LÊ NHẬT HOÀNG 20245498

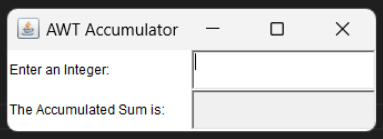
1. Swing components

1.1. AWTAccumulator

AWTAccumulator.java

GUIProject > hust > soict > dsai > swing > AWTAccumulator.java > AWTAccumulator

```java
package hust.soict.dsai.swing;
import java.awt.*;
import java.awt.event.*;

public class AWTAccumulator extends Frame {

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

   // Constructor to setup the GUI components and event handlers
   public AWTAccumulator() {
       setLayout(new GridLayout(2,2));

       add(new Label("Enter an Integer: "));

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

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

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

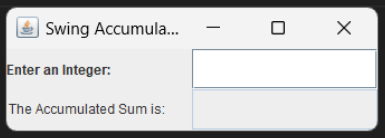


1.2. SwingAccumulator

SwingAccumulator.java

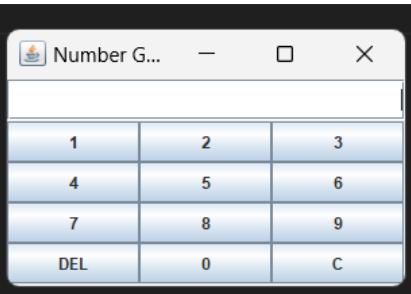
GUIProject > hust > soict > dsai > swing > SwingAccumulator.java > {} hust.soict.dsai.swing

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

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

import javax.swing.*;

public class SwingAccumulator extends JFrame {

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

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

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

       tfInput = new JTextField(10);
       cp.add(tfInput);
       tfInput.addActionListener(new TfInputListener());

       cp.add(new Label("The Accumulated Sum is: "));
```



2. Organizing Swing components with Layout Managers

```java
package hust.soict.dsai.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(200, 30));
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);

        JPanel panelButtons = new JPanel(new GridLayout(4,3));
        addButtons(panelButtons);

        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(tfDisplay, BorderLayout.NORTH);
        cp.add(panelButtons, BorderLayout.CENTER);
```



3. Create a graphical user interface for AIMS with Swing

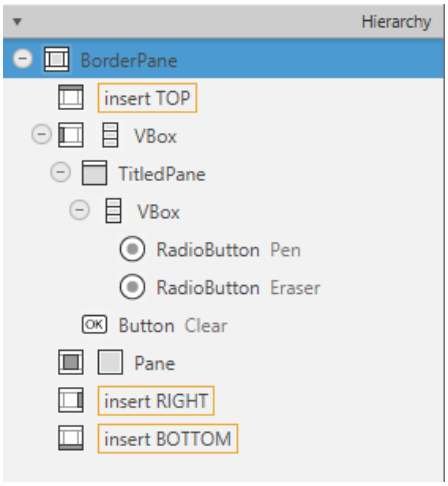
3.1. View Store Screen

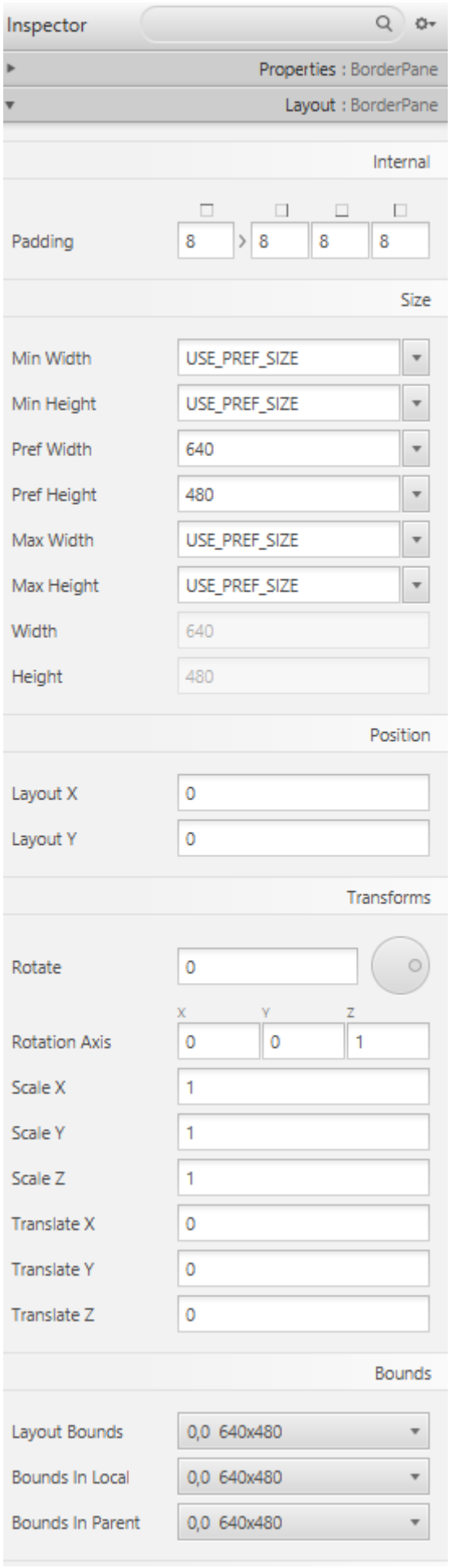
3.2. Adding more user interaction

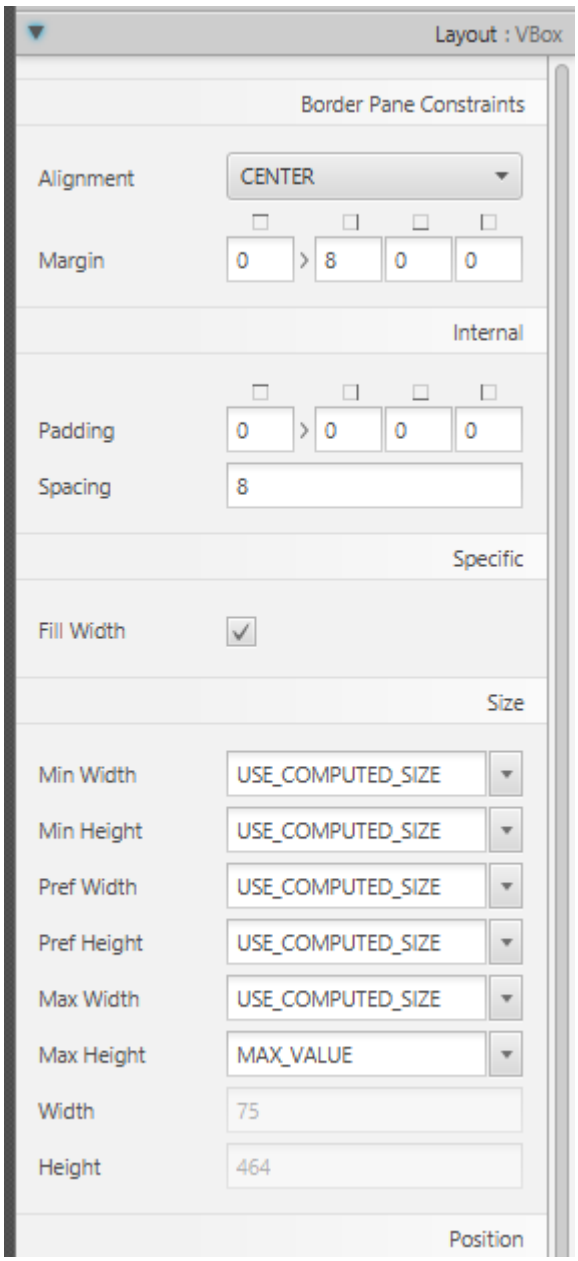
```java
JButton addCartBtn = new JButton("Add to Cart");
addCartBtn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            cart.addMedia(media);
        } catch (LimitExceededException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
});
container.add(addCartBtn);
```

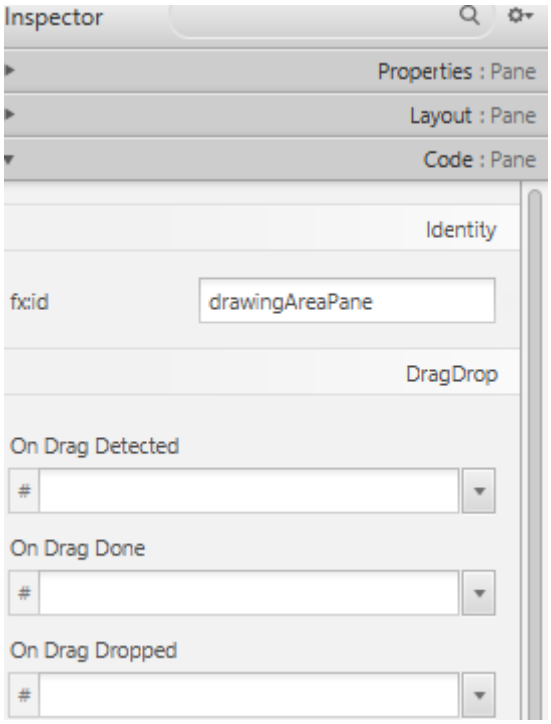
```java
if (media instanceof Playable) {
    JButton playBtn = new JButton("Play");
    playBtn.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            JDialog playDialog = new JDialog();
            JPanel mainGui = new JPanel(new BorderLayout());
            mainGui.setBorder(new EmptyBorder(20, 20, 20, 20));

            // Display Playing Message
            mainGui.add(new JLabel("Playing... " + media.getTitle()), BorderLayout.CENTER);
            System.out.println(media.getTitle());
            // Close Button
            JPanel buttonPanel = new JPanel(new FlowLayout());
            JButton close = new JButton("Stop");
            close.addActionListener(ev -> {
                playDialog.setVisible(false);
                System.out.println("Stopped playing.");
            });
            buttonPanel.add(close);
            mainGui.add(buttonPanel, BorderLayout.SOUTH);

            playDialog.setContentPane(mainGui);
            playDialog.setLocationRelativeTo(playBtn);
            playDialog.pack();

            // Show Dialog
            playDialog.setVisible(true);
        }
    });
    container.add(playBtn);
}
```

