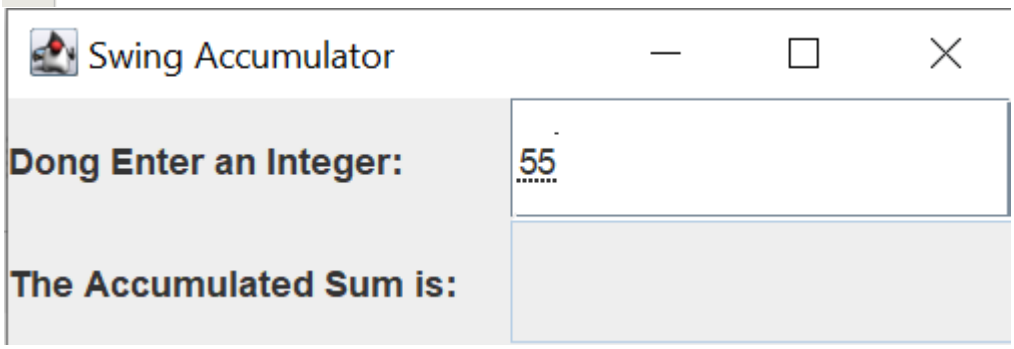


Figure 1.4: Demo of SwingAccumulator

2 Organizing Swing components with Layout Managers

2.1 Code

```java
//Do Hoang Dong 20225807
package hust.soict.hedspi.aims.swing;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

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

    public NumberGrid() {

        tfDisplay = new JTextField();
        tfDisplay.setPreferredSize(new Dimension(width: 200, height: 30));
        tfDisplay.setComponentOrientation(o: 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(comp: tfDisplay, constraints:BorderLayout.NORTH);
        cp.add(comp: panelButtons, constraints:BorderLayout.CENTER);

```

Figure 2.1: Source code of NumberGrid 1

```java
        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
        setTitle(title: "Mai Minh Quân - 20225661 - Number Grid");
        setSize(width: 200, height: 200);
        setVisible(b: true);
    }

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

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

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

        btnReset = new JButton(text: "C");
```

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

    private class ButtonListener implements ActionListener {

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

    }
    public static String delLastCharacter(String str) {
        if ((str != null) && (str.length() > 0)) {
```

Figure 2.2: Source code of NumberGrid 2

2.2 Demo

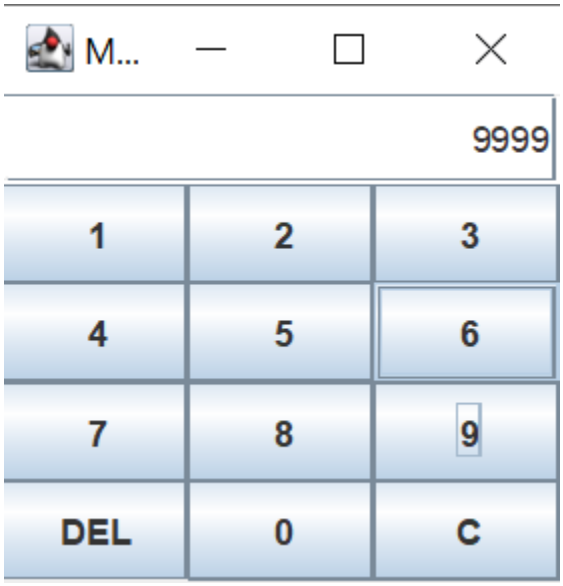


Figure 2.3: Demo buttons 0-9

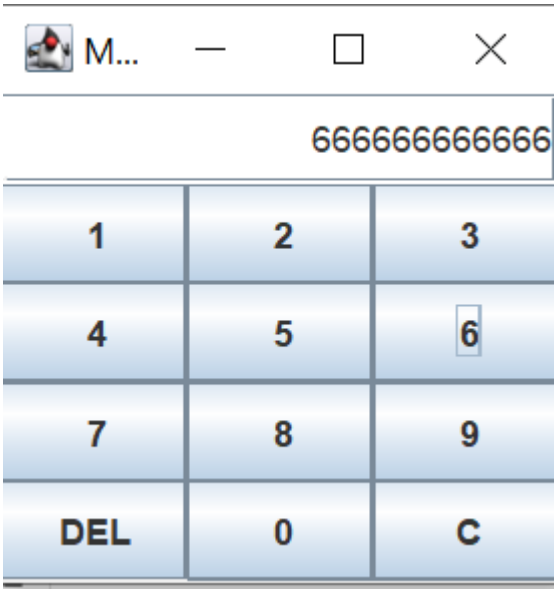


Figure 2.4: Demo DEL button

Figure 2.5: Demo C button

3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

```java
//Do Hoang Dong 20225807
package hust.soict.hedspi.aims.Screen;
import javax.swing.*;
import java.awt.*;
import java.util.*;
import java.awt.event.*;
import src.hust.soict.hedspi.aims.Cart.Cart;
import src.hust.soict.hedspi.aims.media.*;
import src.hust.soict.hedspi.aims.store.Store;

public class StoreScreen extends JFrame {
    private static Store store = new Store();
    private static Cart cart = new Cart();


    public static void main(String[] args) {
        // initSetup();
                new StoreScreen(store);
        }

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

        cp.add(comp: createNorth(), constraints:BorderLayout.NORTH);
```

Figure 3.1: Class StoreScreen 1

```
        cp.add(comp: createNorth(), constraints:BorderLayout.NORTH);
        cp.add(comp: createCenter(), constraints:BorderLayout.CENTER);

        setTitle(title: "Do Hoang Dong - 20225807 - Store");
                setSize(width: 1024, height: 768);
                setVisible(b: true);
        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
    }

    JPanel createNorth() {
                JPanel north = new JPanel();
                north.setLayout(new BoxLayout(target: north, axis:BoxLayout.Y_AXIS));
                north.add(comp: createMenuBar());
                north.add(comp: createHeader());
                return north;
        }

    JMenuBar createMenuBar() {

        JMenu menu = new JMenu(s: "Options");

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

```
        JMenuItem smAddCD = new JMenuItem(text: "Add CD");
        JMenuItem smAddDVD = new JMenuItem(text: "Add DVD");
//        smUpdateStore.add(smAddBook);
//        smUpdateStore.add(smAddCD);
//        smUpdateStore.add(smAddDVD);
//
//        smAddBook.addActionListener(new btnMenuListener());
//        smAddCD.addActionListener(new btnMenuListener());
//        smAddDVD.addActionListener(new btnMenuListener());
//
        menu.add(menuItem: smUpdateStore);

        JMenuItem viewStoreMenu = new JMenuItem(text: "View store");
        JMenuItem viewCartMenu = new JMenuItem(text: "View cart");
//        menu.add(viewStoreMenu);
//        menu.add(viewCartMenu);
//        viewStoreMenu.addActionListener(new ActionListener() {
//            @Override
//            public void actionPerformed(ActionEvent e) {
//                new StoreScreen(store);
//            }
//        });
//        viewCartMenu.addActionListener(new ActionListener() {
//            @Override
//            public void actionPerformed(ActionEvent e) {
```

Figure 3.2: Class StoreScreen 2

```java
//
    JPanel createHeader() {

        JPanel header = new JPanel();
        header.setLayout(new BoxLayout(target: header, axis: BoxLayout.X_AXIS));

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

        JButton cartBtn = new JButton(text: "View cart");
        cartBtn.setPreferredSize(new Dimension(width: 100, height: 50));
        cartBtn.setMaximumSize(new Dimension(width: 100, height: 50));
//        cartBtn.addActionListener(new ActionListener() {
//            @Override
//            public void actionPerformed(ActionEvent e) {
//                new CartScreen(cart);
//            }
//        });
//
        header.add(comp: Box.createRigidArea(new Dimension(width: 10, height: 10)));
        header.add(comp: title);
        header.add(comp: Box.createHorizontalGlue());
        header.add(comp: cartBtn);
        header.add(comp: Box.createRigidArea(new Dimension(width: 10, height: 10)));
```

Figure 3.3: Class StoreScreen 3

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

        return header;
    }
//
    JPanel createCenter() {

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


        ArrayList<Media> mediaInStore = store.getItemsInStore();
        for (int i = 0; i < mediaInStore.size(); i++) {
            MediaStore cell = new MediaStore(media: mediaInStore.get(index: i), cart);
            center.add(comp: cell);
        }

        return center;
    }
```

Figure 3.4: Class StoreScreen 4

3.2 Create class MediaStore

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

import javax.naming.LimitExceededException;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import src.hust.soict.hedspi.aims.Cart.Cart;
import src.hust.soict.hedspi.aims.media.*;

public class MediaStore extends JPanel {
        public MediaStore(Media media, Cart cart) {

            this.setLayout(new BoxLayout(target: this, axis: BoxLayout.Y_AXIS));

            JLabel title = new JLabel(text: media.getTitle());
            title.setFont(new Font(name: title.getFont().getName(), style: Font.PLAIN, size: 20));
            title.setAlignmentX(alignmentX: CENTER_ALIGNMENT);

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

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

//              JButton addToCartButton = new JButton("Add to cart");
//              addToCartButton.addActionListener(new ActionListener() {
```

Figure 3.7: Class MediaStore 1

```java
        header.add(comp: Box.createRigidArea(new Dimension(width: 10, height: 10)));
        header.add(comp: title);
        header.add(comp: Box.createHorizontalGlue());
        header.add(comp: cartBtn);
        header.add(comp: Box.createRigidArea(new Dimension(width: 10, height: 10)));

        return header;
    }
//
    JPanel createCenter() {

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


        ArrayList<Media> mediaInStore = store.getItemsInStore();
        for (int i = 0; i < mediaInStore.size(); i++) {
            MediaStore cell = new MediaStore(media: mediaInStore.get(index: i), cart);
            center.add(comp: cell);
        }

        return center;
    }
```

