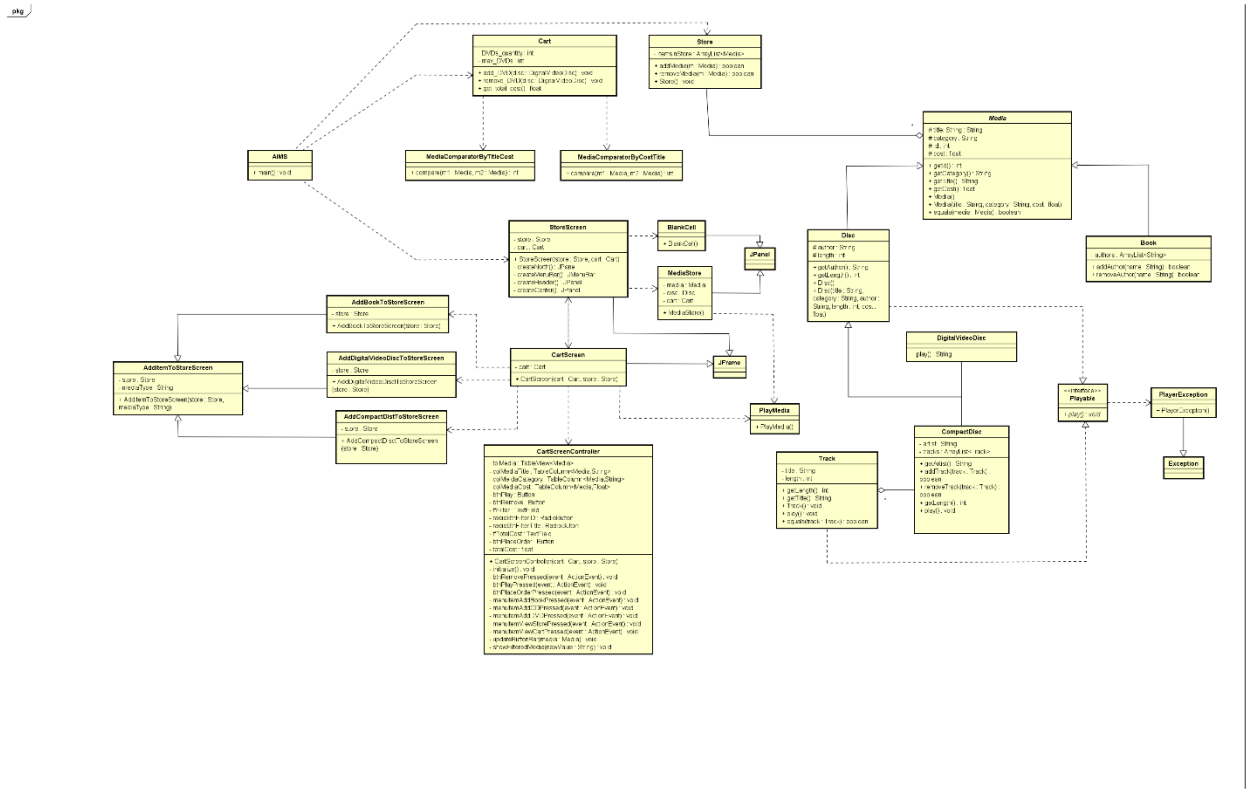