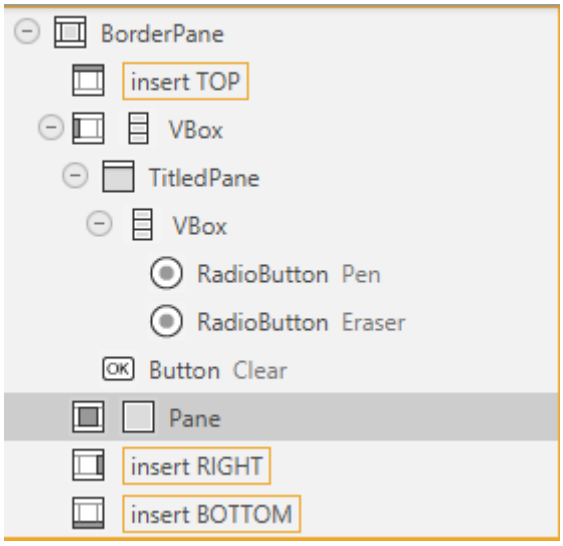
4. JavaFX API

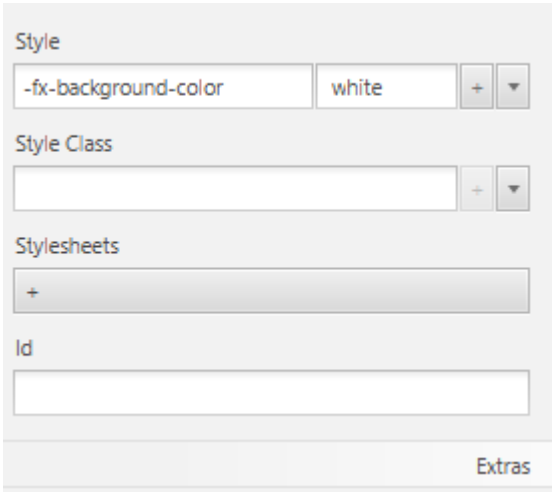


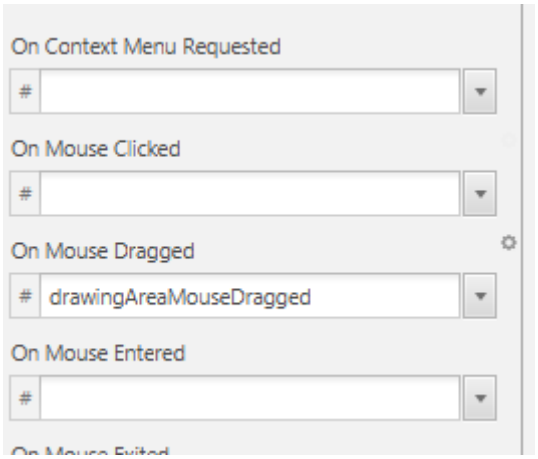

Inspector
Properties : BorderPane
Layout : BorderPane
Internal
Padding
8
8
8
8
Size
Min Width
USE_PREF_SIZE
Min Height
USE_PREF_SIZE
Pref Width
640
Pref Height
480
Max Width
USE_PREF_SIZE
Max Height
USE_PREF_SIZE
Width
640
Height
480
Position
Layout X
0
Layout Y
0
Transforms
Rotate
0
X
Y
Z
Rotation Axis
0
0
1
Scale X
1
Scale Y
1
Scale Z
1
Translate X
0
Translate Y
0
Translate Z
0
Bounds
Layout Bounds
0,0 640x480
Bounds In Local
0,0 640x480
Bounds In Parent
0,0 640x480

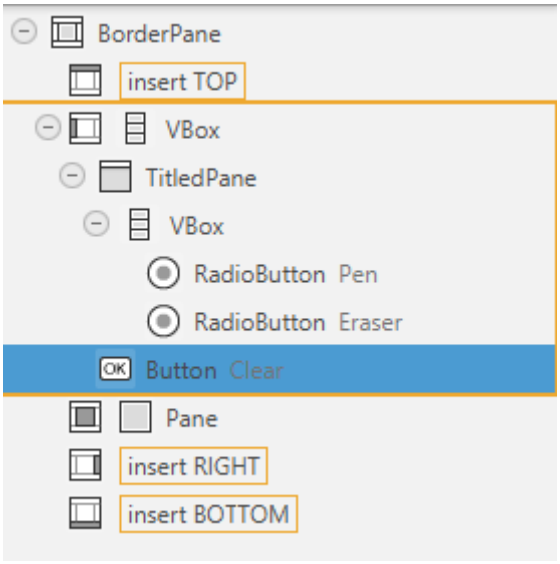

Layout : VBox
Border Pane Constraints
Alignment
CENTER
Margin
0
8
0
0
Internal
Padding
0
0
0
0
Spacing
8
Specific
Fill Width
Size
Min Width
USE_COMPUTED_SIZE
Min Height
USE_COMPUTED_SIZE
Pref Width
USE_COMPUTED_SIZE
Pref Height
USE_COMPUTED_SIZE
Max Width
USE_COMPUTED_SIZE
Max Height
MAX_VALUE
Width
75
Height
464
Position

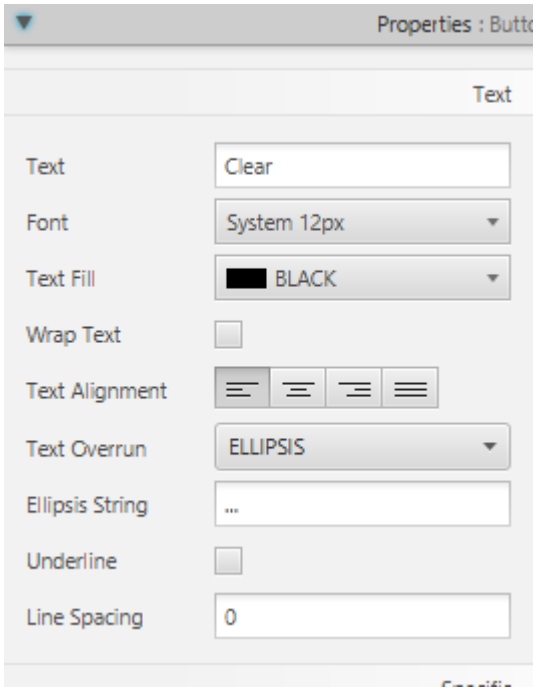

Inspector
Properties : Pane
Layout : Pane
Code : Pane
Identity
fx:id
drawingAreaPane
DragDrop
On Drag Detected
On Drag Done
On Drag Dropped

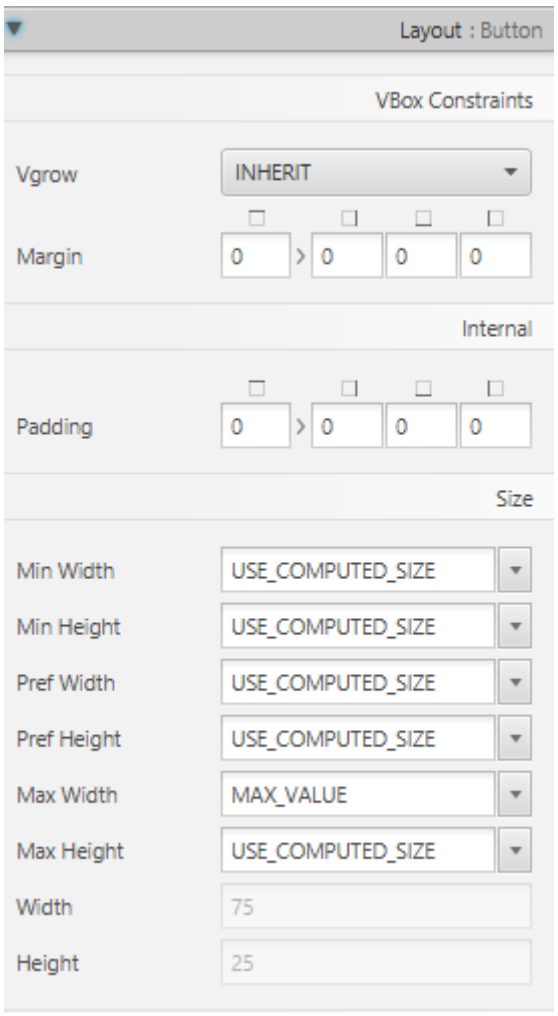

BorderPane
insert TOP
VBox
TitledPane
VBox
RadioButton Pen
RadioButton Eraser
Button Clear
Pane
insert RIGHT
insert BOTTOM