Figure 3.8: Class MediaStore 2

3.3 Demo

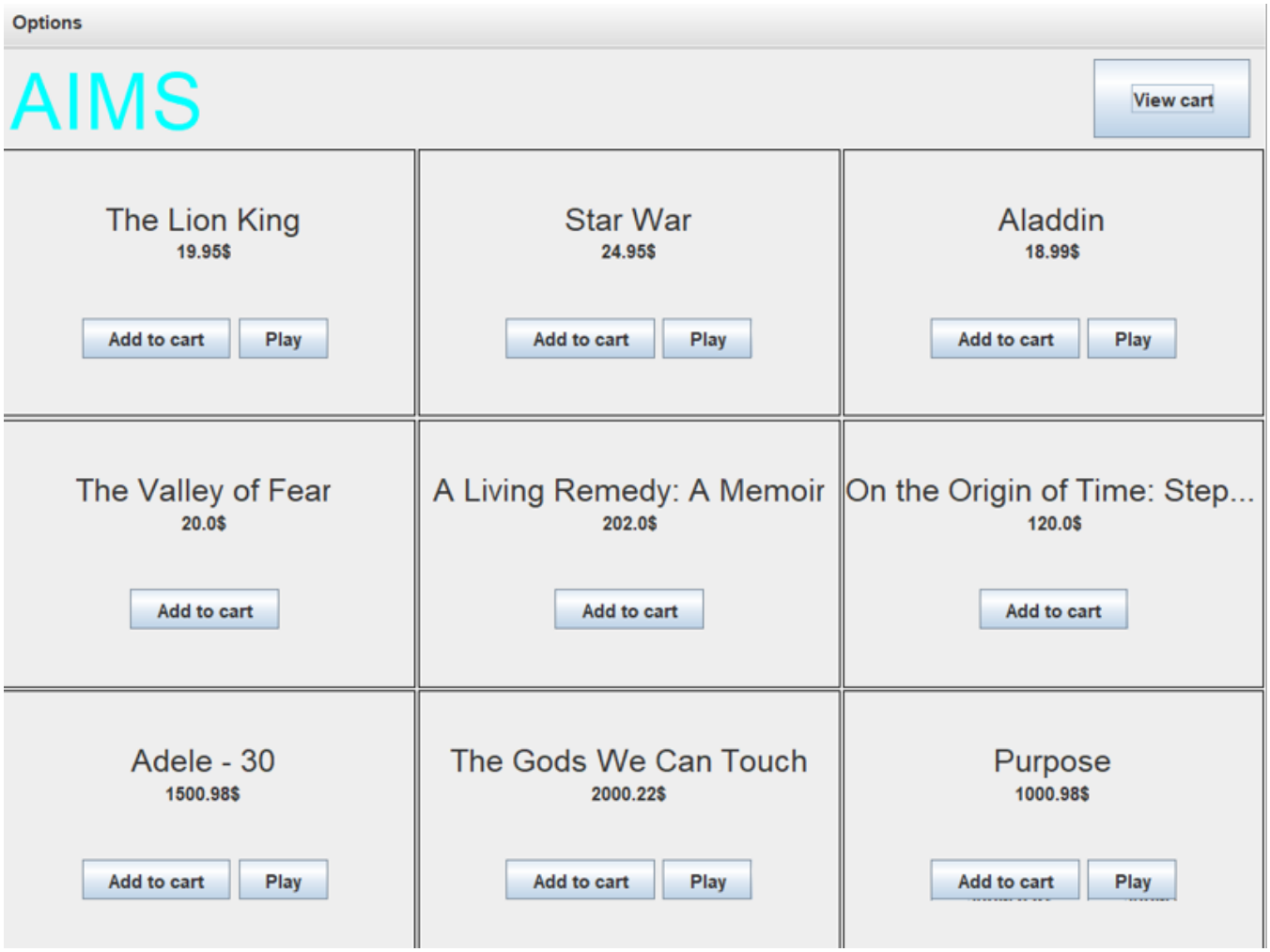


Figure 3.10: StoreScreen

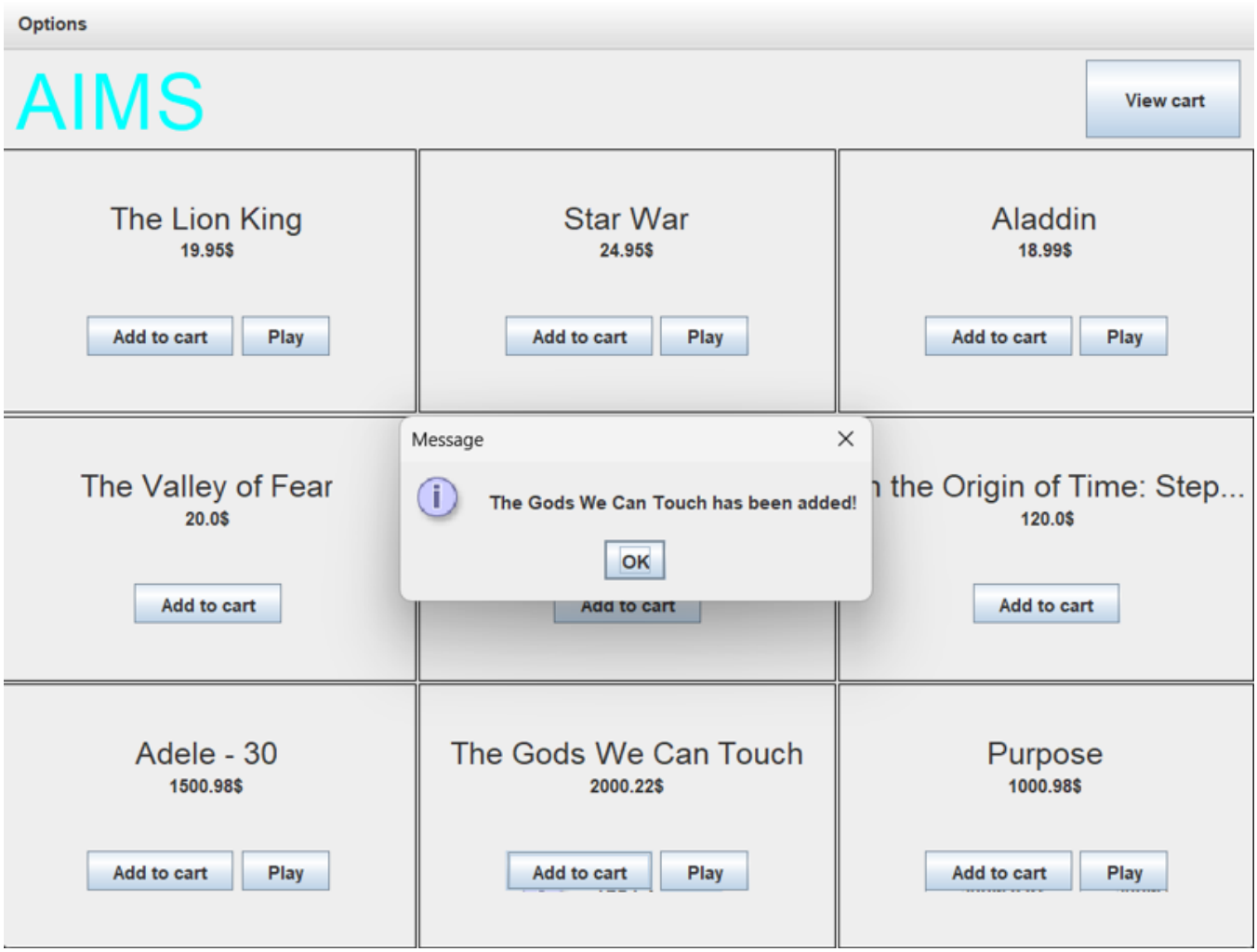


Figure 3.11 Demo Add to cart button

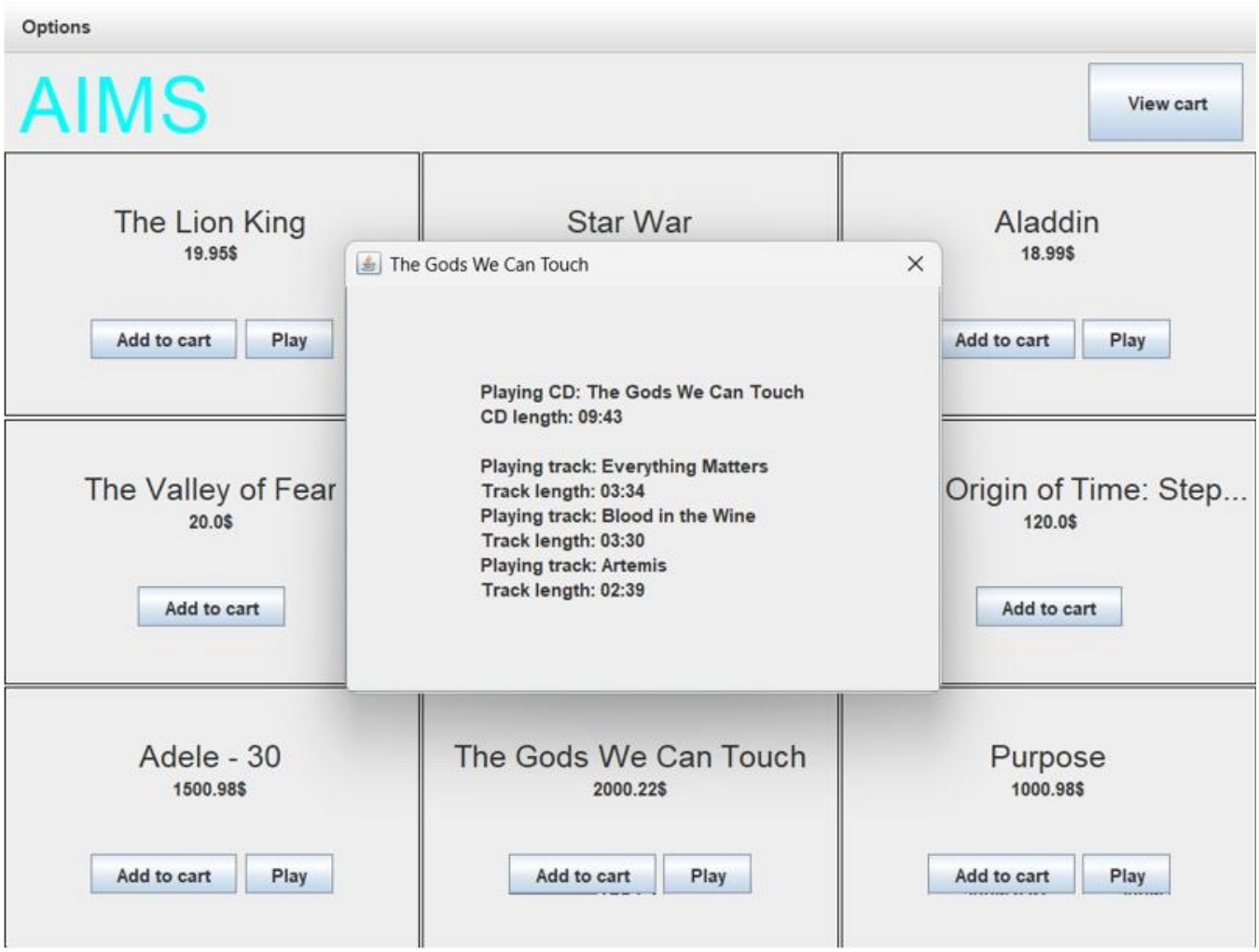


Figure 3.12 Demo Play button

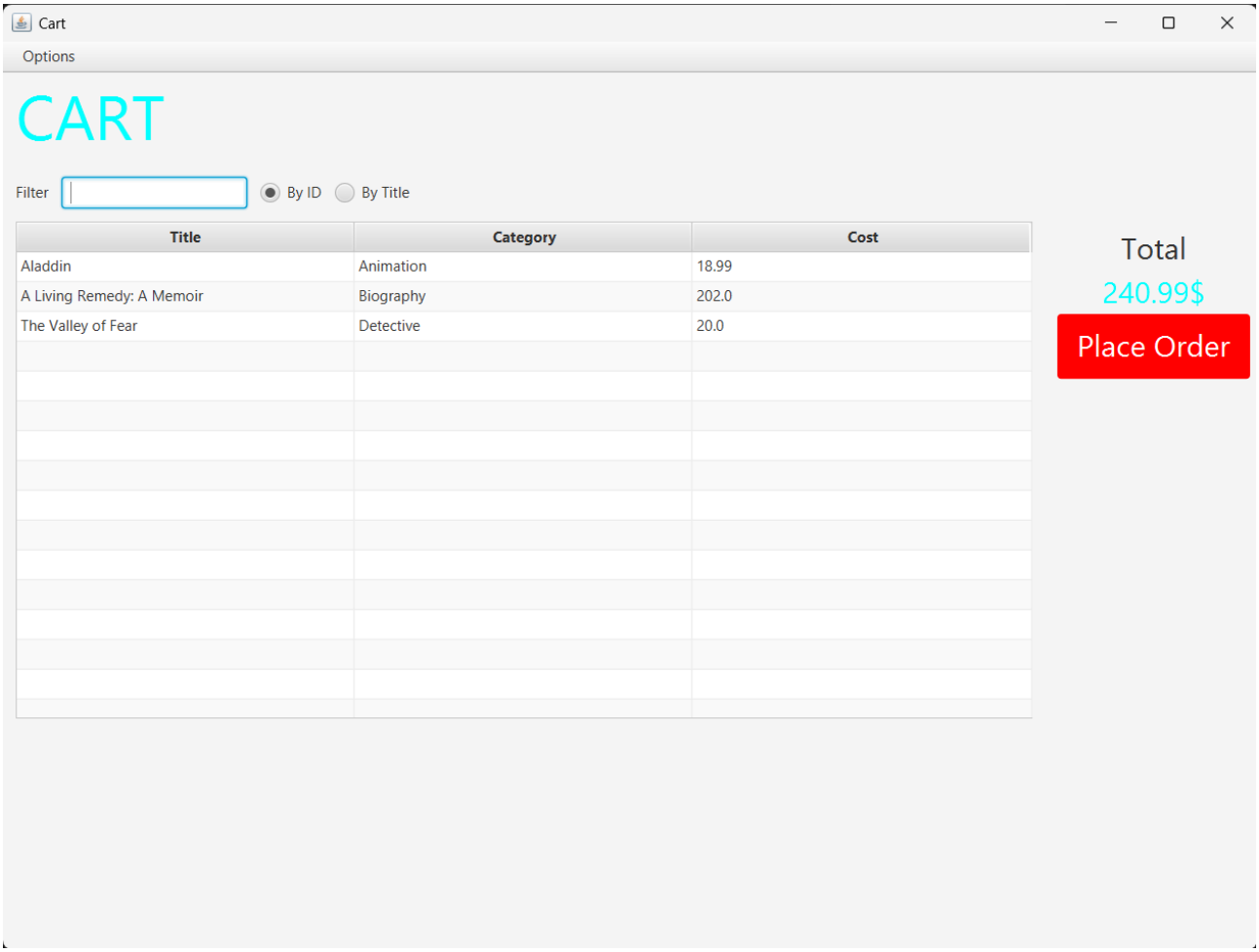


Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

```java
//Do Hoang Dong 225807
package 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(location: getClass()
                                .getResource(name: "javafx/Painter.fxml"));

                Scene scene = new Scene(parent: root);
                stage.setTitle(value: "Painter");
                stage.setScene(value: scene);
                stage.show();
        }
        public static void main(String[] args) {
                launch(args);
        }
}
```