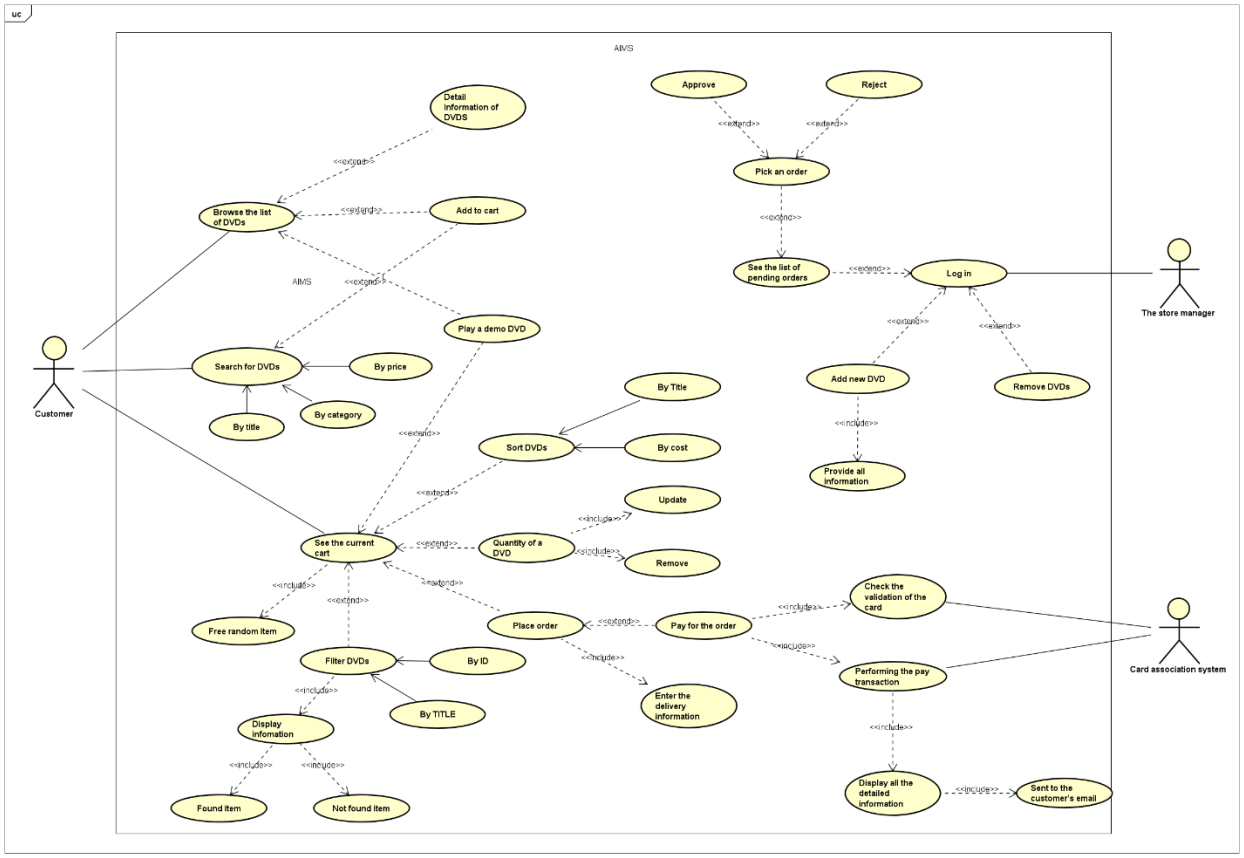
Report