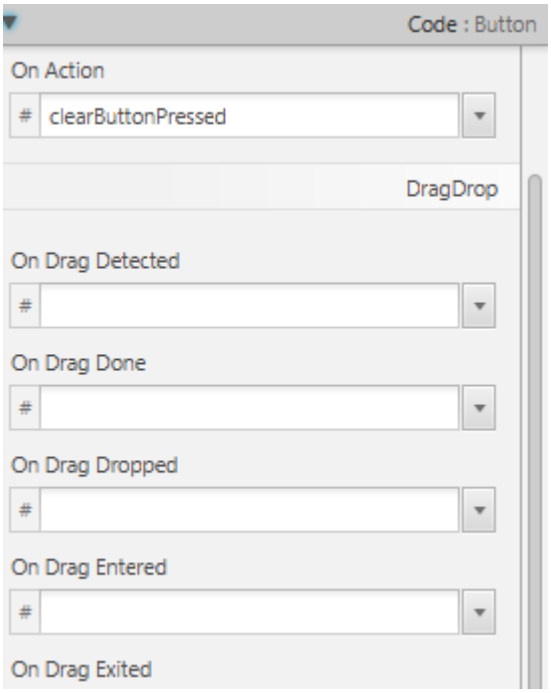

Style
-fx-background-color
white
Style Class
Stylesheets
Id
Extras

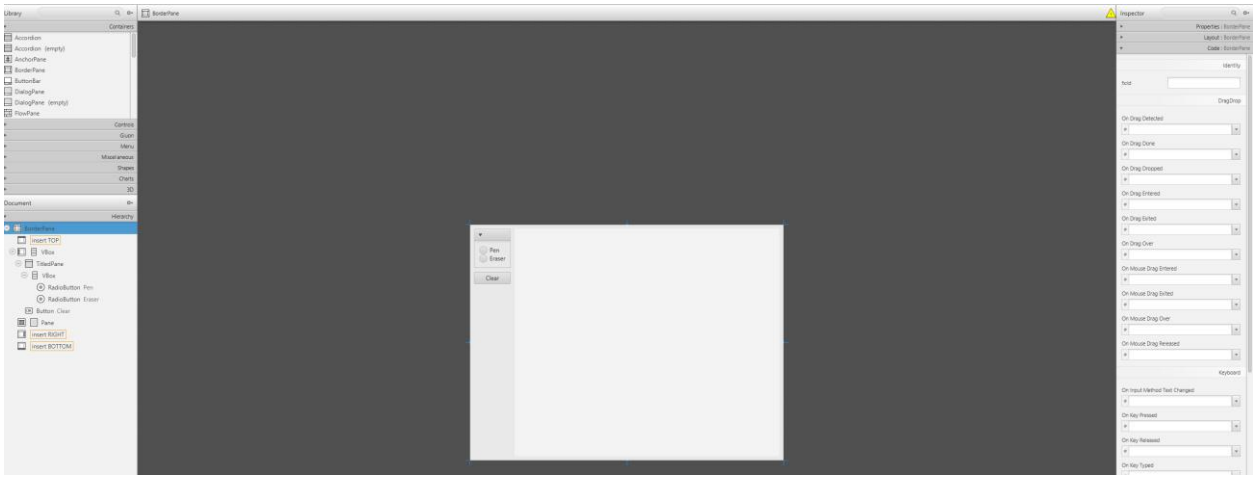

On Context Menu Requested
#
On Mouse Clicked
#
On Mouse Dragged
drawingAreaMouseDragged
On Mouse Entered
#


BorderPane
insert TOP
VBox
TitledPane
VBox
RadioButton Pen
RadioButton Eraser
Button Clear
Pane
insert RIGHT
insert BOTTOM


Properties : Butt
Text
Text
Clear
Font
System 12px
Text Fill
BLACK
Wrap Text
Text Alignment
Text Overrun
ELLIPSIS
Ellipsis String
...
Underline
Line Spacing
0


Layout : Button
VBox Constraints
Vgrow
INHERIT
Margin
0
0
0
0
Internal
Padding
0
0
0
0
Size
Min Width
USE_COMPUTED_SIZE
Min Height
USE_COMPUTED_SIZE
Pref Width
USE_COMPUTED_SIZE
Pref Height
USE_COMPUTED_SIZE
Max Width
MAX_VALUE
Max Height
USE_COMPUTED_SIZE
Width
75
Height
25


Code : Button
On Action
#
clearButtonPressed
DragDrop
On Drag Detected
#
On Drag Done
#
On Drag Dropped
#
On Drag Entered
#
On Drag Exited



```xml
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TitledPane?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0" xm
   <left>
      <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
         <BorderPane.margin>
            <Insets right="8.0" />
         </BorderPane.margin>
         <children>
            <TitledPane>
               <content>
                  <VBox>
                     <children>
                        <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Pen">
                           <toggleGroup>
                              <ToggleGroup fx:id="identity" />
                           </toggleGroup>
                        </RadioButton>
                        <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Eraser" toggleGroup="$identity" />
                     </children>
                  </VBox>
               </content>
            </TitledPane>
            <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear" />
         </children>
      </VBox>
   </left>
   <center>
      <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" style="-fx-background-color: white;" BorderPane.alignment="
   </center>
   <padding>
      <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
   </padding>
</BorderPane>
```

Painter.fxml M ● | PainterController.java M | Painter.java M ×

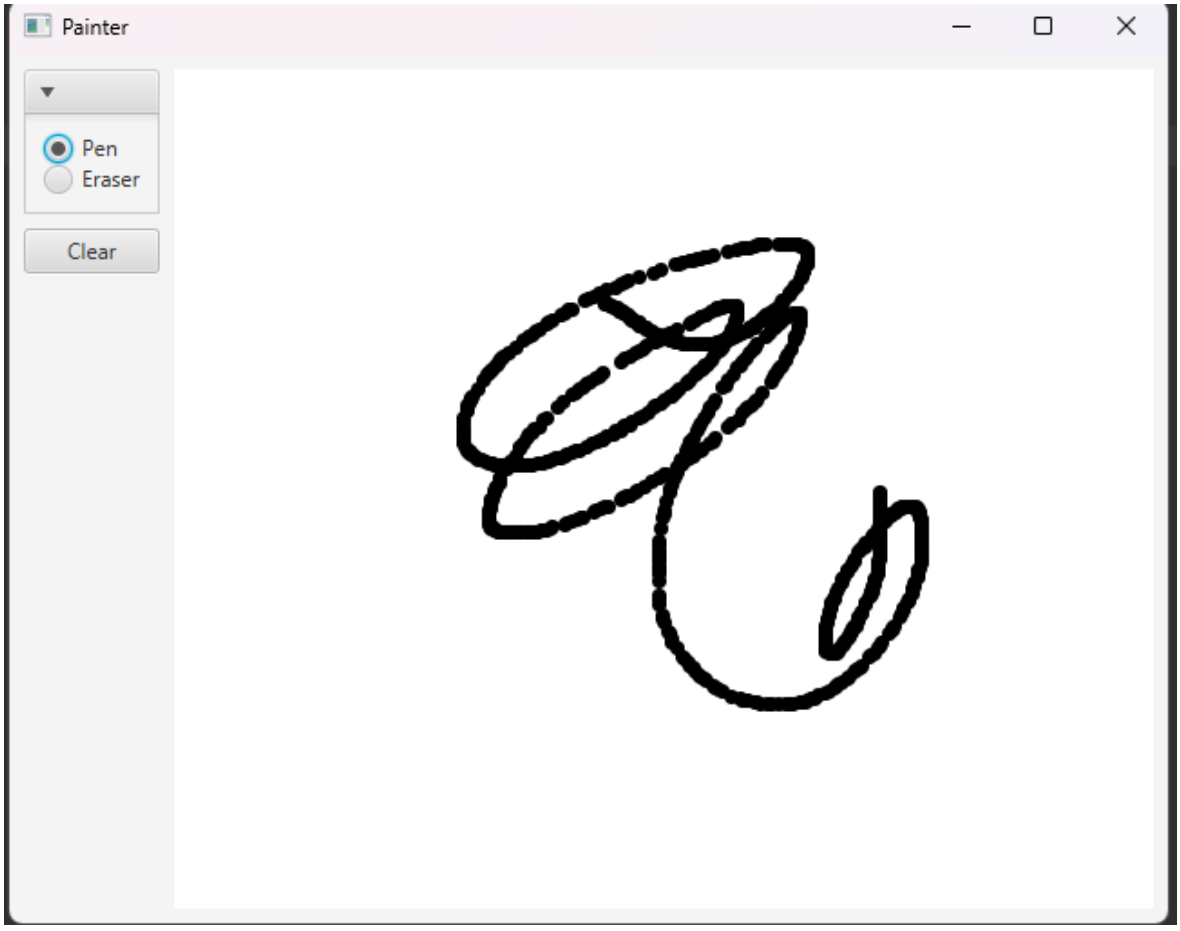
GUIProject > hust > soict > dsai > javaf

C:\Users\hoang le\Workspaces\GUIProject\hust\soict\dsai\javafx\PainterController.java
(preview)

```java
package hust.soict.dsai

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("/hust/soict/dsai/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);
    }
}
```

GUIProject > hust > soict > dsai > javafx > PainterController.java > PainterController

```java

public class PainterController {

    private int color;

    @FXML
    private ToggleGroup identity;

    @FXML
    void chooseOption(ActionEvent event) {
        String button = ((RadioButton)event.getSource()).getText();
        if (button.equals("Pen")) {
            System.out.println("1");
            color = 1;
        }else {
            System.out.println("0");
            color = 0;
        }
    }

    @FXML
    private Pane drawingAreaPane;

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