Figure 4.1: Class Painte

4.2 Create Painter.fxml

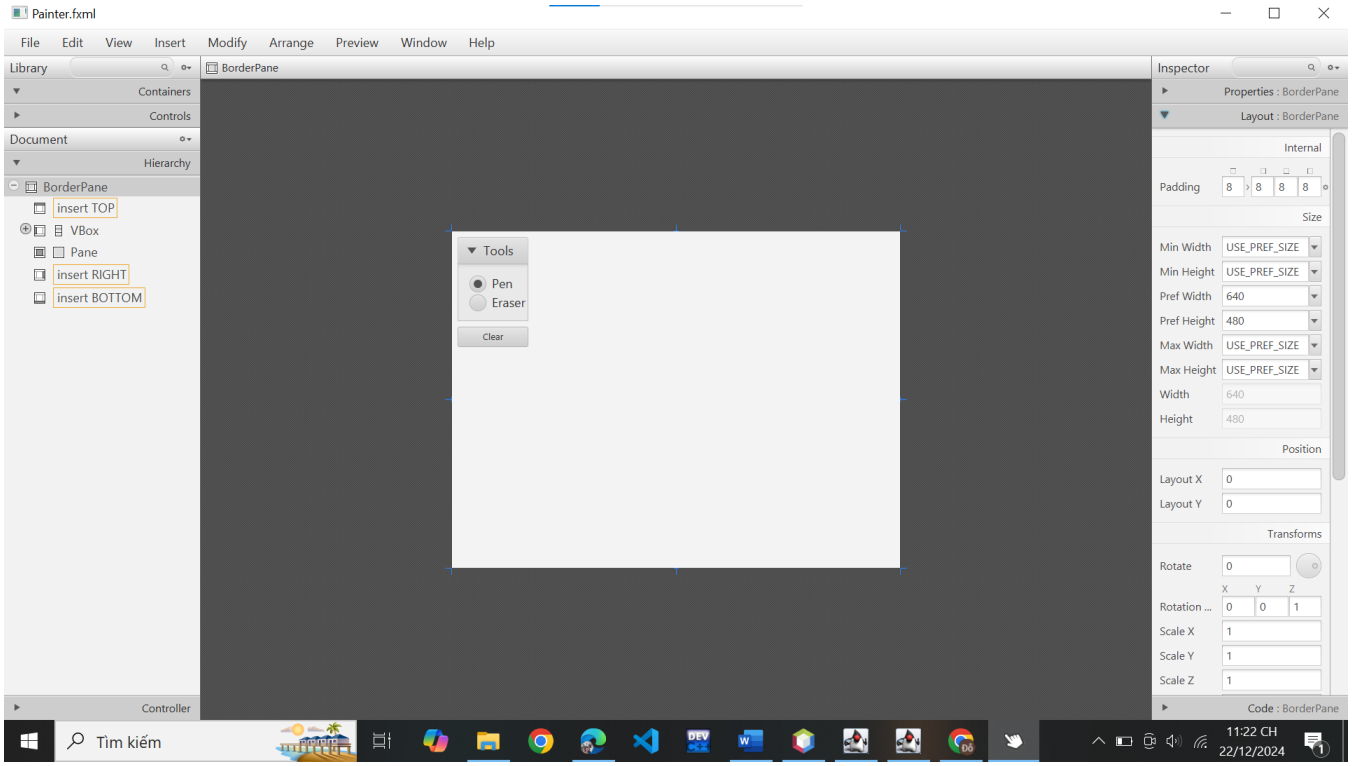


Figure 4.2: Painter.fxml 1

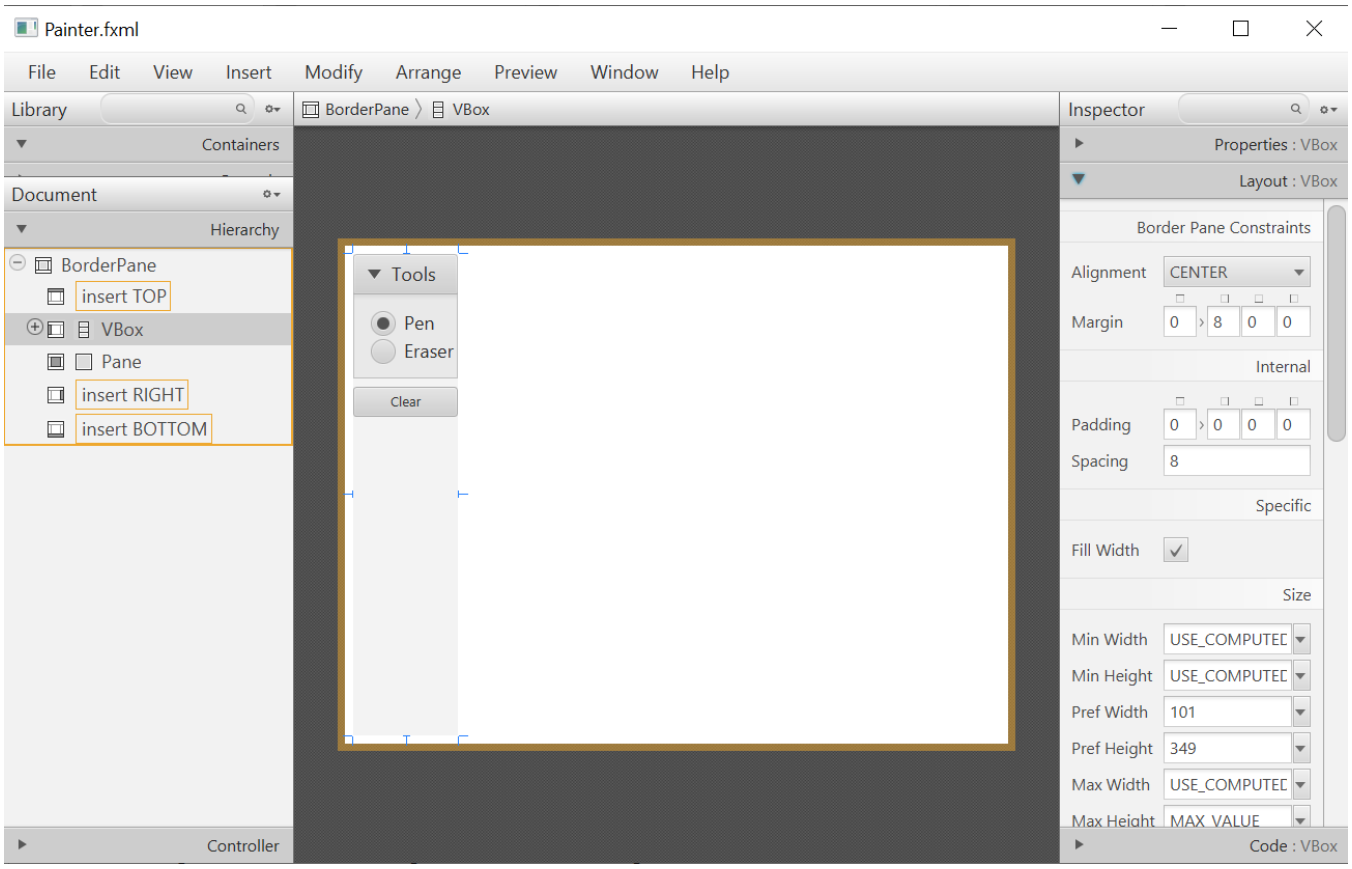


Figure 4.3: Painter.fxml 2

4.3 Create class PainterController

```java
package javafx;

import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Rectangle;

/**
 * FXML Controller class
 *
 * @author HP
 */
public class PainterController implements Initializable {

    @FXML
    private ResourceBundle resources;
    @FXML
```

Figure 4.4: PainterController

```java
public class PainterController implements Initializable {

    @FXML
    private ResourceBundle resources;
    @FXML
    private URL location;
    @FXML
    private VBox drawingAreaPane;
    @FXML
    private RadioButton pen;
    @FXML
    private ToggleGroup Tools;
    @FXML
    private RadioButton eraser;

    /**
     * Initializes the controller class.
     */
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        assert drawingAreaPane != null :
       "fx:id=\"drawingAreaPane\" was not injected: check your FXML file 'Painter.fxml'.";
    }

    @FXML
```

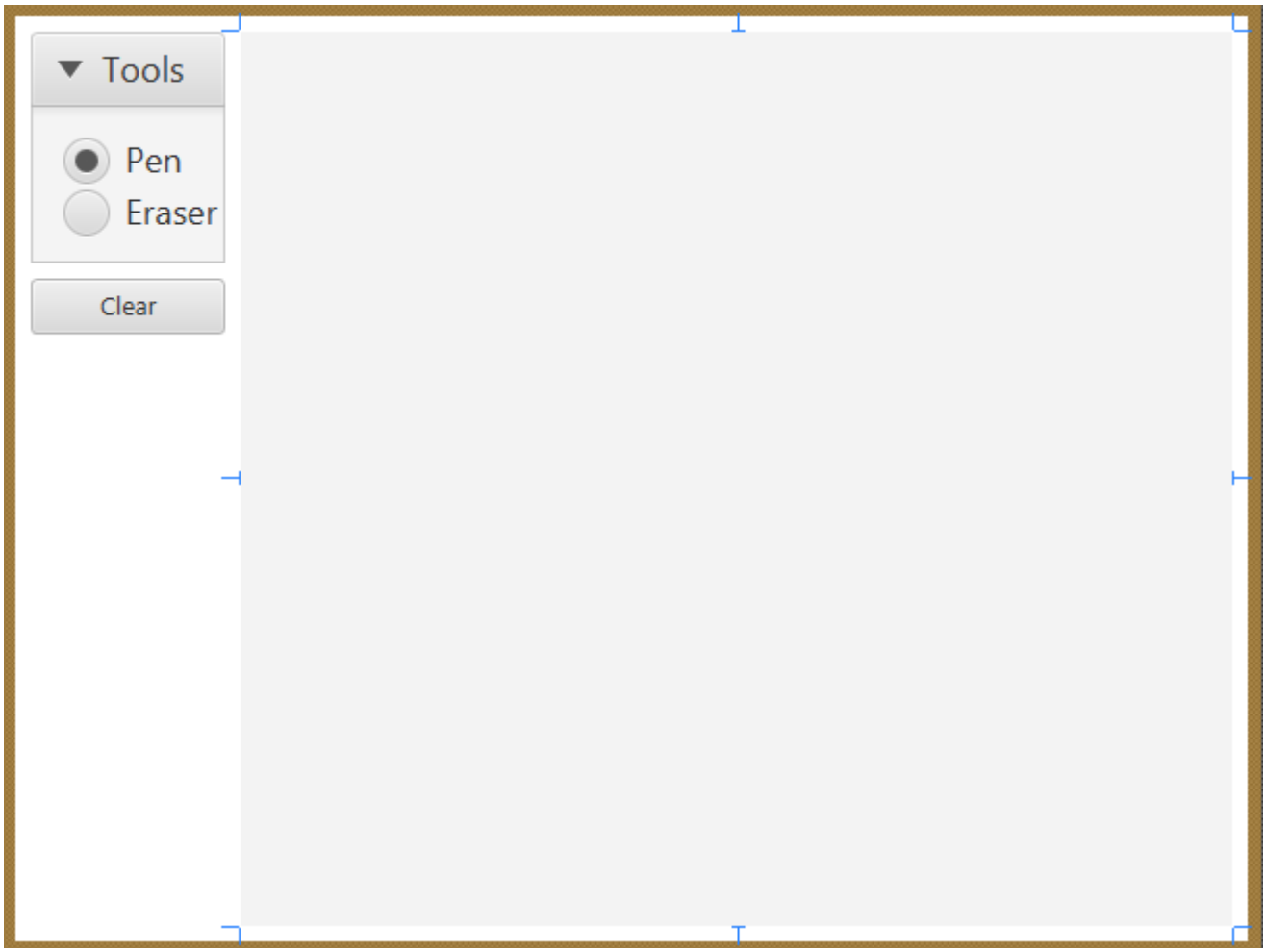


Figure 4.5:

5 View Cart Screen

Figure 5.3: Cart.fxml 3

5.1 Create class CartScreen

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

import javax.swing.JFrame;
import hust.soict.hedspi.aims.Screen.controller.CartScreenController;
import javafx.application.Platform;
import javafx.embed.swing.JFXPanel;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import src.hust.soict.hedspi.aims.Cart.Cart;

public class CartScreen extends JFrame {

    private static Cart cart;

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

    public CartScreen(Cart cart) {

        super();

        CartScreen.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
```

```java
        CartScreen.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(comp: fxPanel);

        this.setTitle(title: "Cart");
        this.setSize(width: 1024, height: 768);
        this.setVisible(b: true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(url:getClass().getResource(name: "cart.fxml"));