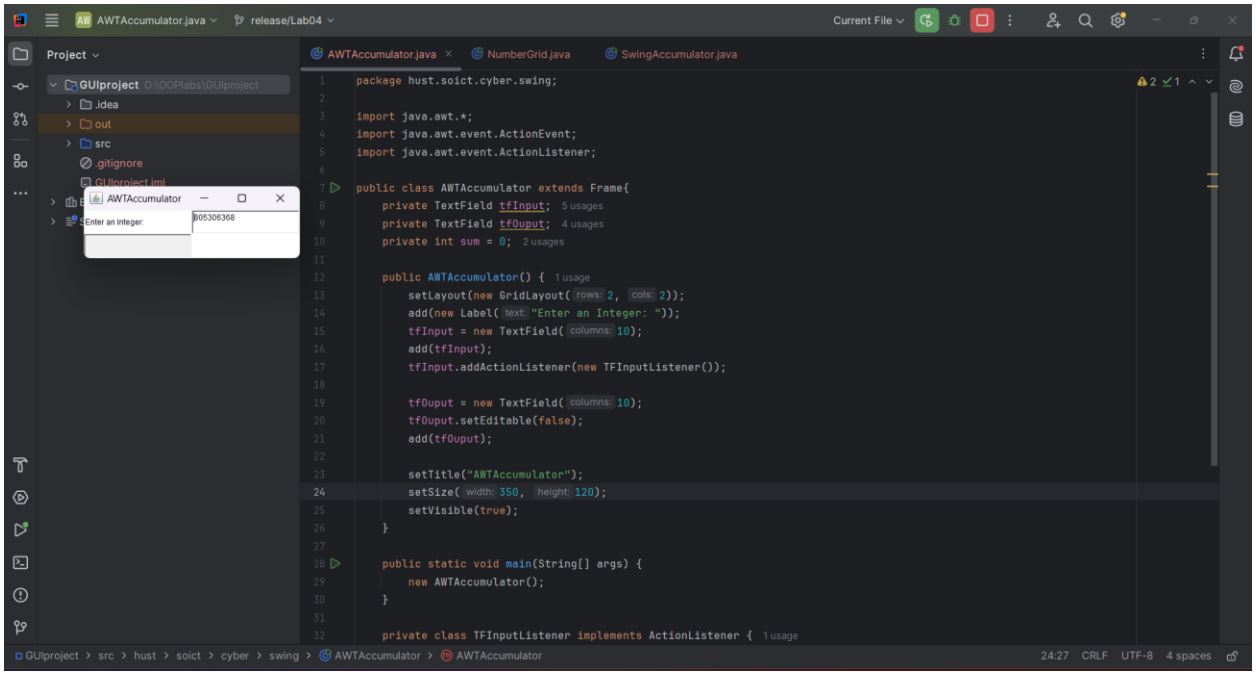
Cập nhập use case và class



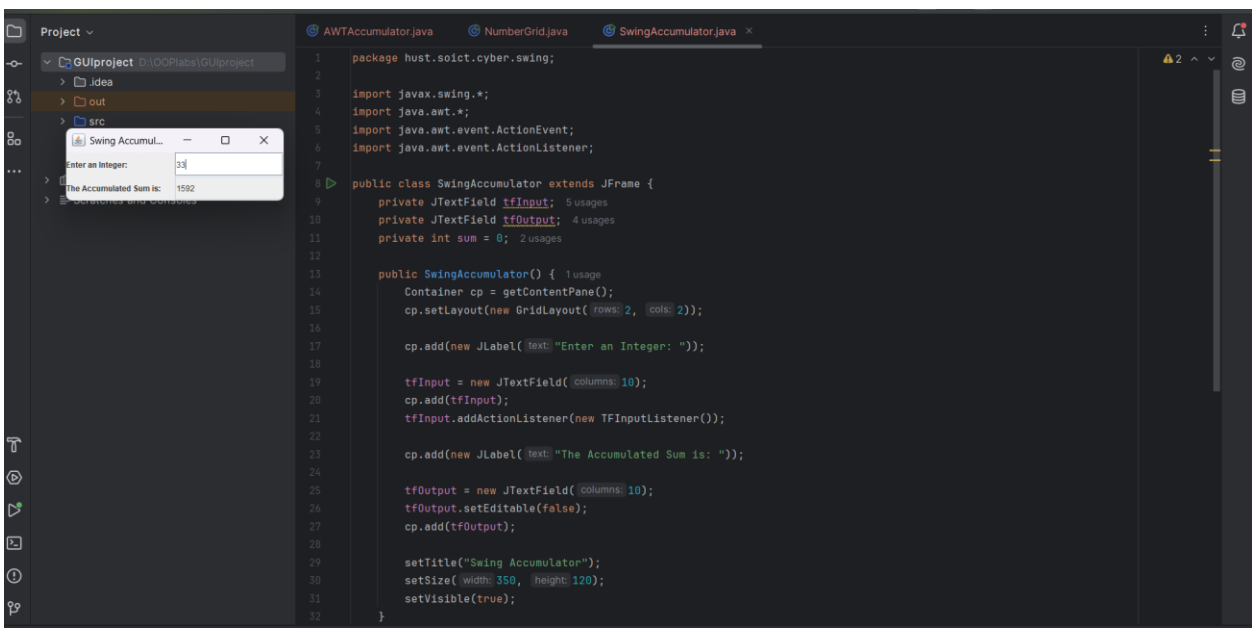


1. Swing

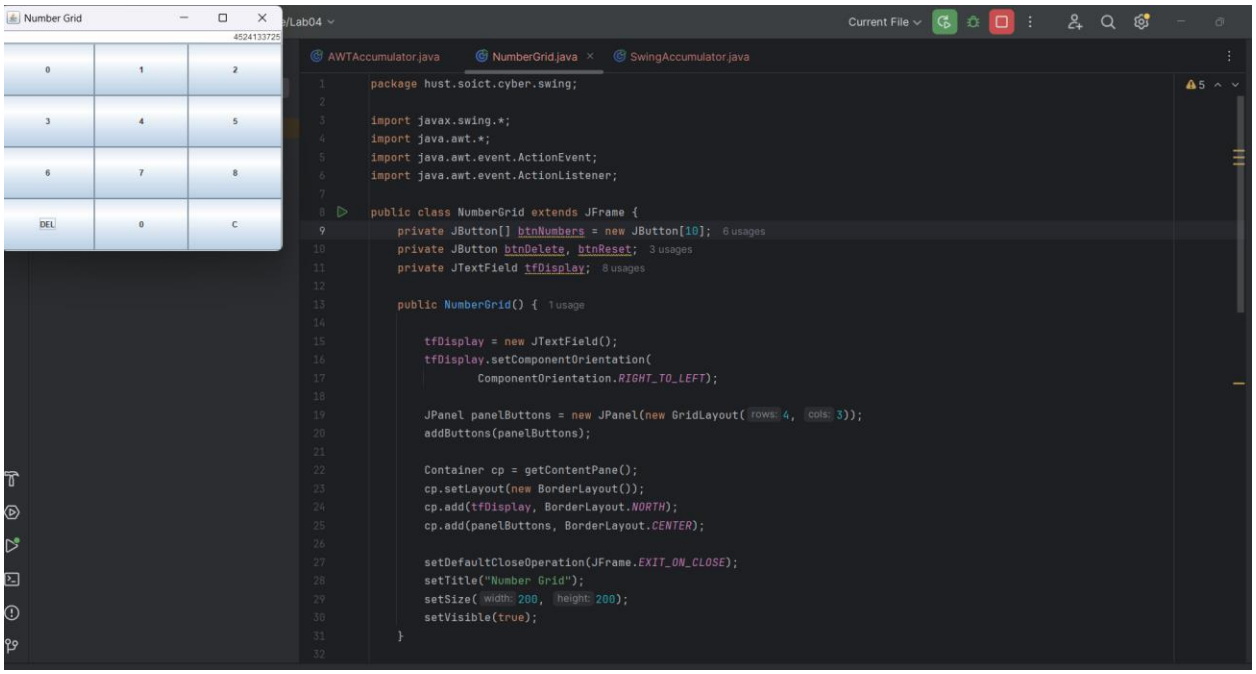
1.1 AWTAccumulator



1.2. SwingAccumulator

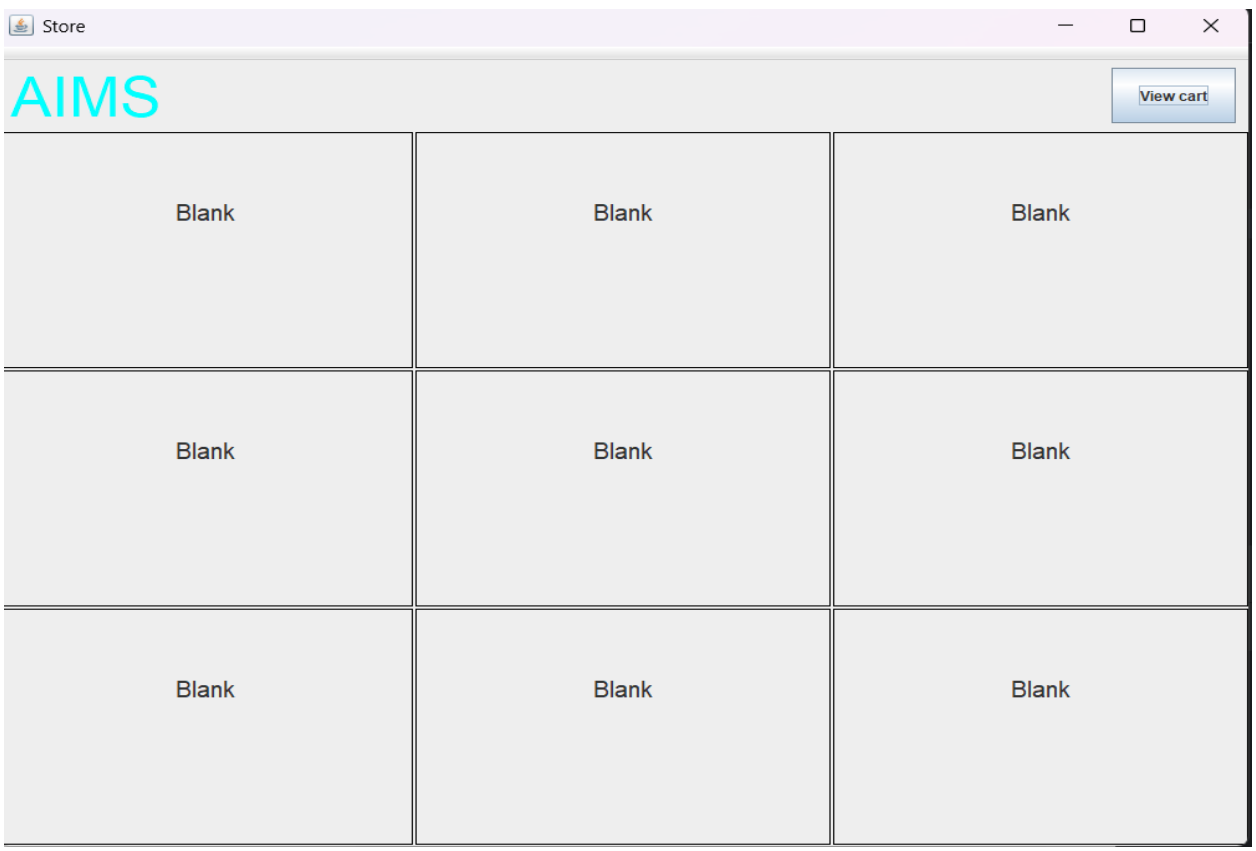


2. Organizing Swing components with Layout Managers



3. Create a graphical user interface for AIMS with Swing

3.1. View Store Screen



3.2 StoreScreen class

```java
public class StoreScreen extends JFrame {  3 usages
    private Store store;  3 usages
    private Cart cart;  3 usages

    JPanel createNorth() {  1 usage
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }

    JMenuBar createMenuBar() {  1 usage
        JMenu menu = new JMenu( s: "Options");

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