```

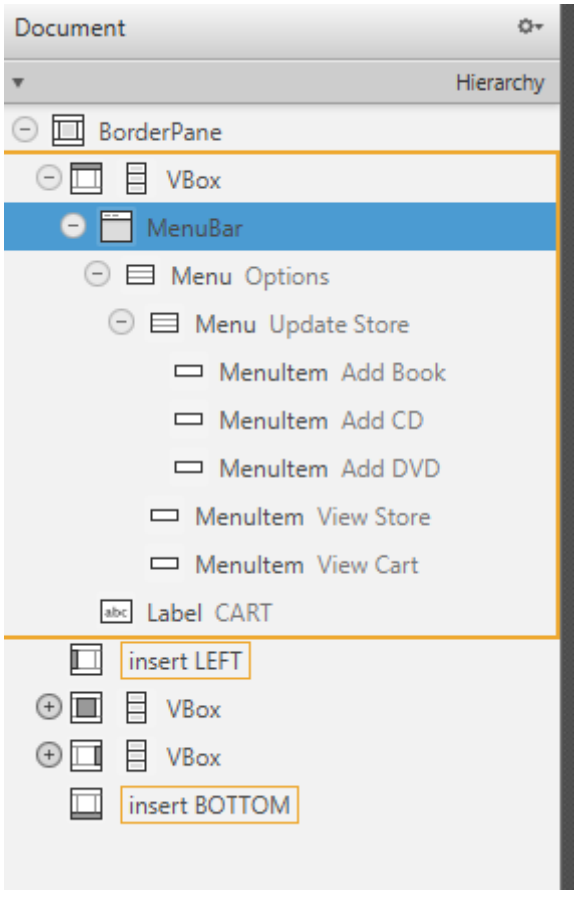


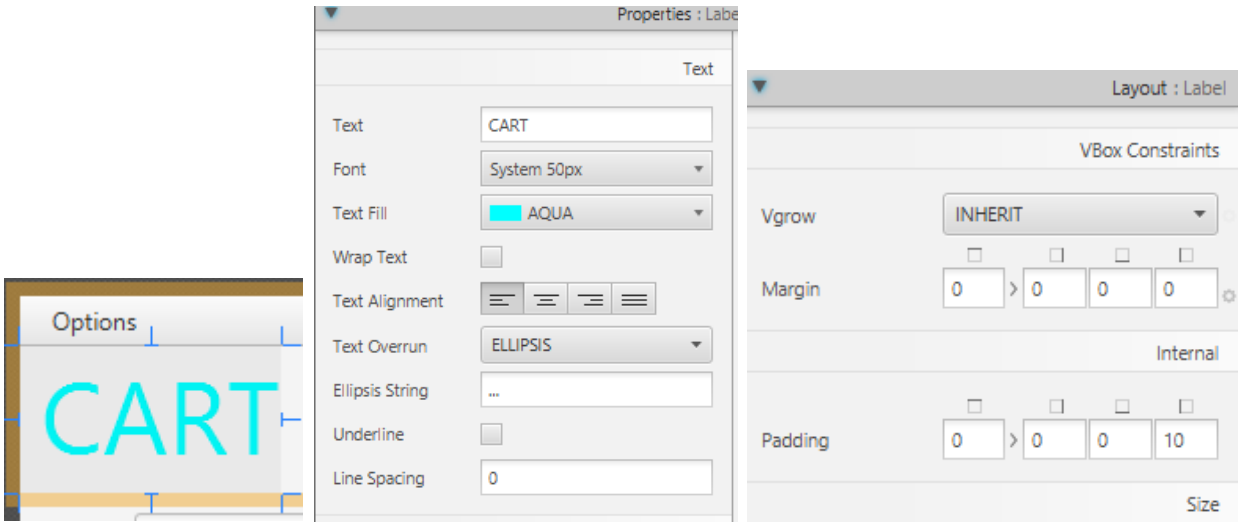
Handle Erase Function

```
    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        if (color == 1 && event.getX()>=0) {
            Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.BLACK);
            drawingAreaPane.getChildren().add(newCircle);
        }else if(color == 0 && event.getX()>=0){
            Circle newCircle = new Circle(event.getX(), event.getY(), 4.0, Color.WHITE);
            drawingAreaPane.getChildren().add(newCircle);
        }
    }

}
```

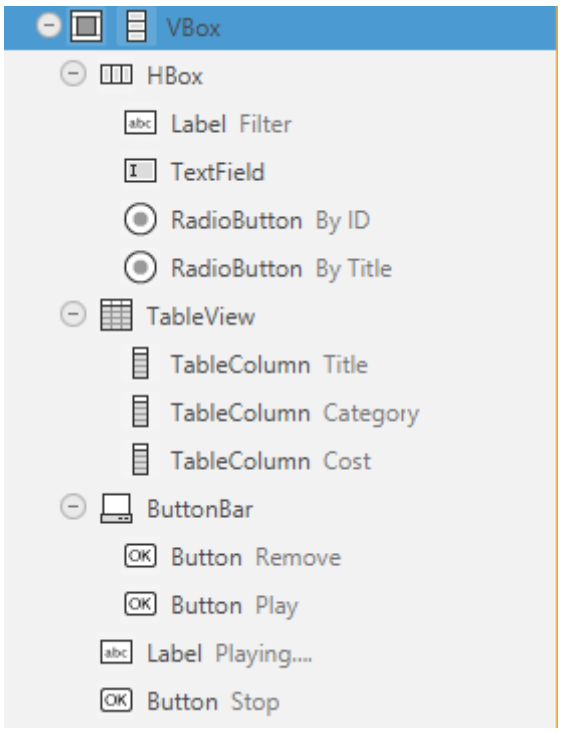
5. Setting up the View Cart Screen with ScreenBuilder