                    CartScreenController controller = new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(parent: root));
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

Create class CartScreenController

```java
package hust.soict.hedspi.aims.Screen.controller;

import src.hust.soict.hedspi.aims.Cart.Cart;
import src.hust.soict.hedspi.aims.exception.PlayerException;
import src.hust.soict.hedspi.aims.media.Media;
import src.hust.soict.hedspi.aims.media.Playable;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.transformation.FilteredList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;

public class CartScreenController {
    private Cart cart;

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;

    @FXML
    private TableColumn<Media, Float> colMediaCost;
```

Figure 5.5: CartScreenController 1

```java
    private TableColumn<Media, String> colMediaTitle;

    @FXML
    private TableColumn<Media, String> colMediacategory;

    @FXML
    private TableView<Media> tblMedia;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private RadioButton radioBtnFilterId;

    @FXML
    private RadioButton radioBtnFilterTitle;

    @FXML
    private Label costLabel;

    @FXML
    private TextField tfFilter;

    @FXML
    private Button placeOrder;
```

Figure 5.6: CartScreenController 2

```java

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

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

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

```

```java
        new PropertyValueFactory<Media, Float>(string: "cost")
    );
    //tblMedia.setItems(this.cart.getItemsOrdered());

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

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

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
        new ChangeListener<Media>() {

            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newV
                if (newValue != null) {
                    updateButtonBar(media: newValue);
                }
            }

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

```
                btnPlay.setVisible(value: true);
            } else {
                btnPlay.setVisible(value: false);
            }
        }
    }
);

tfFilter.textProperty().addListener(
    new ChangeListener<String>() {
    @Override
    public void changed(ObservableValue<? extends String> observable, String oldValue, String newVal
        showFilteredMedia(keyword:newValue);
    }
    private void showFilteredMedia(String keyword) {
        FilteredList<Media> filteredList = new FilteredList<>(ol: cart.getItemsOrdered());

        if (!keyword.isEmpty() && radioBtnFilterId.isSelected()) {
            filteredList.setPredicate(media -> {
                String idString = String.valueOf(i: media.getId());
                return idString.equals(anObject: keyword);
            });
        } else if (!keyword.isEmpty() &&  radioBtnFilterTitle.isSelected()) {
            filteredList.setPredicate(media -> {
                String title = media.getTitle().toLowerCase();
                return title.contains(s: keyword.toLowerCase());
```

```
    @Override
    public void changed(ObservableValue<? extends String> observable, String oldValue, String newVa
        showFilteredMedia(keyword:newValue);
    }
    private void showFilteredMedia(String keyword) {
        FilteredList<Media> filteredList = new FilteredList<>(ol: cart.getItemsOrdered());

        if (!keyword.isEmpty() && radioBtnFilterId.isSelected()) {
            filteredList.setPredicate(media -> {
                String idString = String.valueOf(i: media.getId());
                return idString.equals(anObject: keyword);
            });
        } else if (!keyword.isEmpty() &&  radioBtnFilterTitle.isSelected()) {
            filteredList.setPredicate(media -> {
                String title = media.getTitle().toLowerCase();
                return title.contains(s: keyword.toLowerCase());
            });
        } else {
            filteredList.setPredicate(predicate: null);
        }
        tblMedia.setItems(value: filteredList);
    }
});

}
```

5.2 Demo

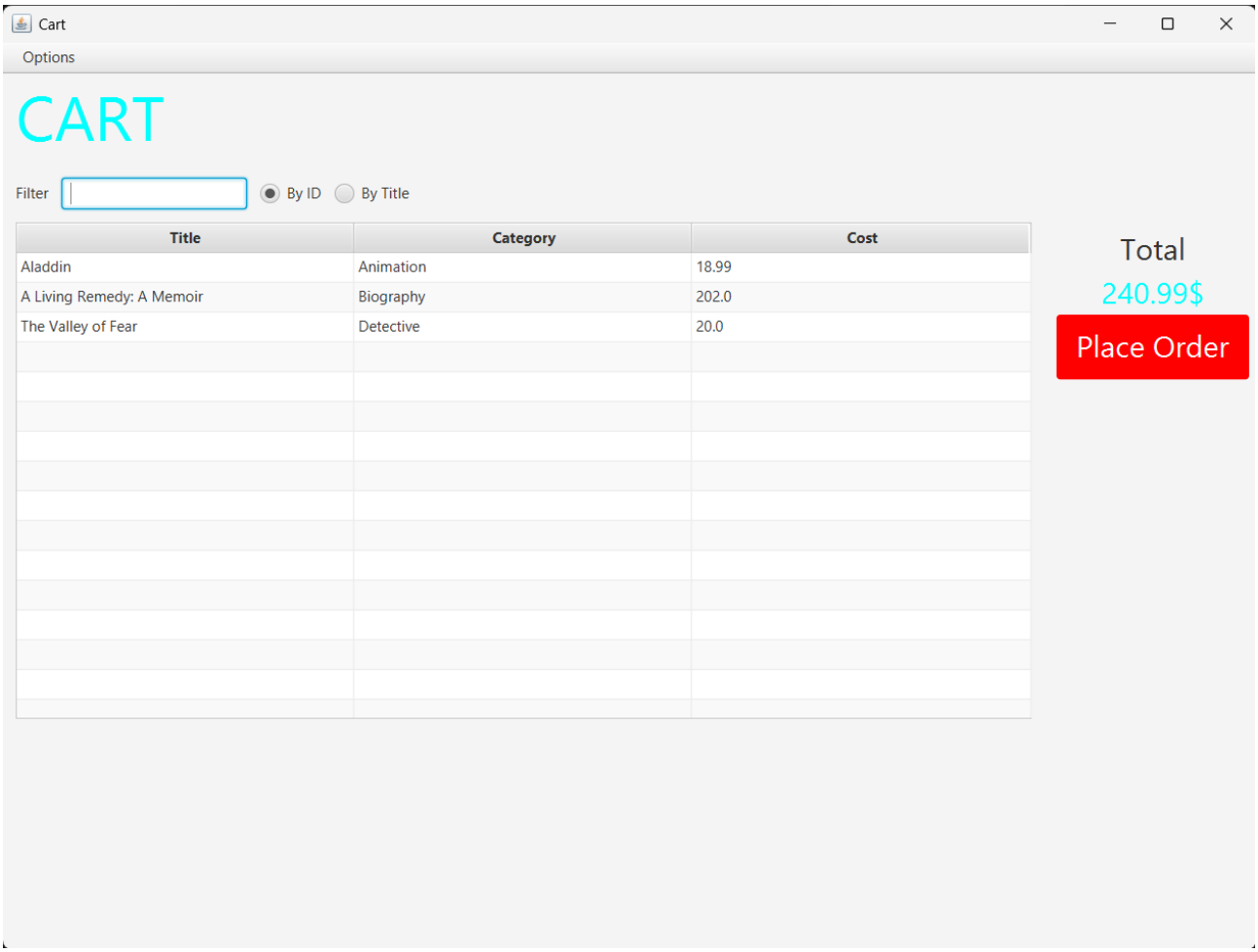


Figure 5.7: Demo CartScreen

6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController

```java
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(media: newValue);
                }
            }

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

Figure 6.1: CartScreenController 1

```java
tfFilter.textProperty().addListener(
    new ChangeListener<String>() {
    @Override
    public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
        showFilteredMedia(keyword:newValue);
    }
    private void showFilteredMedia(String keyword) {
        FilteredList<Media> filteredList = new FilteredList<>(ol: cart.getItemsOrdered());

        if (!keyword.isEmpty() && radioBtnFilterId.isSelected()) {
            filteredList.setPredicate(media -> {
                String idString = String.valueOf(i: media.getId());
                return idString.equals(anObject: keyword);
            });
        } else if (!keyword.isEmpty() &&  radioBtnFilterTitle.isSelected()) {
            filteredList.setPredicate(media -> {
                String title = media.getTitle().toLowerCase();
                return title.contains(s: keyword.toLowerCase());
            });
        } else {
            filteredList.setPredicate(predicate: null);
        }
        tblMedia.setItems(value: filteredList);
    }
```