        menu.add(smUpdateStore);
        menu.add(new JMenuItem( text: "View store"));
        menu.add(new JMenuItem( text: "View cart"));

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

        return menuBar;
    }
```

```java
JPanel createHeader() {  1 usage
    JPanel header = new JPanel();
    header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

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

    JButton cart = new JButton( text: "View cart");
    cart.addMouseListener(new ViewCartMouseListener());
    cart.setPreferredSize(new Dimension( width: 100,  height: 50));
    cart.setMaximumSize(new Dimension( width: 100,  height: 50));

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

    return header;

}
```

```java
JPanel createCenter() {  1 usage
    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 < 9; i++){
        try {
            MediaStore cell = new MediaStore(mediaInStore.get(i), cart);
            center.add(cell);
        } catch (IndexOutOfBoundsException e) {
            BlankCell cell = new BlankCell();
            center.add(cell);
        }
    }

    return center;

}

// putting all together
public StoreScreen(Store store, Cart cart) {  2 usages
    this.store = store;
    this.cart = cart;
    Container cp = getContentPane();
    cp.setLayout(new BorderLayout());

    cp.add(createNorth(), BorderLayout.NORTH);
    cp.add(createCenter(), BorderLayout.CENTER);

    setVisible(true);
    setTitle("Store");
    setSize( width: 1024,  height: 768);
```