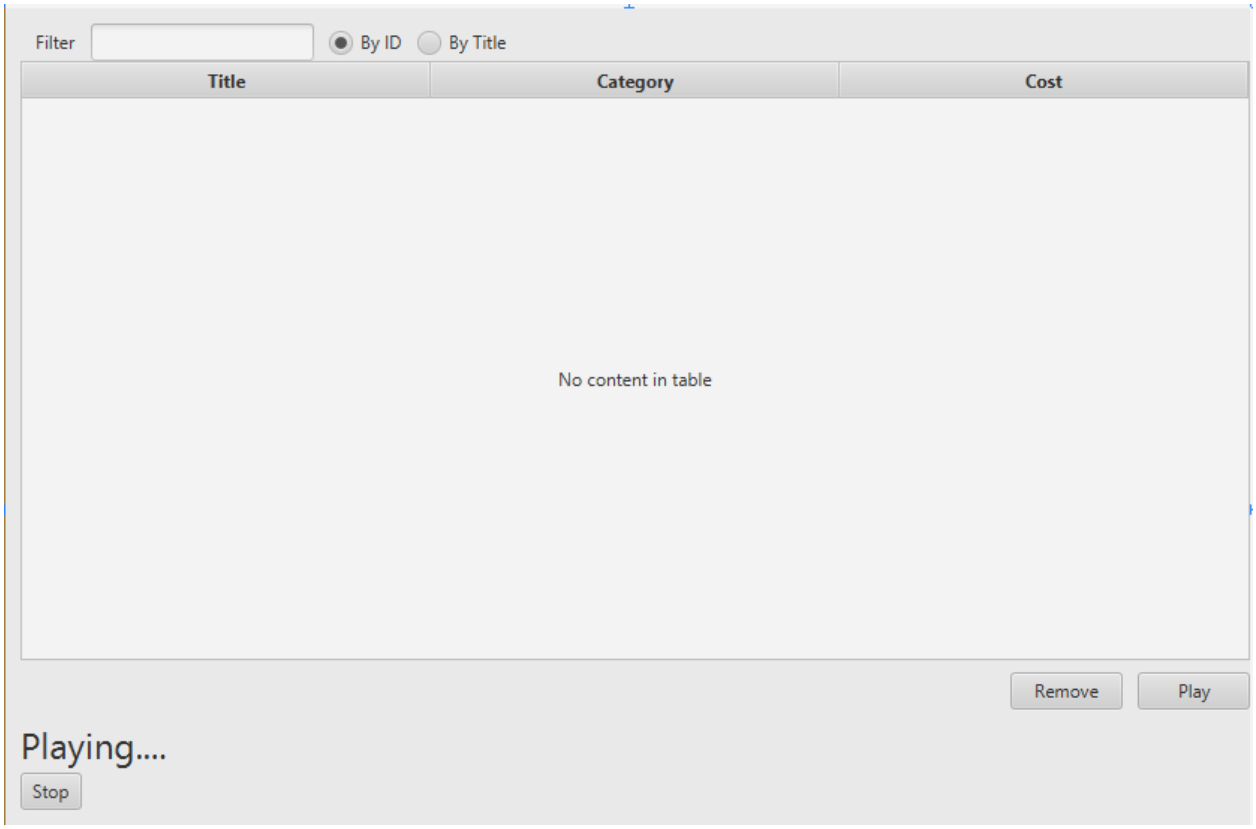
5.2. Setting up the TOP area



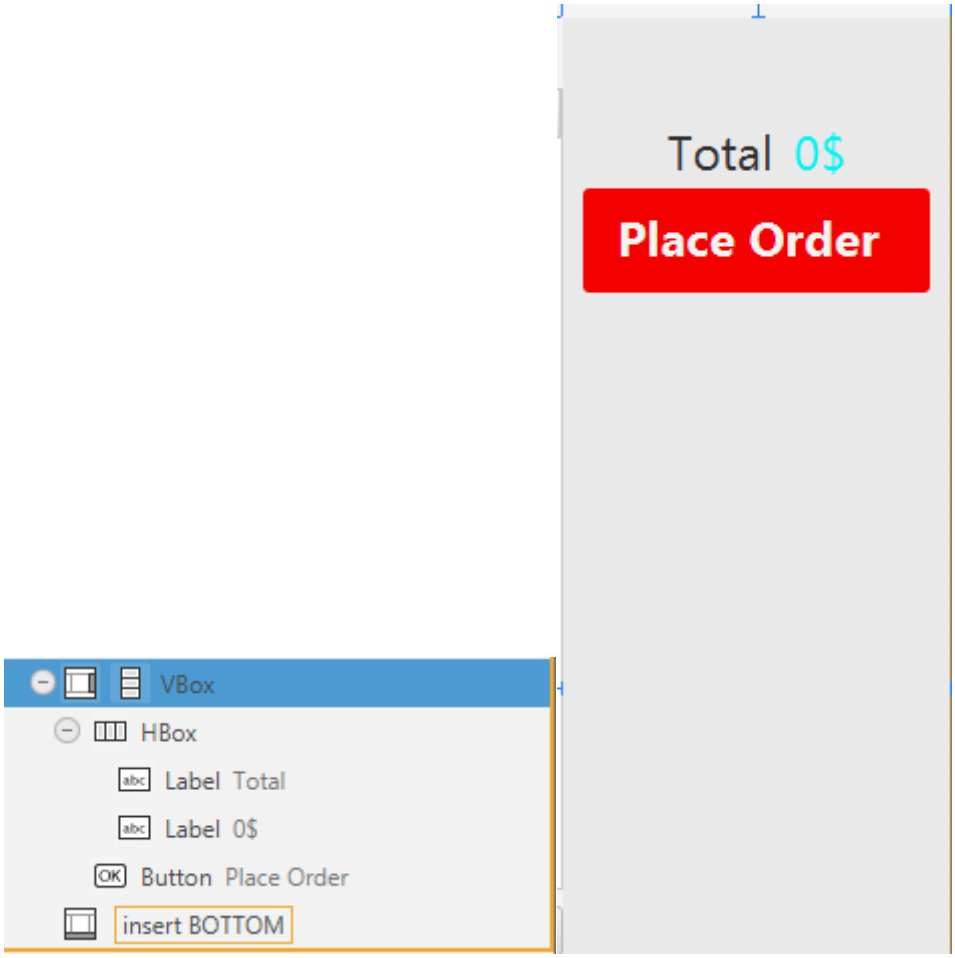


5.3. Setting up the CENTER area





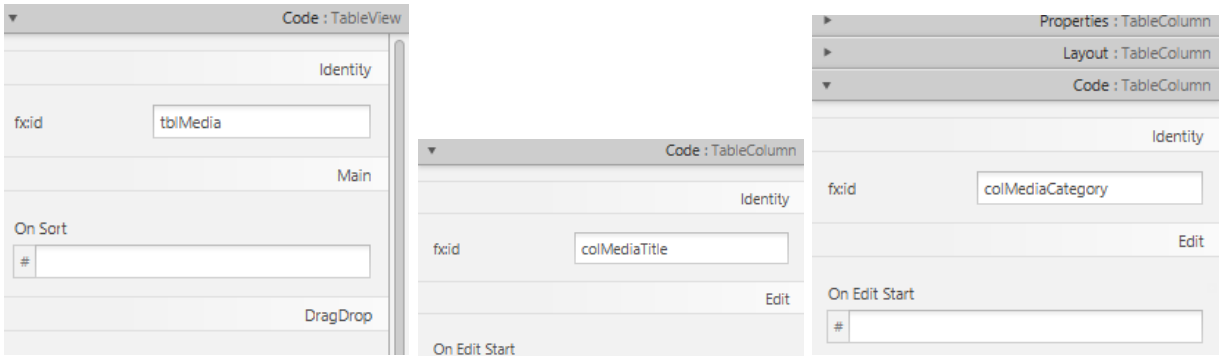
5.4. Setting up the RIGHT area

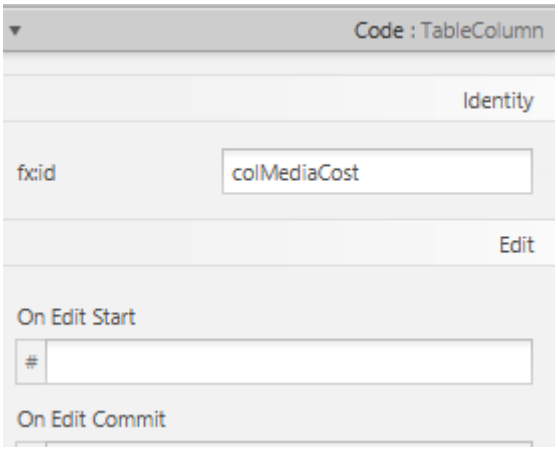


6. Integrating JavaFX into Swing application – The JFXPanel class

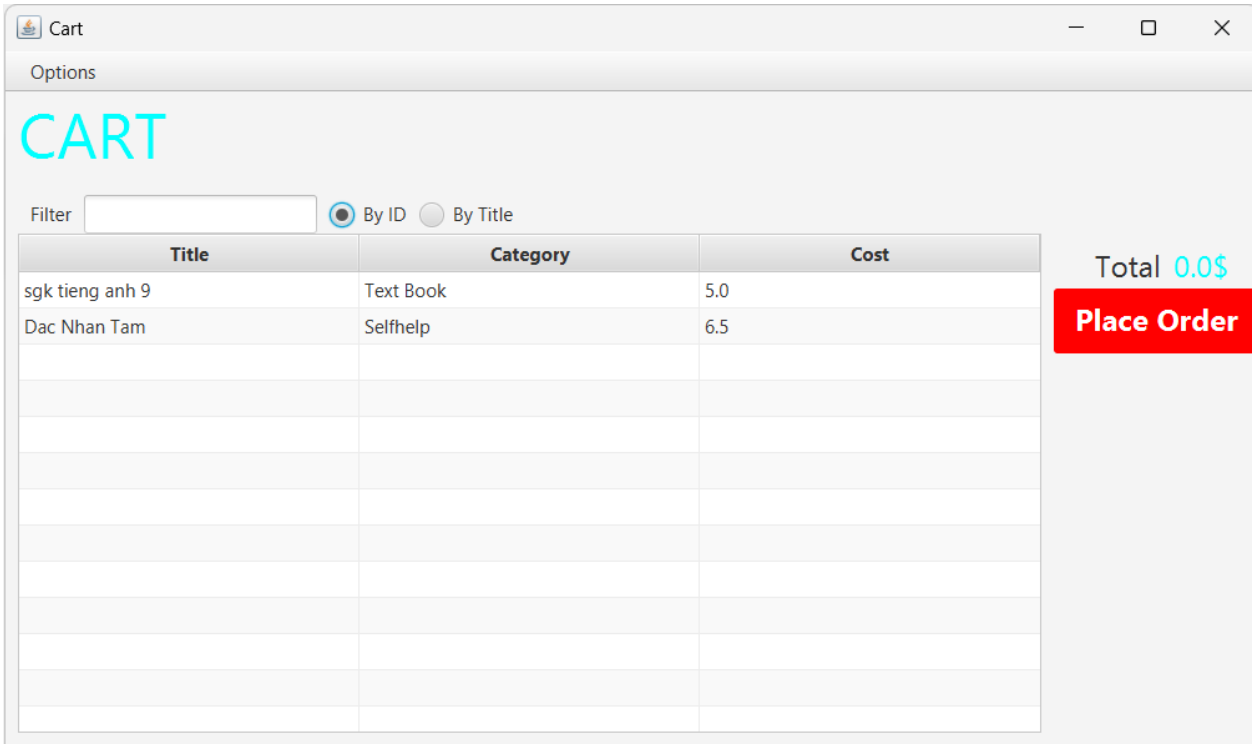
```java
public class CartScreen extends JFrame {

    private Cart cart;
    private ControllerScreen controllerScreen;

    public CartScreen(Cart cart, ControllerScreen c) {
        super();

        this.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle("Cart");
        // this.setVisible(true);
        this.setPreferredSize(new Dimension(1024, 768));
        pack();
        Platform.runLater(new Runnable() {

            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass()
                            .getResource("/hust/soict/dsai/aims/screen/cart.fxml"));
                    CartScreenController controller = new CartScreenController(cart, c);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root, 1024, 768));
                } catch (IOException e) {
                    e.printStackTrace();
                }

            }
        });
    }

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

}
```

7. View the items in cart – JavaFX's data-driven UI

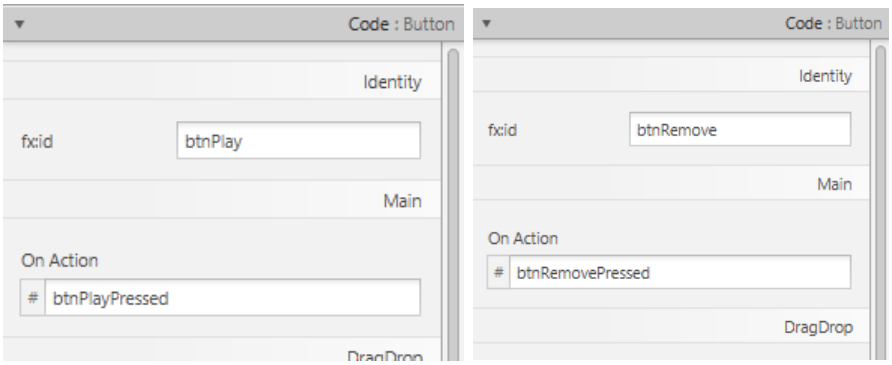




- Update itemsOrdered to ObservableList



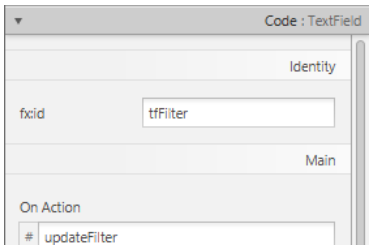
8. Updating buttons based on selected item in TableView – ChangeListener