Figure 6.2: CartScreenController 2

6.2 Demo

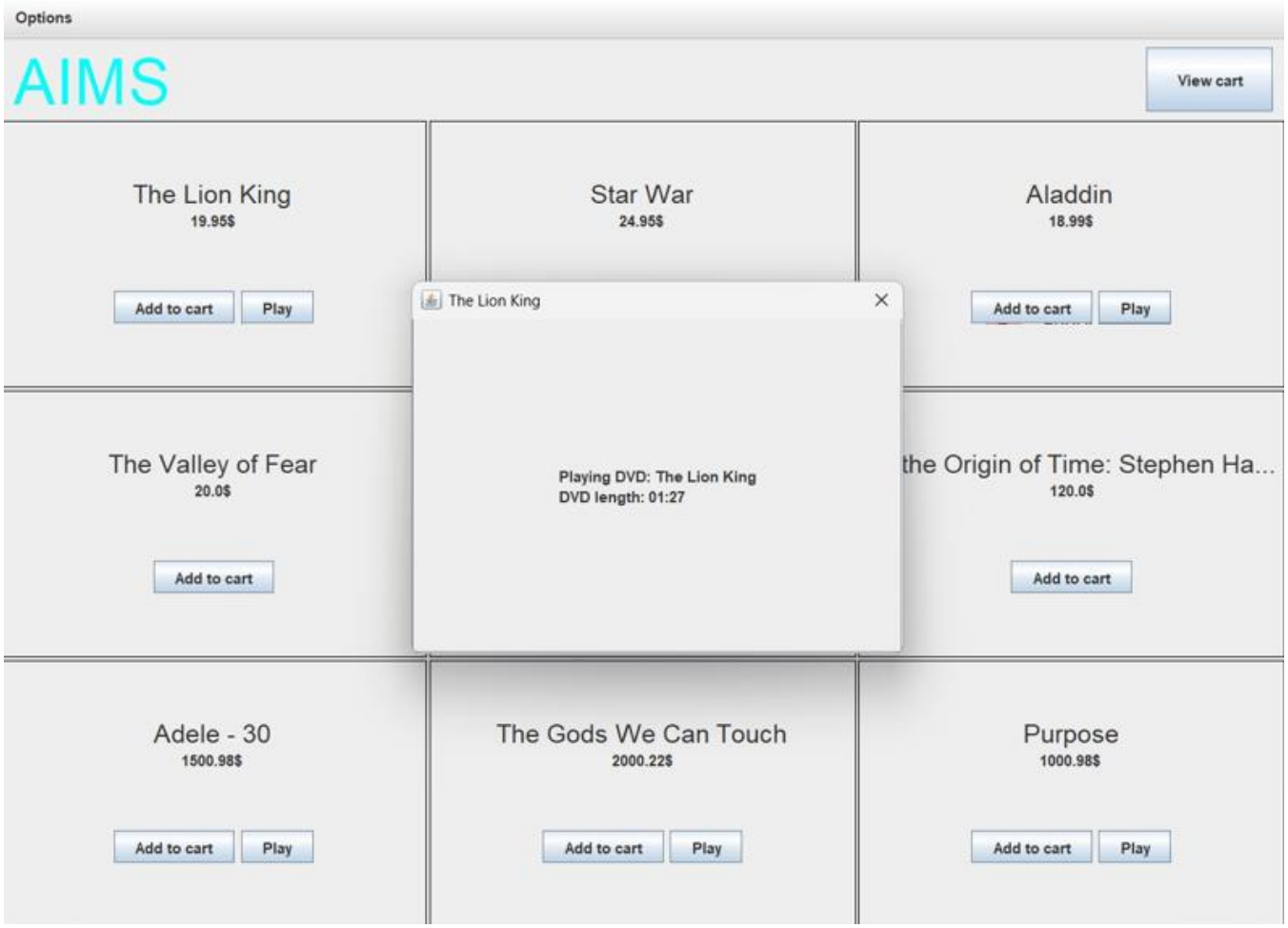


Figure 6.3: Demo media playable

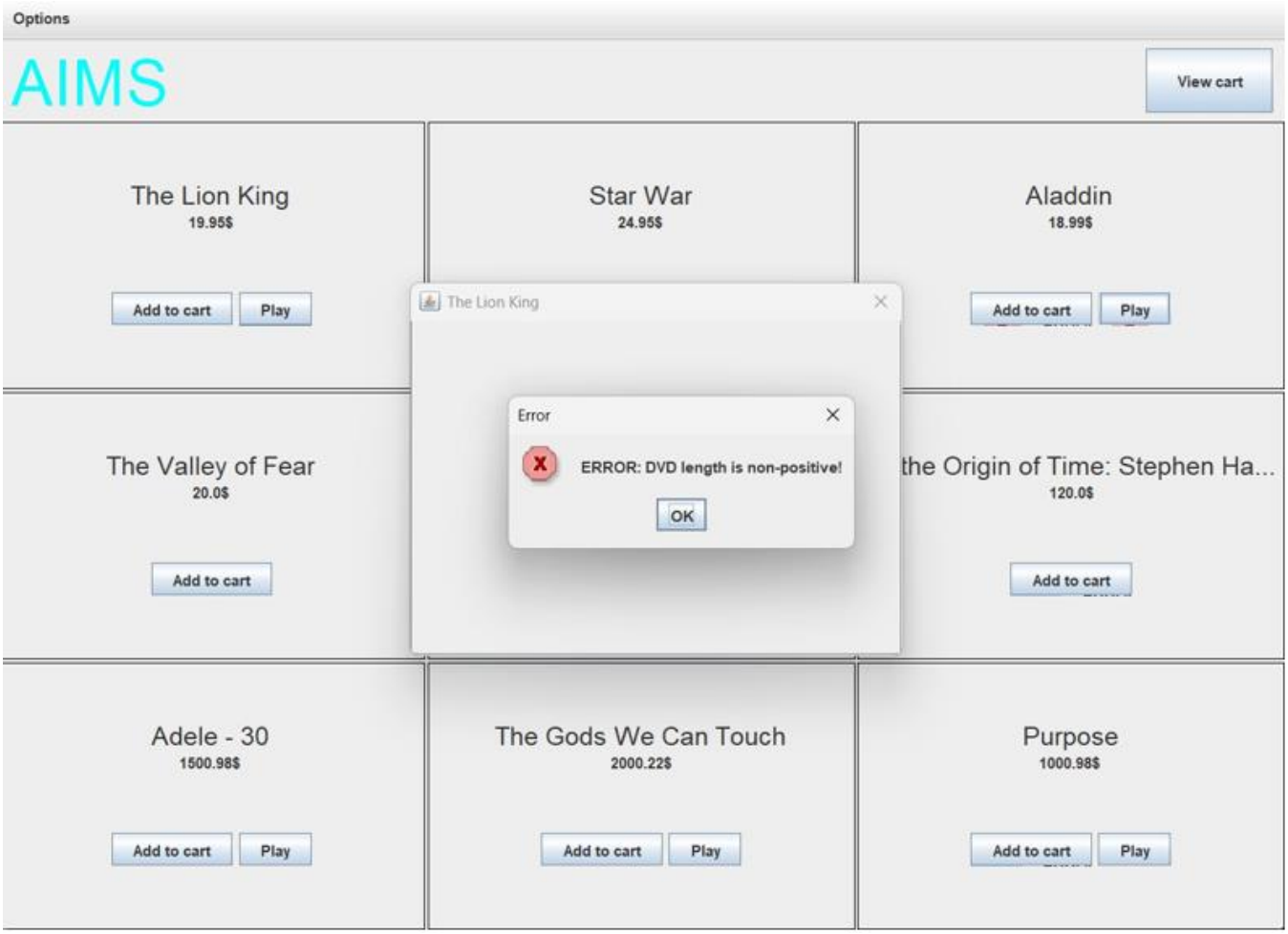


Figure 6.4: Demo media unplayable

7 Deleting a media

7.1 Code

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

Figure 7.1: btnRemovePressed Method

7.2 Demo

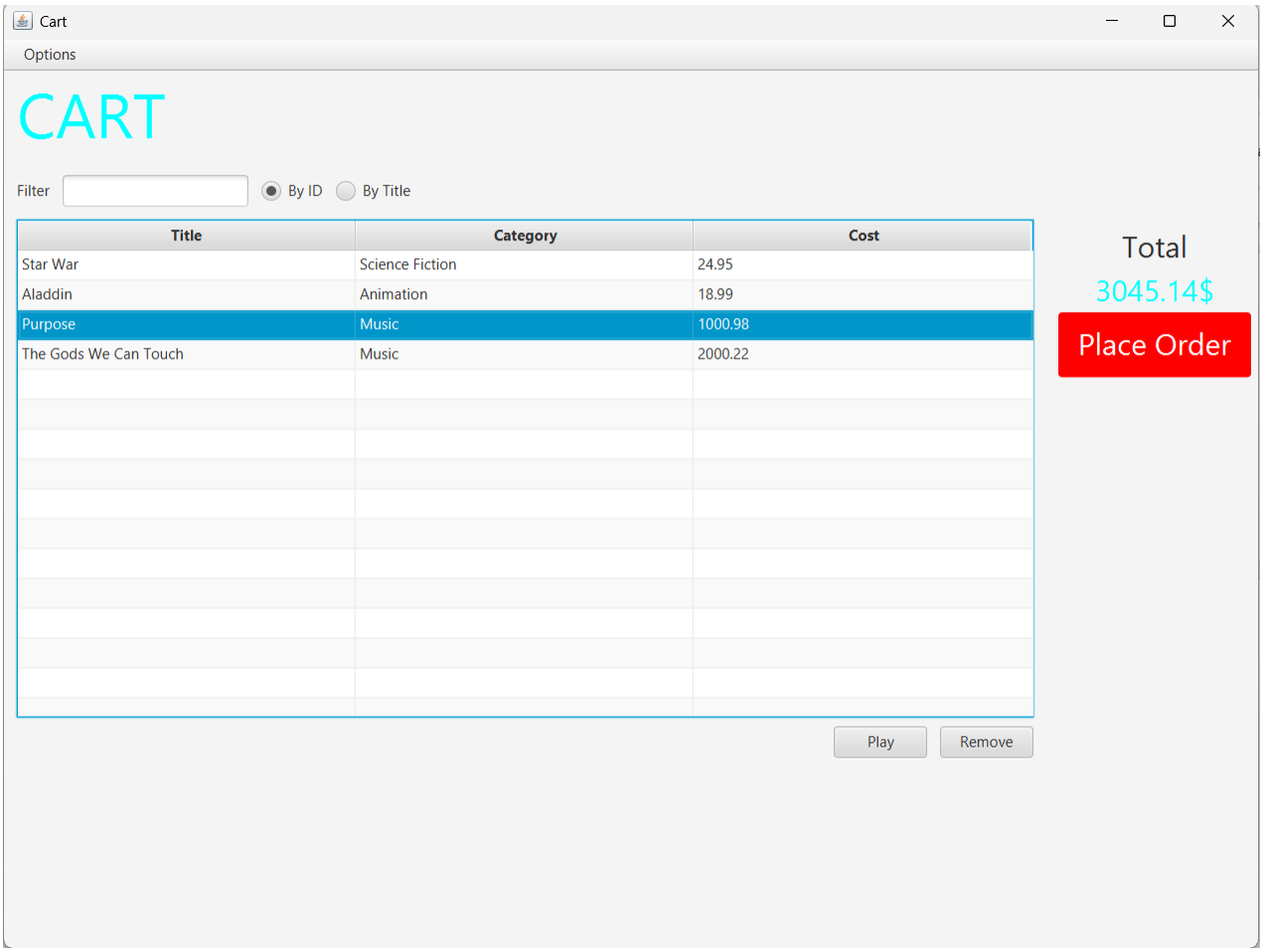


Figure 7.2: button Remove