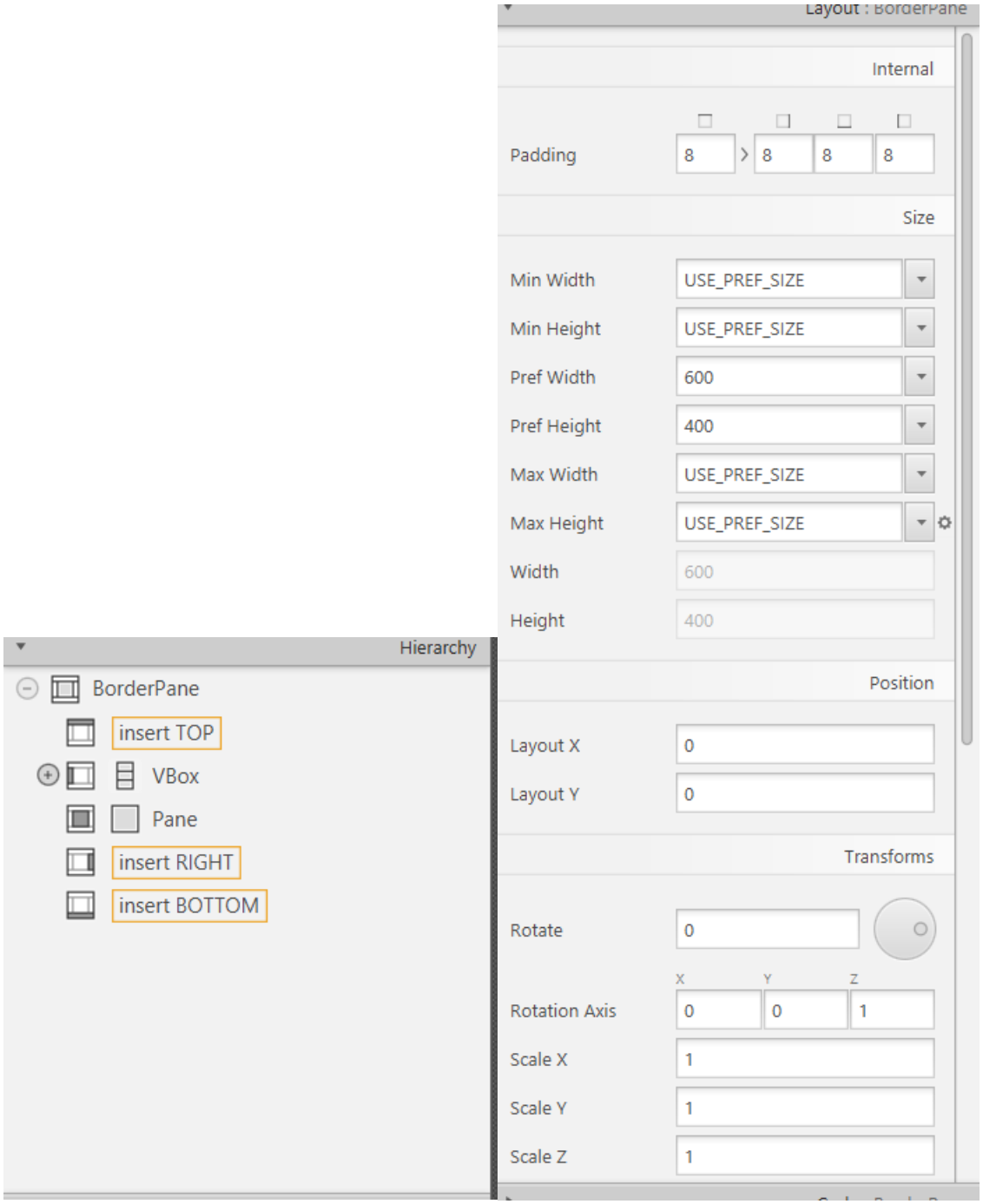
3.3 The MediaStore class

```java
public class MediaStore extends JPanel {  2 usages
    private Media media;  2 usages
    private Playable disc;  2 usages
    private Cart cart;  3 usages

    public MediaStore (Media media, Cart cart) {  1 usage
        this.media = media;
        this.cart = cart;
        this.setLayout(new BoxLayout( target: this, BoxLayout.Y_AXIS));

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

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

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

        // Add to cart button
        JButton addCartButton = new JButton( text: "Add to cart");
        container.add(addCartButton);
        addCartButton.addMouseListener(new AddToCartListener());

        // Play button -> for playable classes
        if (media instanceof Playable) {
            JButton PlayButton = new JButton( text: "Play");
            container.add(PlayButton);
            disc = (Playable) media;
            PlayButton.addMouseListener(new PlayButtonListener());
        }
```