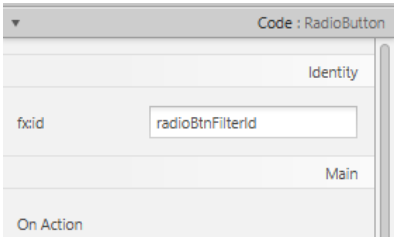


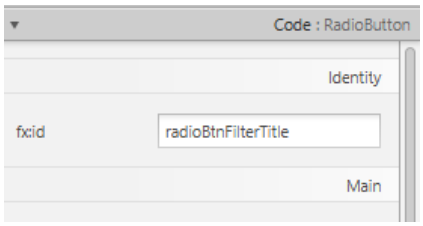
9. Deleting a media

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

10. Filter items in cart – FilteredList



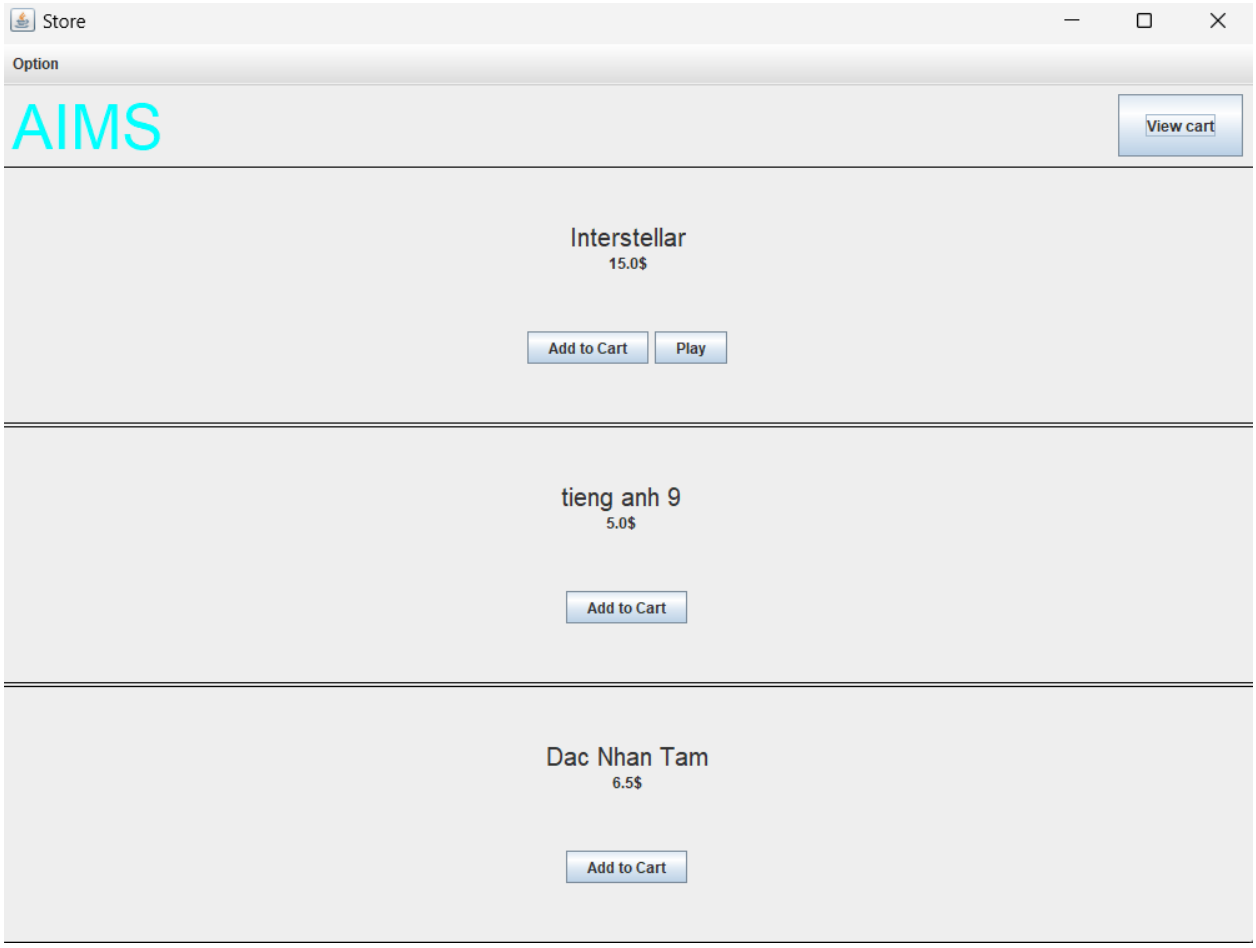


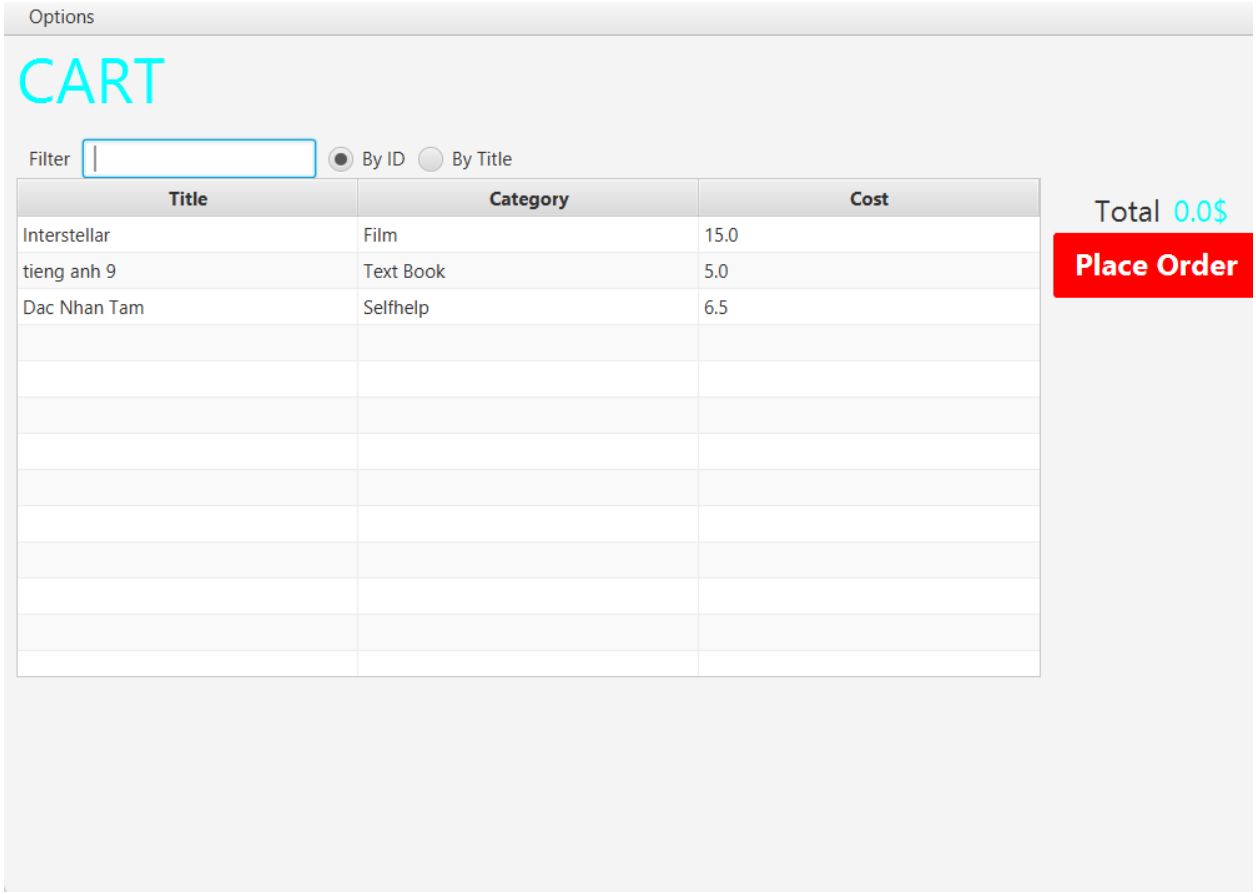



```
            @Override
            public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {

                showFilterMedia(newValue);
            }
        });
```

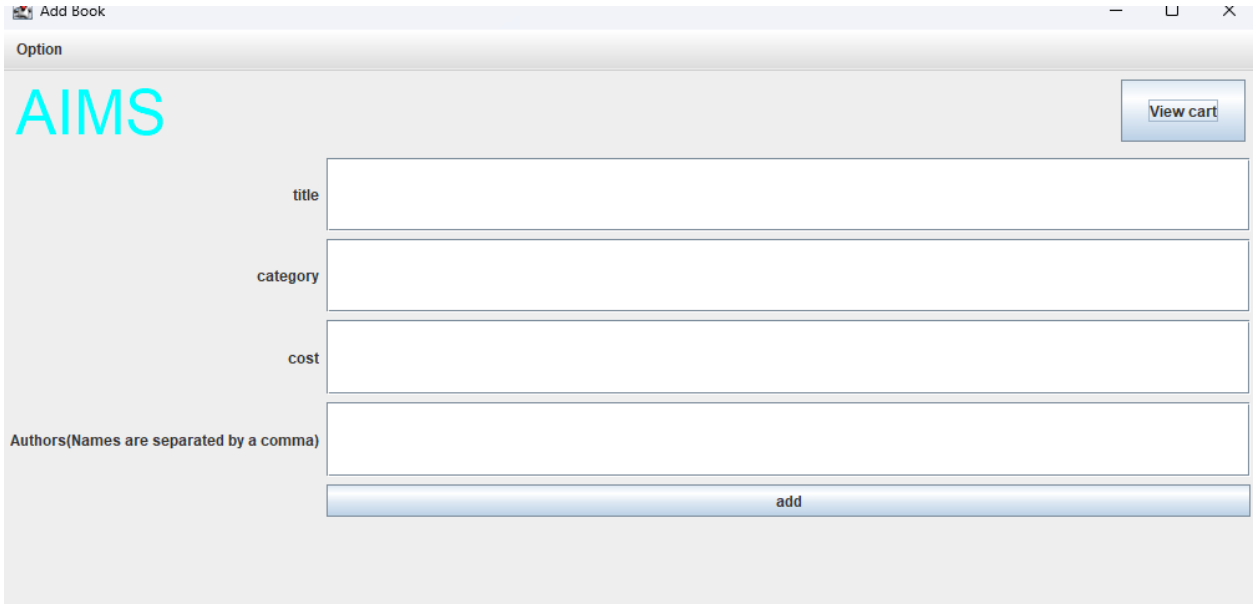
```
void showFilterMedia(String searchString) {
    if(searchString.isEmpty()) {
        tblMedia.setItems(this.cart.getItemsOrdered());
    } else {
        if(radioBtnFilterId.isSelected()) {
            tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getId()==Integer.parseInt(searchString)));
        }else
        tblMedia.setItems(new FilteredList<Media>(this.cart.getItemsOrdered(),item-> item.getTitle().contains(searchString)));
    }
}
```