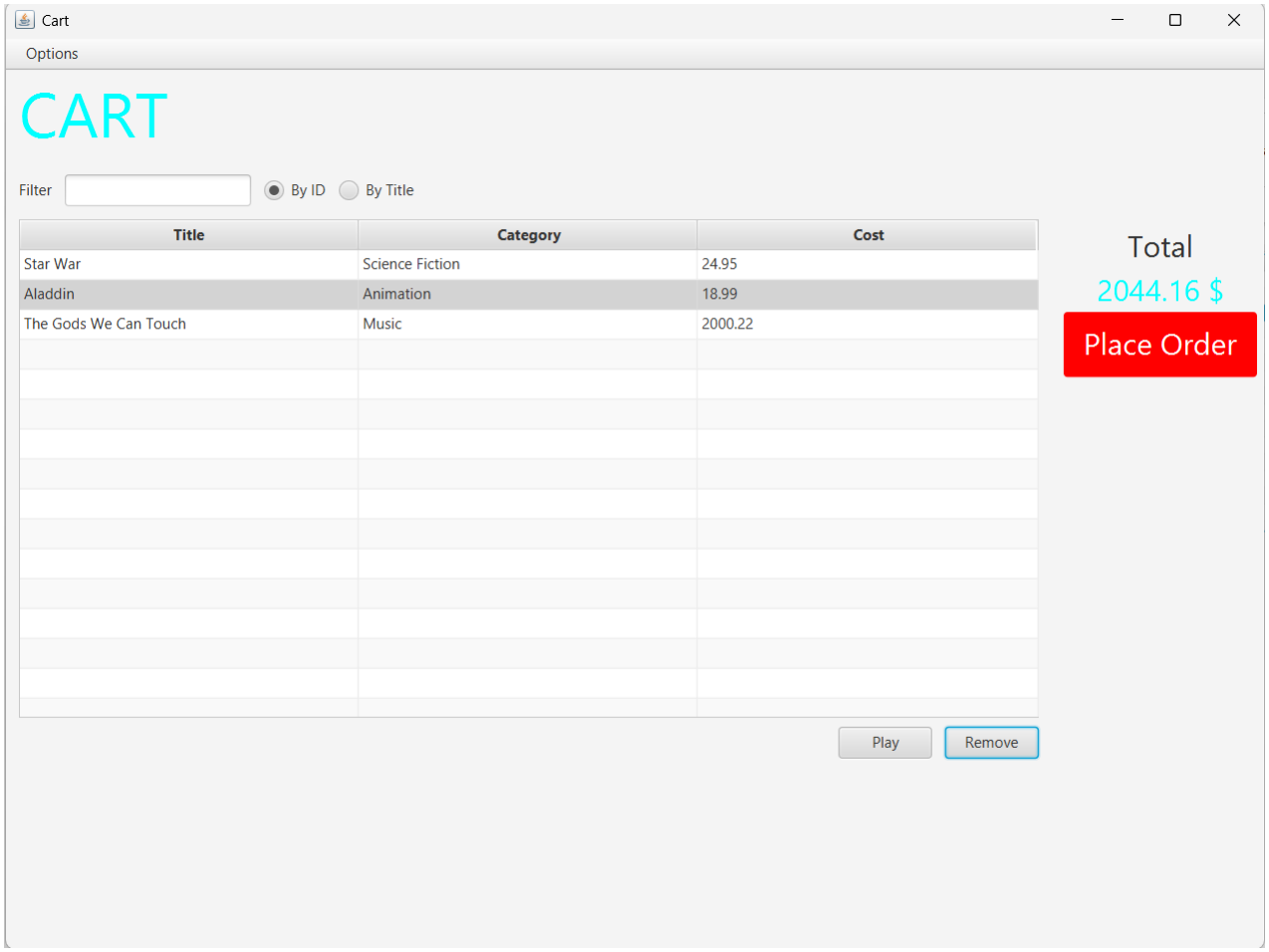


Figure 7.3: button Remove

8 Complete the Aims GUI application

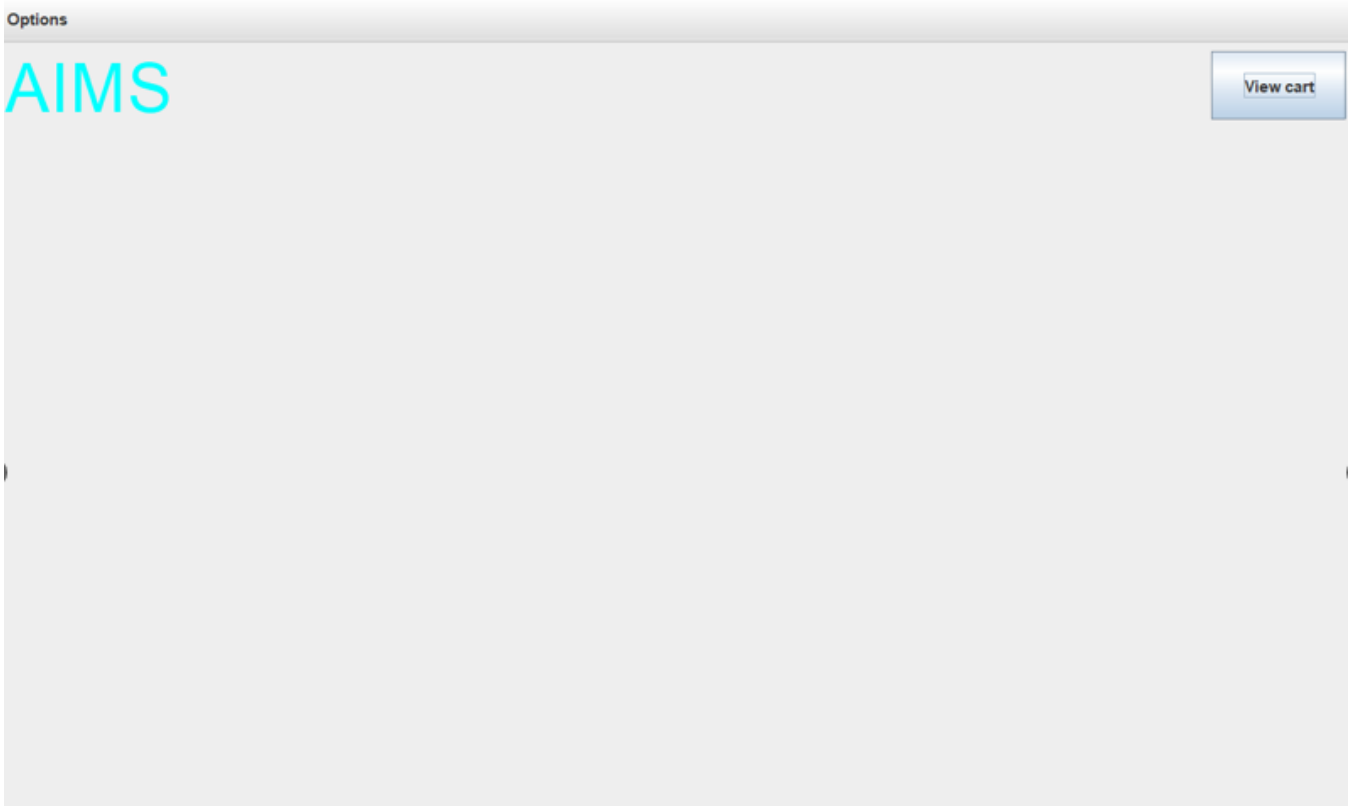


Figure 8.1: Store before add book

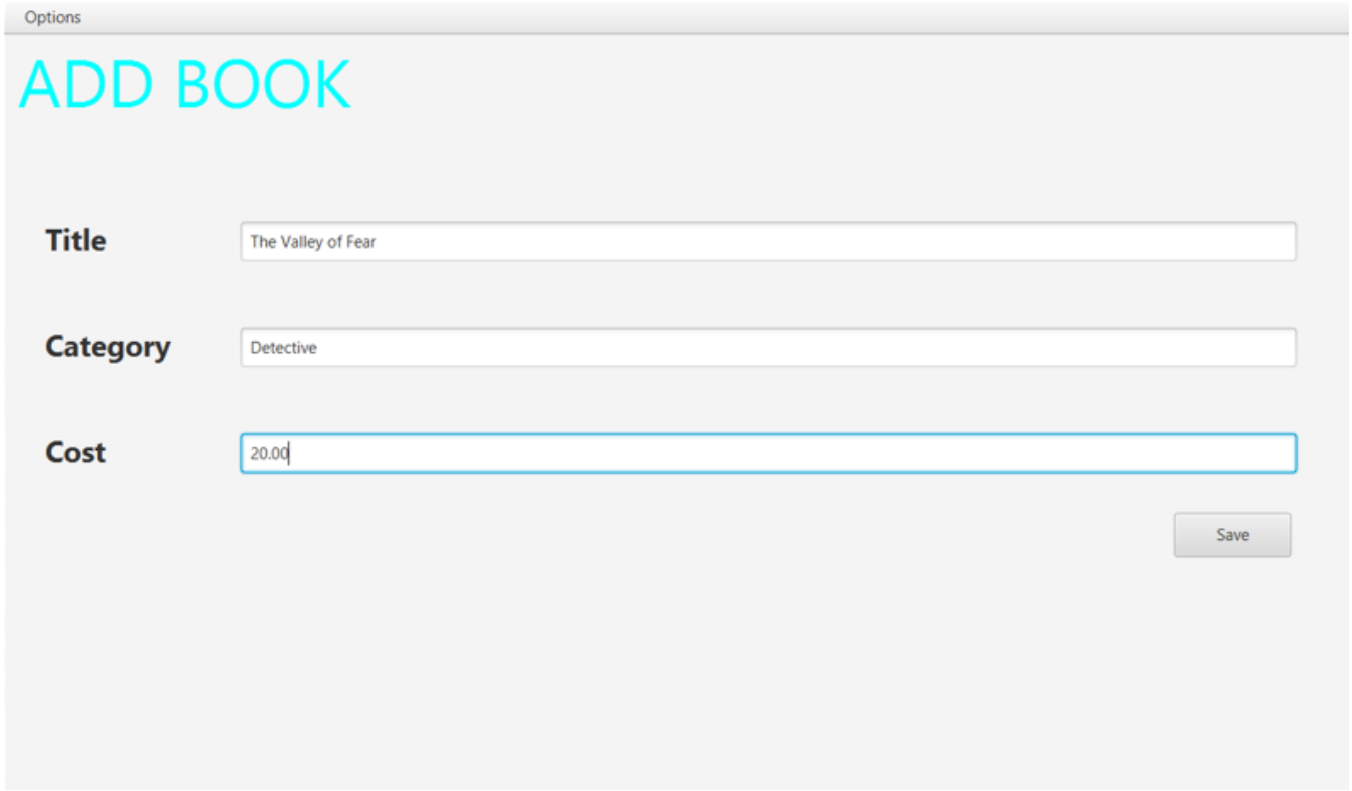


Figure 8.2: Add book

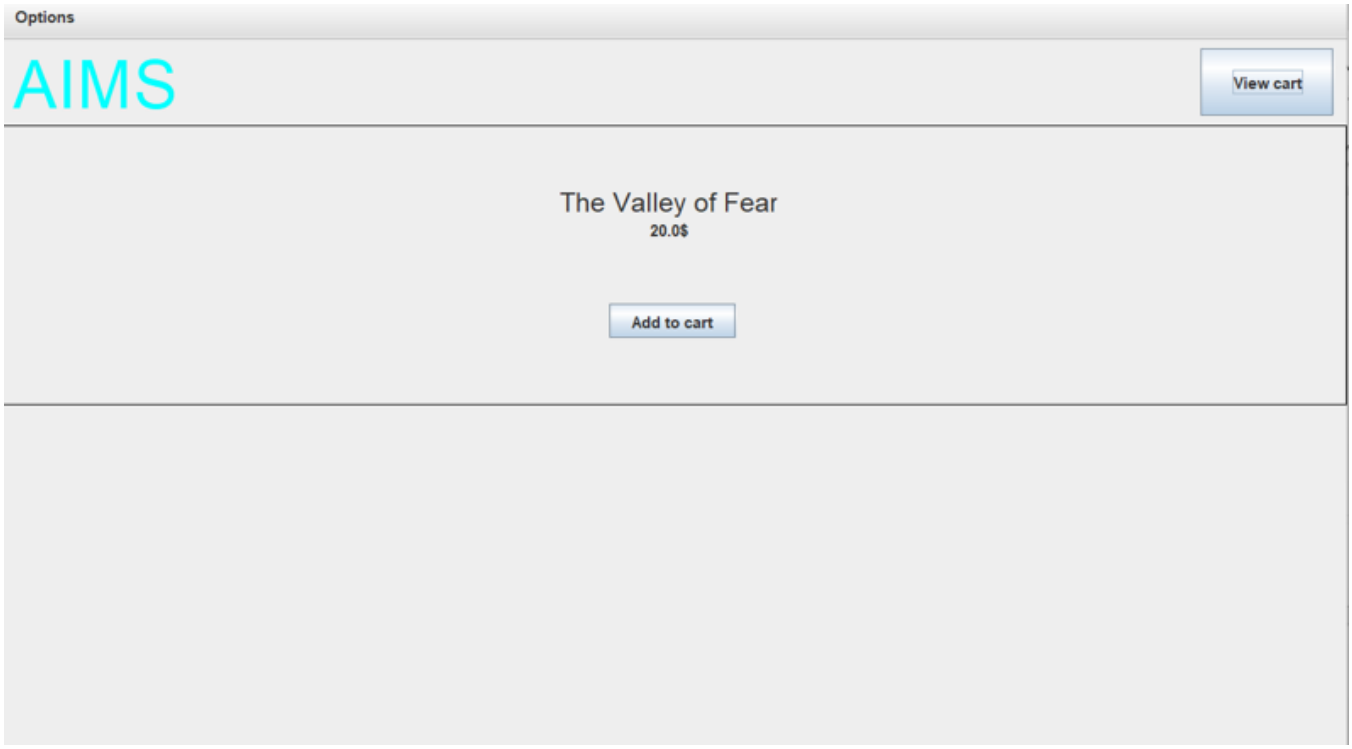


Figure 8.3: Store after add book

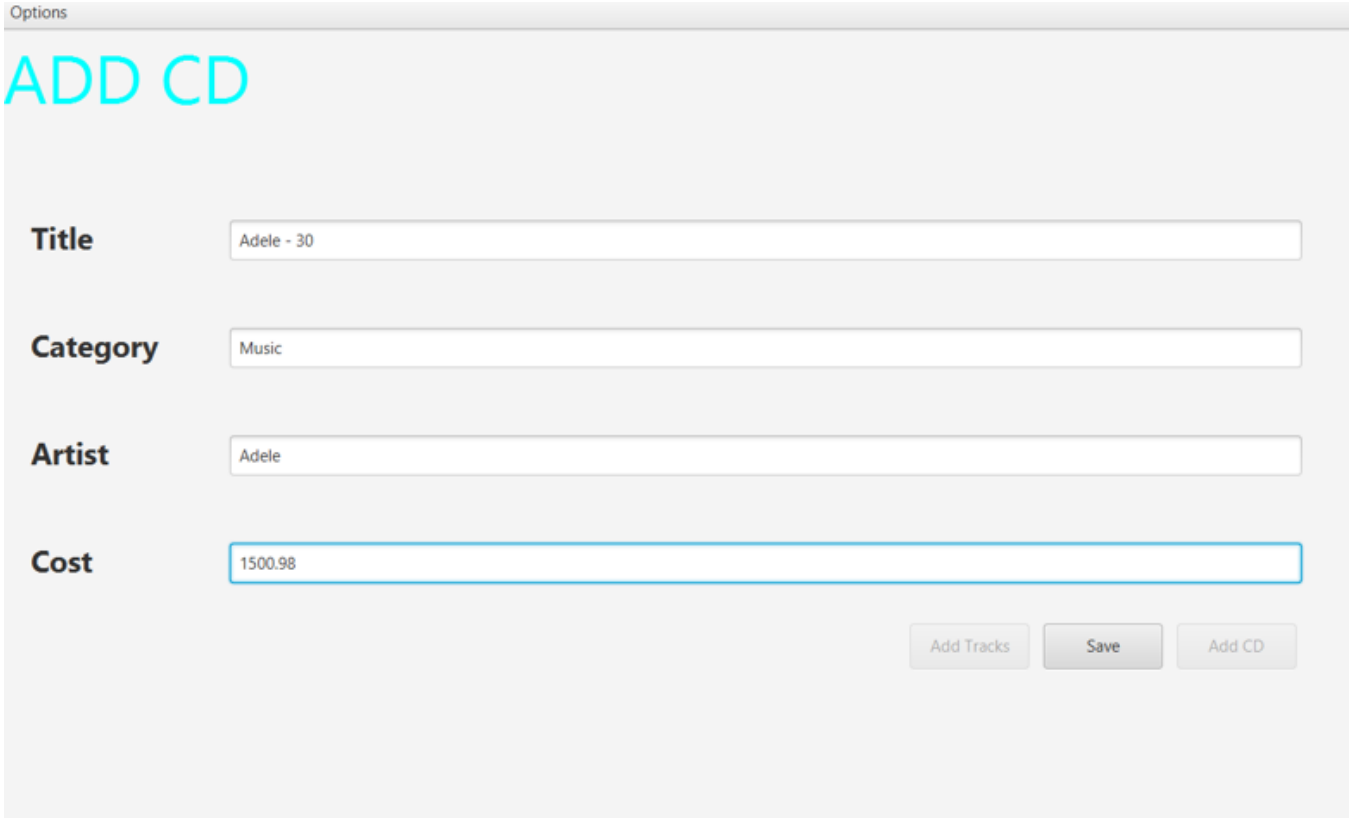
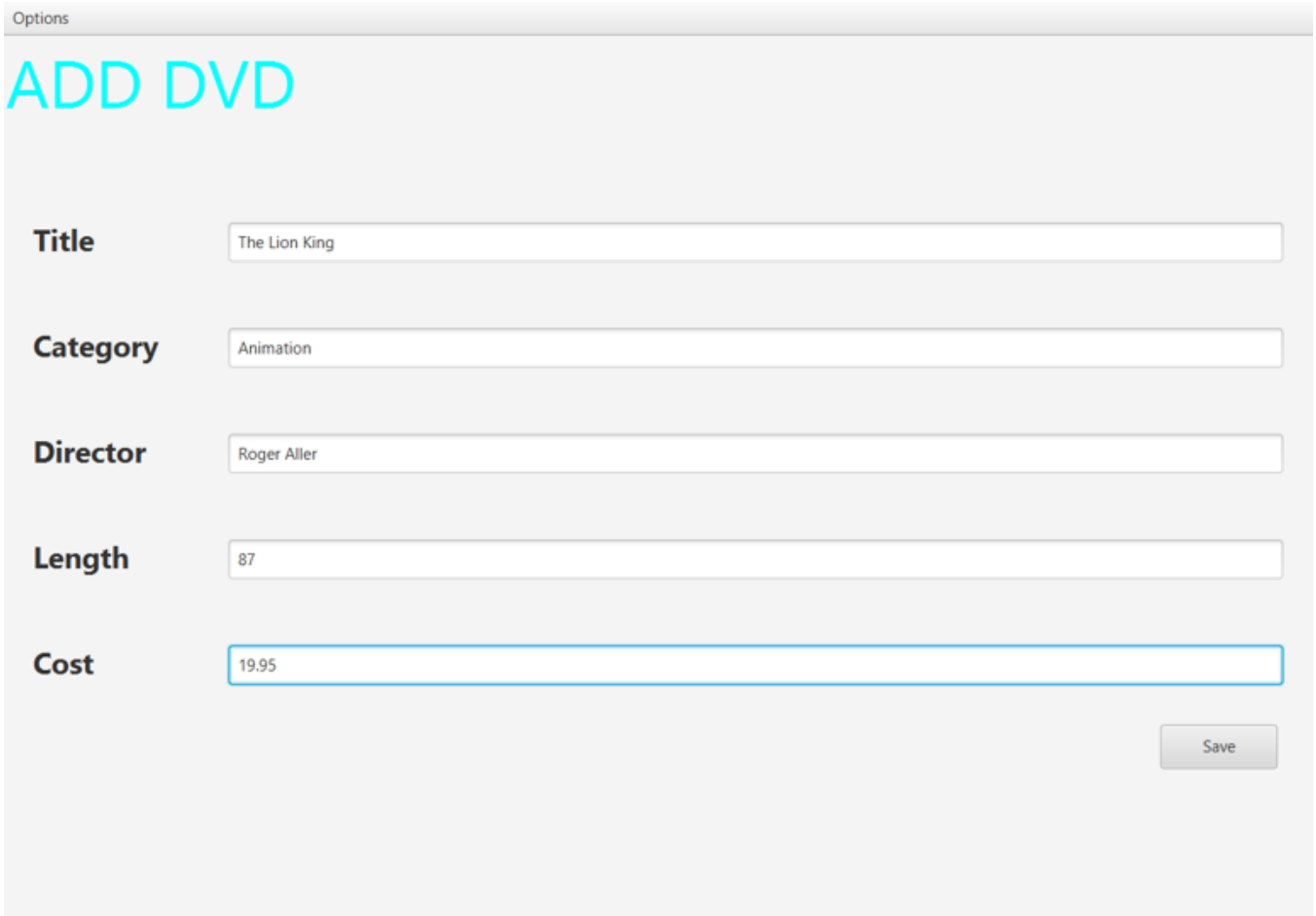


Figure 8.4: Add CD


Options
ADD DVD
Title
The Lion King
Category
Animation
Director
Roger Aller
Length
87
Cost
19.95
Save

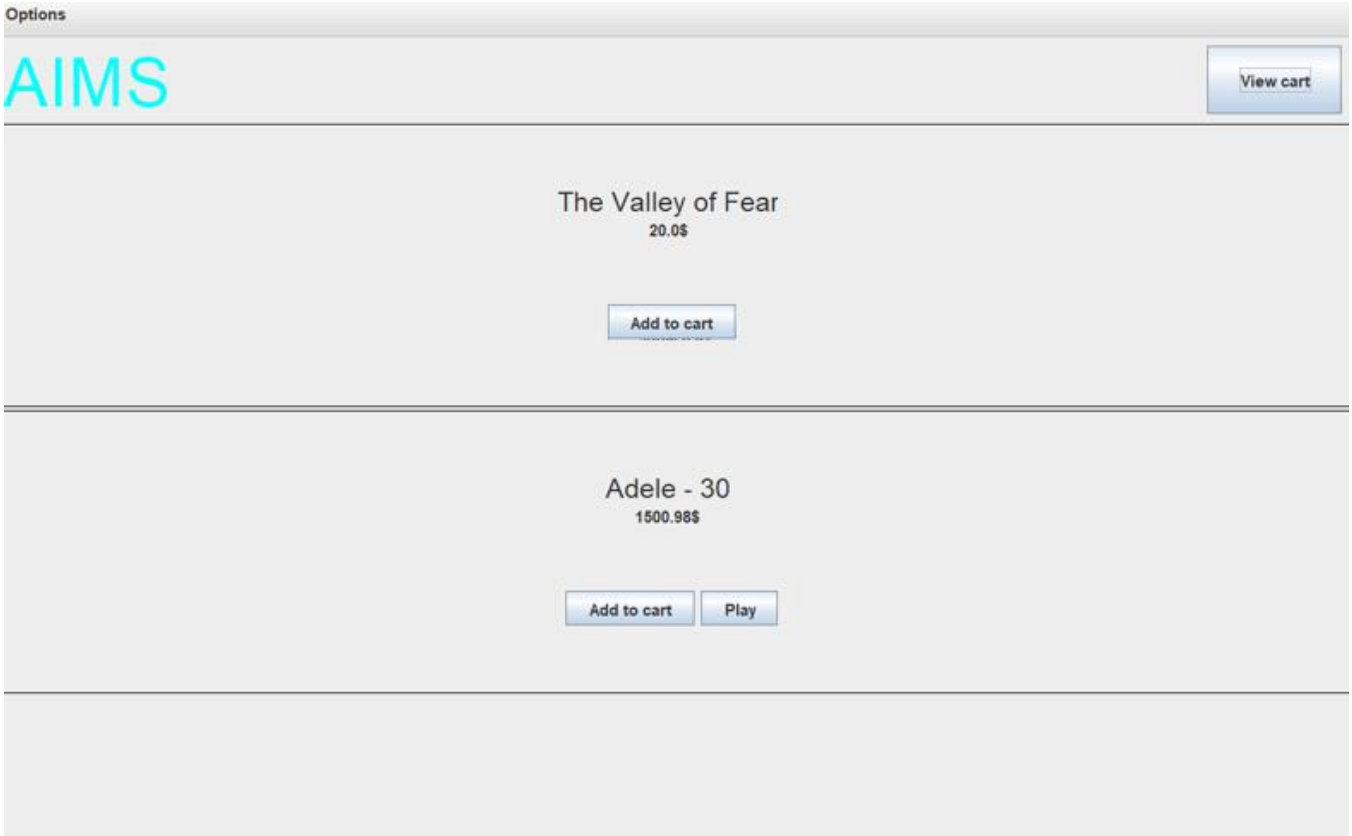


Figure 8.5: Store after add CD

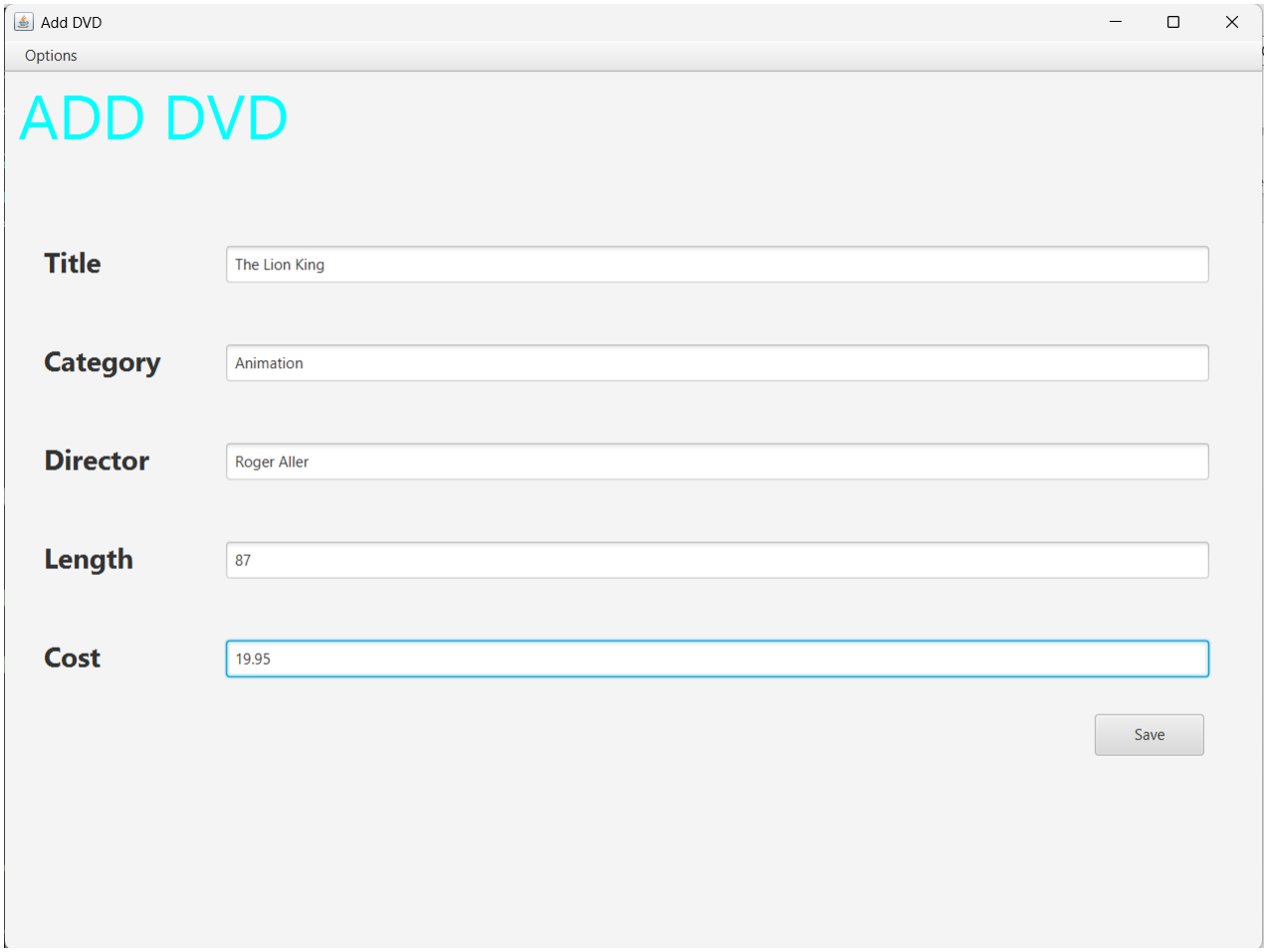


Figure 8.6 Add DVD