11. Complete the Aims GUI application

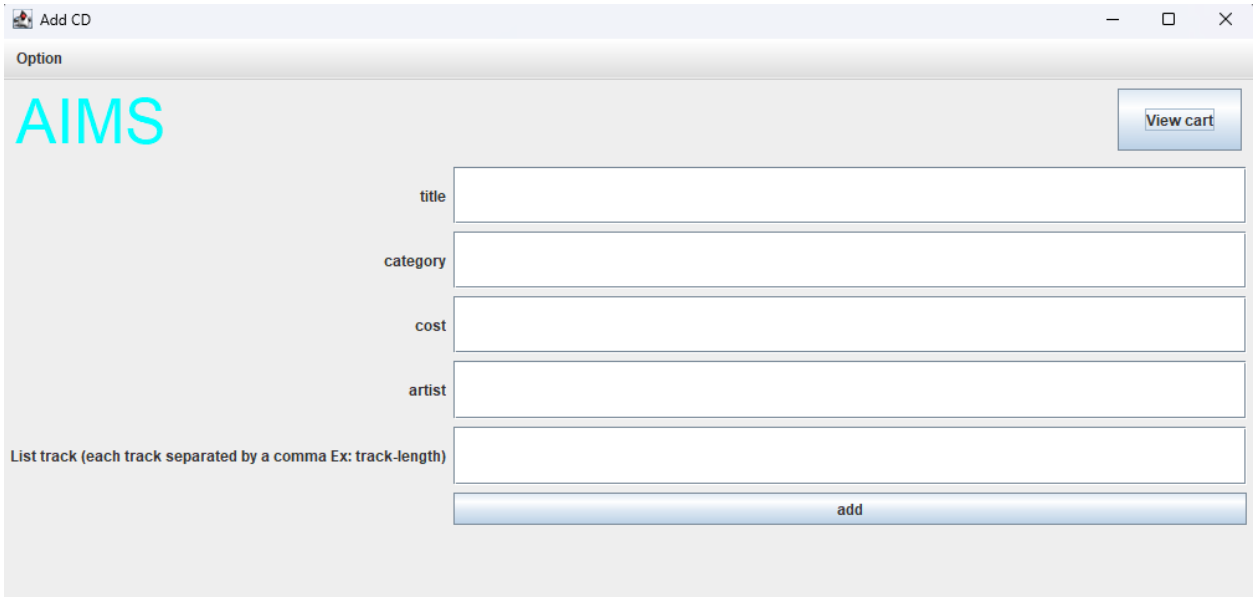

Store
Option
AIMS
View cart
Interstellar
15.0$
Add to Cart
Play
tieng anh 9
5.0$
Add to Cart
Dac Nhan Tam
6.5$
Add to Cart


Options
CART
Filter
By ID
By Title
Title
Category
Cost
Interstellar
Film
15.0
tieng anh 9
Text Book
5.0
Dac Nhan Tam
Selfhelp
6.5
Total 0.0$
Place Order

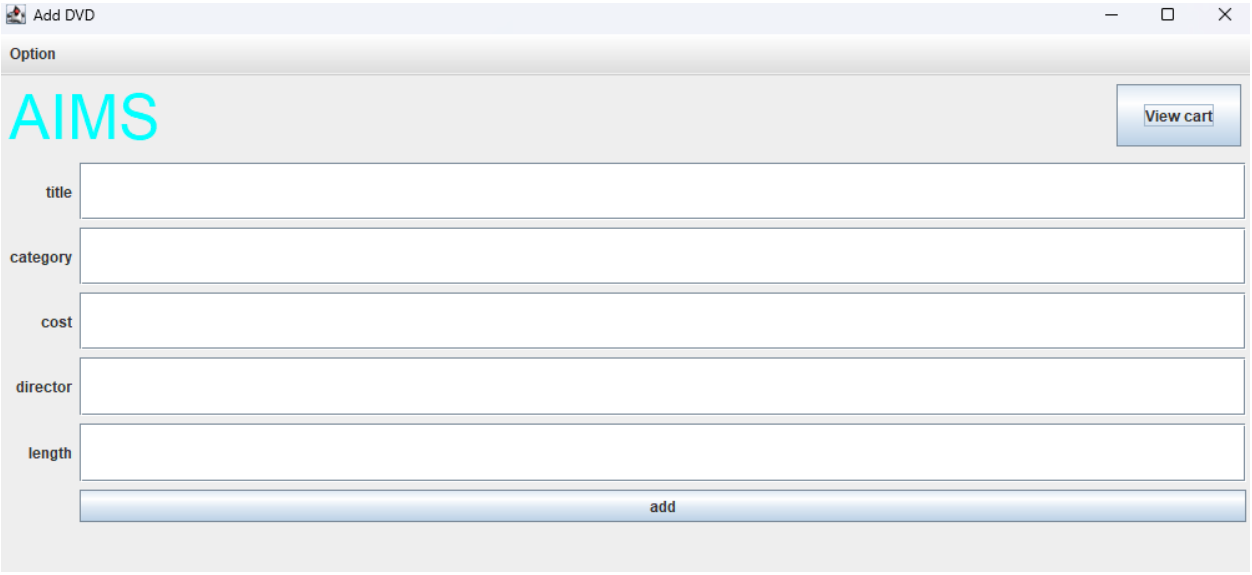
Add Book



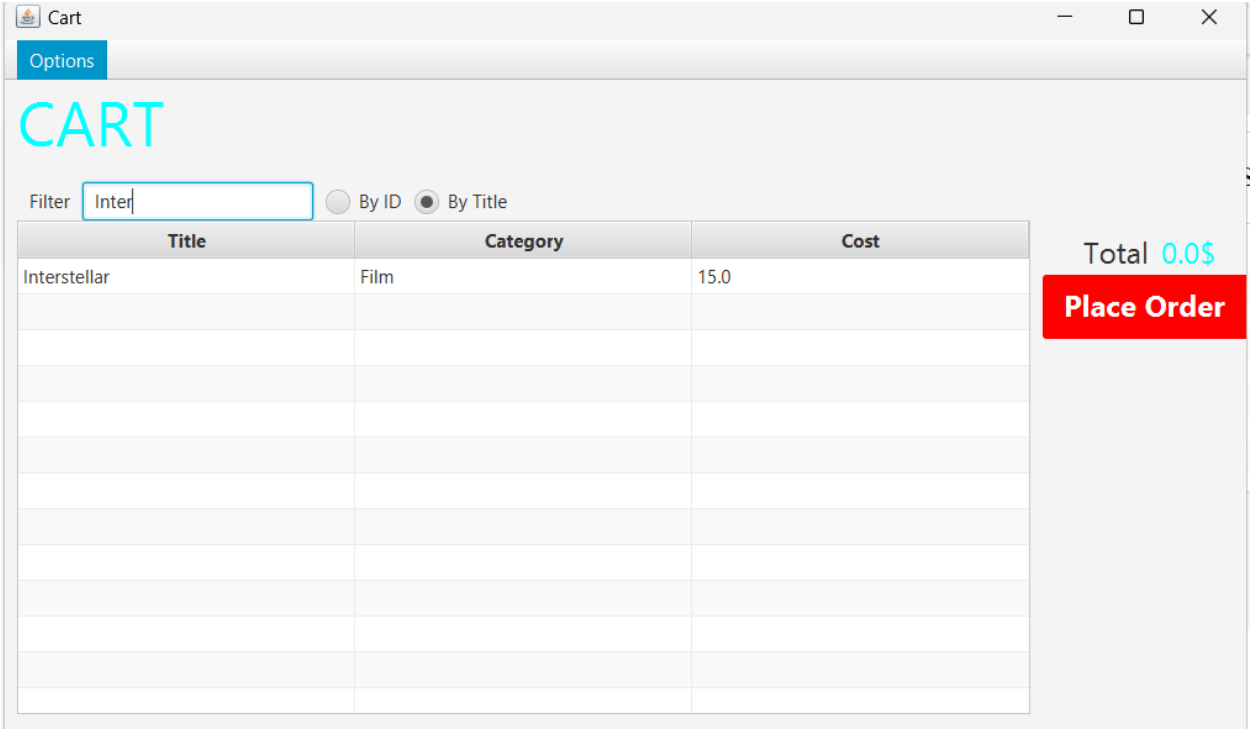
Add CD



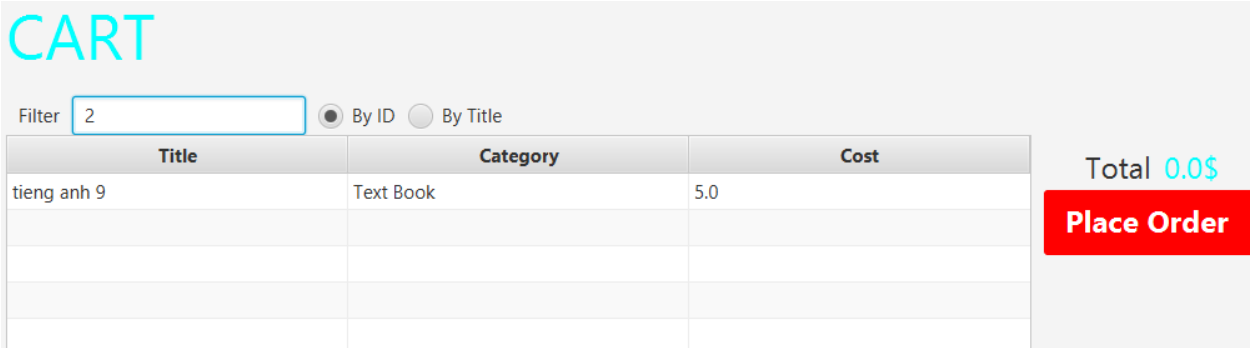
Add DVD



Filter by Title



Filter by Id



AddItemToStoreScreen

```
public class AddItemToStoreScreen extends JFrame {
    protected Store store;
    protected Cart cart;
    protected JTextField title;
    protected JTextField category;
    protected JTextField cost;
    protected JButton addBtn;
    protected ControllerScreen c;
    protected JPanel center;
    protected int numberInput = 3;

    JMenuBar createMenuBar() {
        JMenu menu = new JMenu("Option");

        JMenu smUpdateStore = new JMenu("Update Store");
        JMenuItem addBookMenu = new JMenuItem("Add Book");
        addBookMenu.addActionListener(e -> {
            c.showAddBookScreen();
        });
        smUpdateStore.add(addBookMenu);
        JMenuItem addCDMenu = new JMenuItem("Add CD");
        addCDMenu.addActionListener(e -> {
            c.showAddCDCreen();
        });
        smUpdateStore.add(addCDMenu);
        JMenuItem addDVDMenu = new JMenuItem("Add DVD");
        addDVDMenu.addActionListener(e -> {
            c.showAddDVDScreen();
        });
        smUpdateStore.add(addDVDMenu);

        menu.add(smUpdateStore);
        JMenuItem viewStoreMenu = new JMenuItem("View store");
        viewStoreMenu.addActionListener(e -> {
            c.showStoreScreen();
        });
        menu.add(viewStoreMenu);

        JMenuItem viewCartMenu = new JMenuItem("View cart");
        viewCartMenu.addActionListener(e -> {
            c.showCartScreen();
```