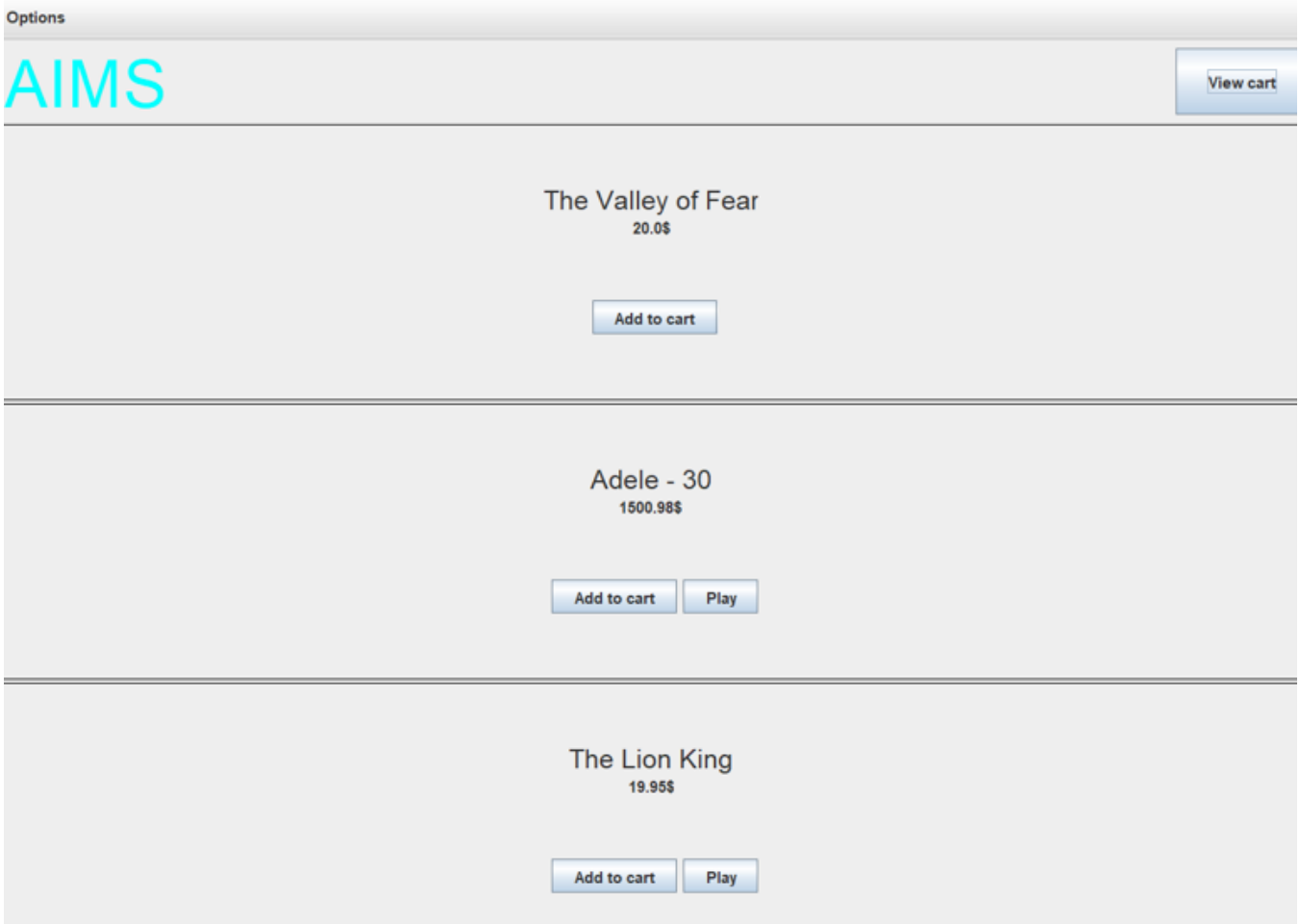


Figure 8.7: Store after add DVD

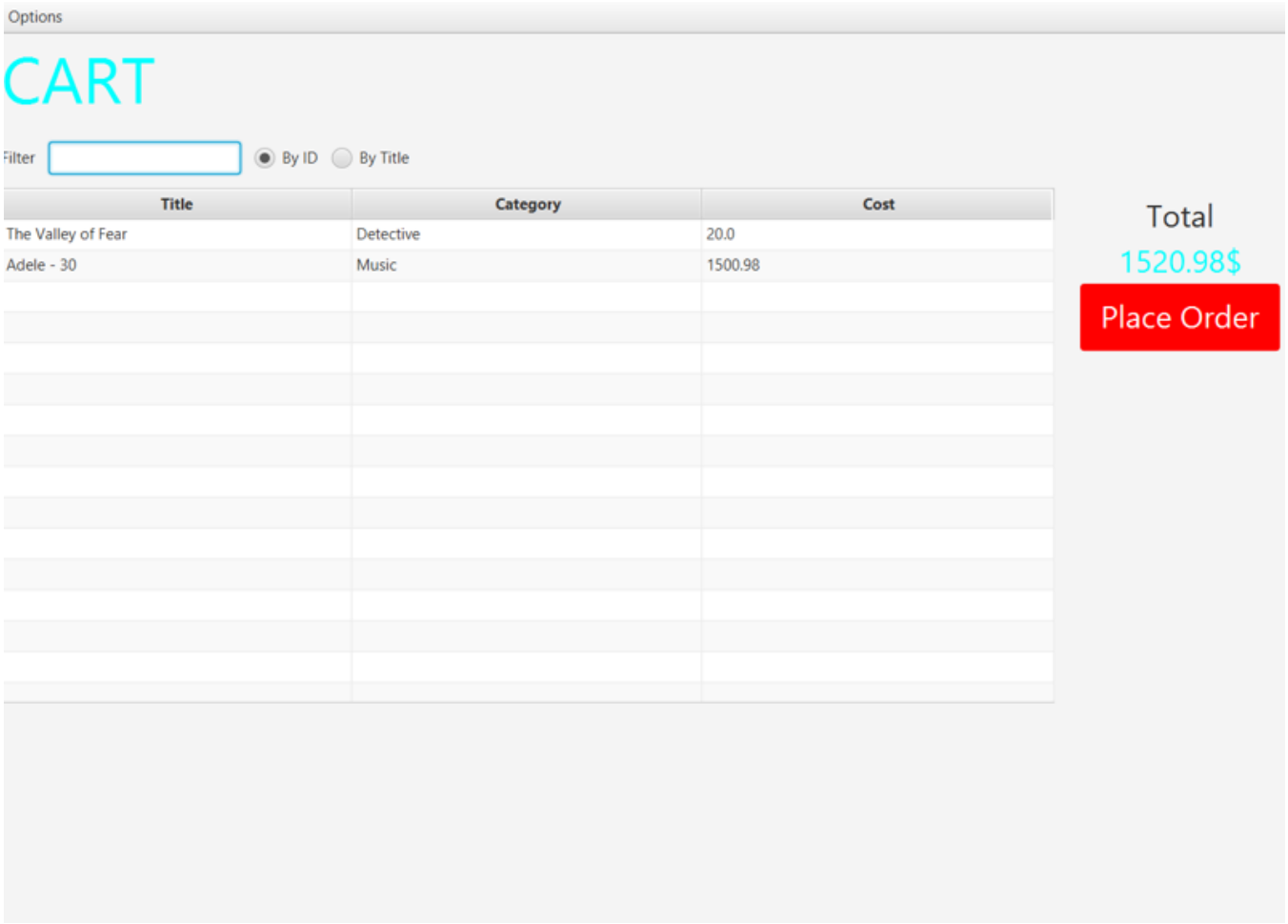
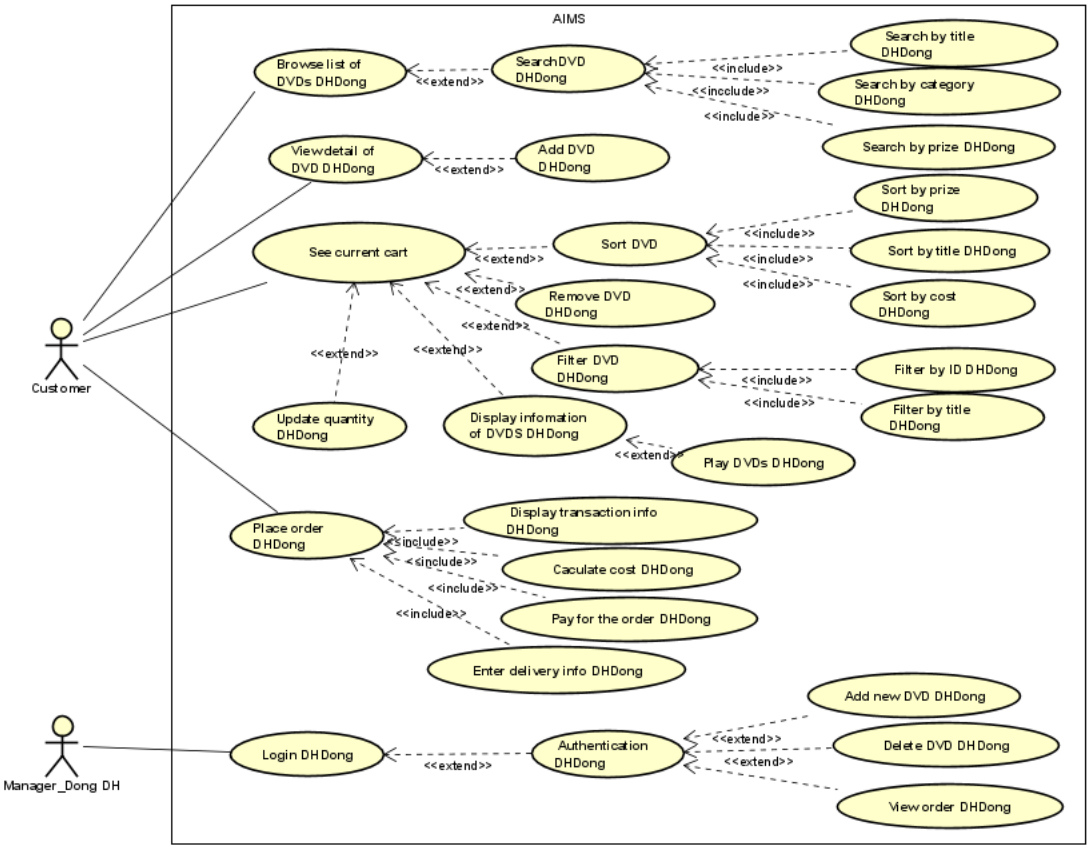


Figure 8.8: Cart

9 Use case Diagram



10. Class Diagram

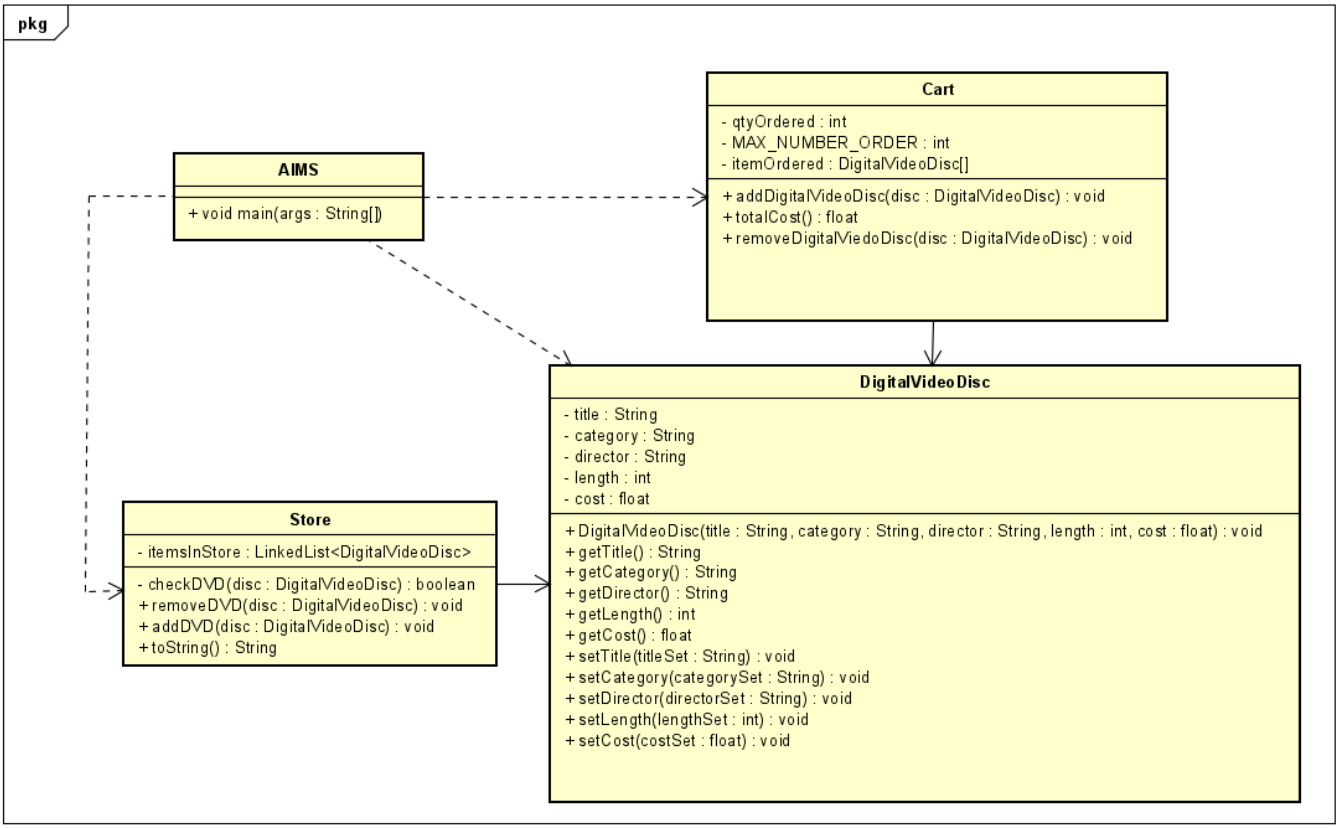


Figure 8.9: Exception