AddDigitalVideoDiscToStoreScreen

```
public class AddDigitalVideoDiscToStoreScreen extends AddItemToStoreScreen {
    private JTextField director;
    private JTextField length;

    public AddDigitalVideoDiscToStoreScreen(Store store, Cart cart, ControllerScreen c) {
        super(store, cart, c);

    }
    @Override
    void updateAdd(JPanel panel) {
        this.numberInput = 6;
        // SpringLayout layout =new SpringLayout();
        // JPanel p = new JPanel(layout);
        JLabel directorLabel = new JLabel("director", JLabel.TRAILING);
        panel.add(directorLabel);
        director = new JTextField(2);
        director.setPreferredSize(new Dimension(150, 20));
        directorLabel.setLabelFor(director);
        panel.add(director);
        JLabel lengthLabel = new JLabel("length", JLabel.TRAILING);
        panel.add(lengthLabel);
        length = new JTextField(2);
        lengthLabel.setLabelFor(length);
        panel.add(length);
        JButton tes = new JButton("add");
        tes.setVisible(false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(100, 300));
        addBtn = new JButton("add");
        addBtn.addActionListener(e -> {
            addMedia();
        });
        panel.add(addBtn);
    }

    public void addMedia() {
        String title = this.title.getText();
        String director = this.director.getText();
        String category = this.category.getText();
        float cost = Float.parseFloat(this.cost.getText());
        int length = Integer.parseInt(this.length.getText());
        DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, director, length, cost);
```

AddCompactDiscToStoreScreen

```java
public class AddCompactDiscToStoreScreen extends AddItemToStoreScreen {
    private JTextField artist;
    private JTextField listTrack;

    public AddCompactDiscToStoreScreen(Store store, Cart cart, ControllerScreen c) {
        super(store, cart, c);

    }

    @Override
    void updateAdd(JPanel panel) {
        this.numberInput = 6;

        JLabel artistLabel = new JLabel("artist", JLabel.TRAILING);
        panel.add(artistLabel);
        artist = new JTextField(2);
        artist.setPreferredSize(new Dimension(150, 20));
        artistLabel.setLabelFor(artist);
        panel.add(artist);
        JLabel listTrackLabel = new JLabel("List track (each track separated by a comma Ex: track-length)",
                JLabel.TRAILING);

        panel.add(listTrackLabel);
        listTrack = new JTextField(2);
        listTrackLabel.setLabelFor(listTrack);
        panel.add(listTrack);
        JButton tes = new JButton("add");
        tes.setVisible(false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(100, 300));
        addBtn = new JButton("add");
        addBtn.addActionListener(e -> {
            addMedia();
        });
        panel.add(addBtn);
    }

    public void addMedia() {
        String title = this.title.getText();
        String listTrack = this.listTrack.getText();
        String category = this.category.getText();
        float cost = Float.parseFloat(this.cost.getText());
```

AddBookToStoreScreen

```java
public class AddBookToStoreScreen extends AddItemToStoreScreen {
    private JTextField listAuthor;

    public AddBookToStoreScreen(Store store, Cart cart, ControllerScreen c) {
        super(store, cart, c);

    }
    @Override
    void updateAdd(JPanel panel) {
        this.numberInput = 5;

        JLabel listAuthorLabel = new JLabel("Authors(Names are separated by a comma)", JLabel.TRAILING);
        panel.add(listAuthorLabel);
        listAuthor = new JTextField(2);
        listAuthor.setPreferredSize(new Dimension(150, 20));
        listAuthorLabel.setLabelFor(listAuthor);
        panel.add(listAuthor);

        JButton tes = new JButton("add");
        tes.setVisible(false);
        panel.add(tes);
        panel.setPreferredSize(new Dimension(100, 300));
        addBtn = new JButton("add");
        addBtn.addActionListener(e -> {
            addMediaToStore();
        });
        panel.add(addBtn);
    }

    public void addMediaToStore() {
        String title = this.title.getText();
        String listAuthor = this.listAuthor.getText();
        String[] arrayAuthor=listAuthor.split(",");
        String category = this.category.getText();
        float cost = Float.parseFloat(this.cost.getText());
        Book book = new Book(title, category, cost);
        for(String author:arrayAuthor) {
        book.addAuthor(author);
        }
        this.store.addMedia(book);
        JOptionPane.showMessageDialog(null, "add Book successfully!");
        clearTextField();
```

12. Check all the previous source codes to catch/handle/delegate runtime exceptions

```java
public void addMedia(Media media) throws LimitExceededException {
    if ((itemsOrdered.size()) >= MAX_ORDERED) {
        throw new LimitExceededException("Full");
    }
    else if (itemsOrdered.contains(media)) {
        System.out.println("This is already in your order!");
    }
    else {
        itemsOrdered.add(media);
        System.out.println("Media added!");
        }
}
```

13. Create a class which inherits from Exception

```java

package hust.soict.dsai.aims.exception;

public class PlayerException extends RuntimeException {

    public PlayerException() {
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message) {
        super(message);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(Throwable cause) {
        super(cause);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message, Throwable cause) {
        super(message, cause);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
        // TODO Auto-generated constructor stub
    }

}
```

CompactDisc

```java
@Override
public void play() throws PlayerException{
    System.out.println("CompactDisc Artist: " + this.getArtist());
    System.out.println("Total length: " + this.getLength());

    if (this.getLength() > 0) {
        System.out.println("Compactdisc: " + this.getTitle());
        System.out.println("CompactDisc Artist: " + this.getArtist());
        System.out.println("Total length: " + this.getLength());
        java.util.Iterator iter = tracks.iterator();
        Track nextTrack;
        while (iter.hasNext()) {
            nextTrack = (Track) iter.next();
            try {
                nextTrack.play();
            }
            catch(PlayerException e ) {
                throw e;
            }
        }
    }
    else {
        throw new PlayerException(message:"Error: CD length is non-positive!");
    }

    System.out.println("----------------Play All Tracks----------------");
    for (Track track: tracks) {
        track.play();
    }
}
```

DVD

```
@Override
public void play() throws PlayerException{
    if (this.getLength() > 0) {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
    else {throw new PlayerException(message:"Error: DVD length is non-positive!");}
}
```

Track

```
    @Override
    public void play() throws PlayerException {
        if (this.getLength() > 0) {
            System.out.println("Playing track: " + this.getTitle());
            System.out.println("Track length: " + this.getLength());
        }
        else {throw new PlayerException(message:"Error: Track length is non-positive!");}
    }
}
```

14. Update the Aims class

```
        switch (choice) {
            case 1:
                cart.addMedia(media);
                System.out.println("Media added to cart.");
                break;
            case 2:
                if (media instanceof Playable) {
                    if(media instanceof DigitalVideoDisc){
                        try {
                            ((DigitalVideoDisc) media).play();
                        } catch (PlayerException e) {
                            // TODO Auto-generated catch block
                            e.getMessage();
                            e.toString();
                            e.printStackTrace();
                        }
                    }
                    else if(media instanceof CompactDisc){
                        try {
                            ((CompactDisc)media).play();
                        } catch (PlayerException e) {
                            // TODO Auto-generated catch block
                            e.getMessage();
                            e.toString();
                            e.printStackTrace();
                        }
                    }
                } else {
                    System.out.println("This media cannot be played.");
                }
                break;
            case 0:
                break;
            default:
                System.out.println("Invalid choice! Please choose a number between 0-2.");
        }
    } while (choice != 0);
```

```java
public static void playMedia(Scanner scanner) {
    System.out.print("Enter the title of the media to play: ");
    String title = scanner.nextLine();
    Media media = store.searchByTitle(title);

    if (media != null) {
        if (media instanceof Playable) {
            if(media instanceof DigitalVideoDisc){
                try {
                    ((DigitalVideoDisc) media).play();
                } catch (PlayerException e) {
                    // TODO Auto-generated catch block
                    e.getMessage();
                    e.toString();
                    e.printStackTrace();
                }
            }
            else if(media instanceof CompactDisc){
                try {
                    ((CompactDisc)media).play();
                } catch (PlayerException e) {
                    // TODO Auto-generated catch block
                    e.getMessage();
                    e.toString();
                    e.printStackTrace();
                }
            }
        } else {
            System.out.println("This media cannot be played.");
        }
    } else {
        System.out.println("Media not found in the store.");
    }
}
```

15. Modify the equals() method of Media class

```java
@Override
public boolean equals(Object o) {
    if (o instanceof Media) {
        Media media = (Media) o;
        if (this.title.equals(media.title)) {
            return true;
        }else {
            return false;
        }
    }
    return false;
}
```

17. Update Aims class diagram