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

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

// Button Mouse Listener

private class PlayButtonListener implements MouseListener{  1 usage
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}

    public void mouseClicked(MouseEvent e) {
        try {
            new PlayMedia(disc);
        } catch (PlayerException pe) {
            pe.printStackTrace();
        }
    }
}
```

```java
    // Add to class Listener
    private class AddToCartListener implements MouseListener {  1 usage
        public void mousePressed(MouseEvent e) {}
        public void mouseReleased(MouseEvent e) {}
        public void mouseEntered(MouseEvent e) {}
        public void mouseExited(MouseEvent e) {}

        public void mouseClicked(MouseEvent e) {
            try {
                AddToCart adc = new AddToCart(media, cart);
                cart = adc.getUpdatedCart();
            } catch (LimitExceededException lee) {
                lee.printStackTrace();
            }
        }
    }
}
```

4. JavaFX API


Layout : BorderPane
Internal
Padding
8
8
8
8
Size
Min Width
USE_PREF_SIZE
Min Height
USE_PREF_SIZE
Pref Width
600
Pref Height
400
Max Width
USE_PREF_SIZE
Max Height
USE_PREF_SIZE
Width
600
Height
400
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
Hierarchy
BorderPane
insert TOP
VBox
Pane
insert RIGHT
insert BOTTOM

Code : Pane

Identity

fx:id drawingAreaPane

DragDrop

On Drag Detected

On Drag Done

On Drag Dropped

On Drag Entered

On Drag Exited

On Drag Over

On Mouse Drag Entered

On Mouse Drag Exited

On Mouse Drag Over

On Mouse Drag Released

Keyboard

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 75.2

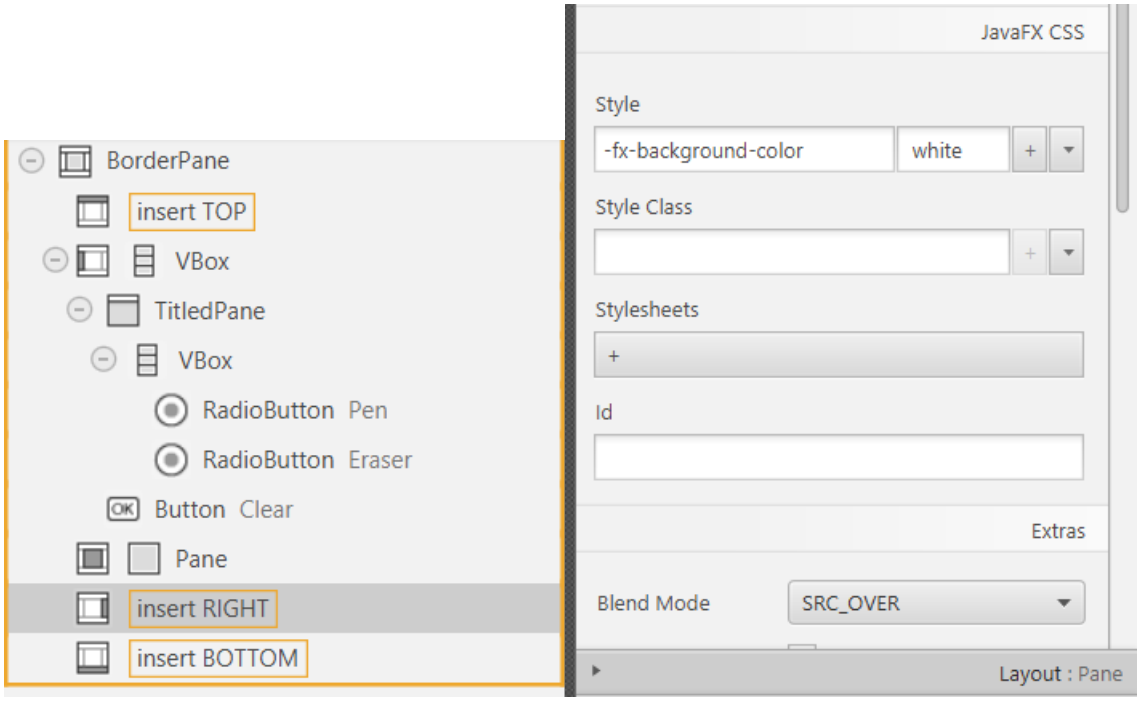
Height 384

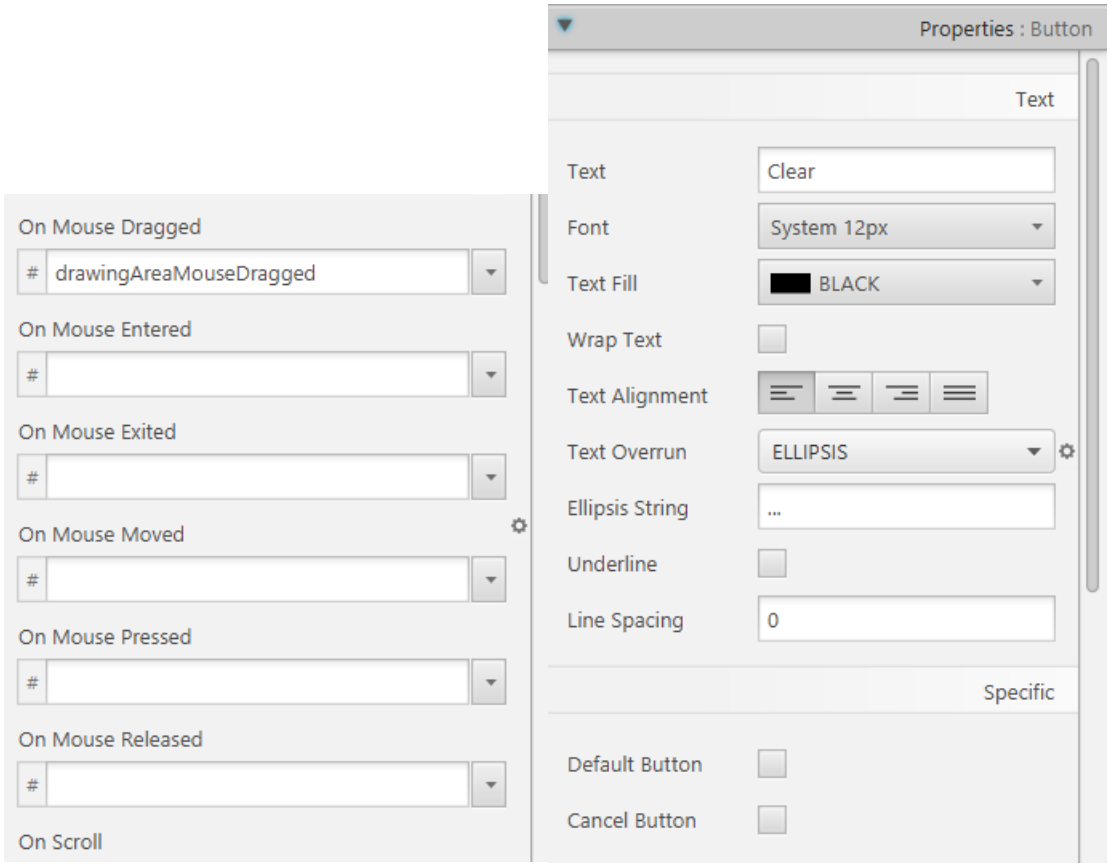
Position

Layout X 0

Layout Y 0

Code : VBox


BorderPane
insert TOP
VBox
TitledPane
VBox
RadioButton Pen
RadioButton Eraser
Button Clear
Pane
insert RIGHT
insert BOTTOM
JavaFX CSS
Style
-fx-background-color
white
Style Class
Stylesheets
Id
Extras
Blend Mode
SRC_OVER
Layout : Pane


On Mouse Dragged
drawingAreaMouseDragged
On Mouse Entered
On Mouse Exited
On Mouse Moved
On Mouse Pressed
On Mouse Released
On Scroll
Properties : Button
Text
Text
Clear
Font
System 12px
Text Fill
BLACK
Wrap Text
Text Alignment
Text Overrun
ELLIPSIS
Ellipsis String
...
Underline
Line Spacing
0
Specific
Default Button
Cancel 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: 75.2

Height: 25.6

Position

Layout X: 0

Layout Y: 0

Transforms

Code : Buttor

Identity

fx:id:

Main

On Action: clearButtonPressed

DragDrop

On Drag Detected:

On Drag Done:

On Drag Dropped:

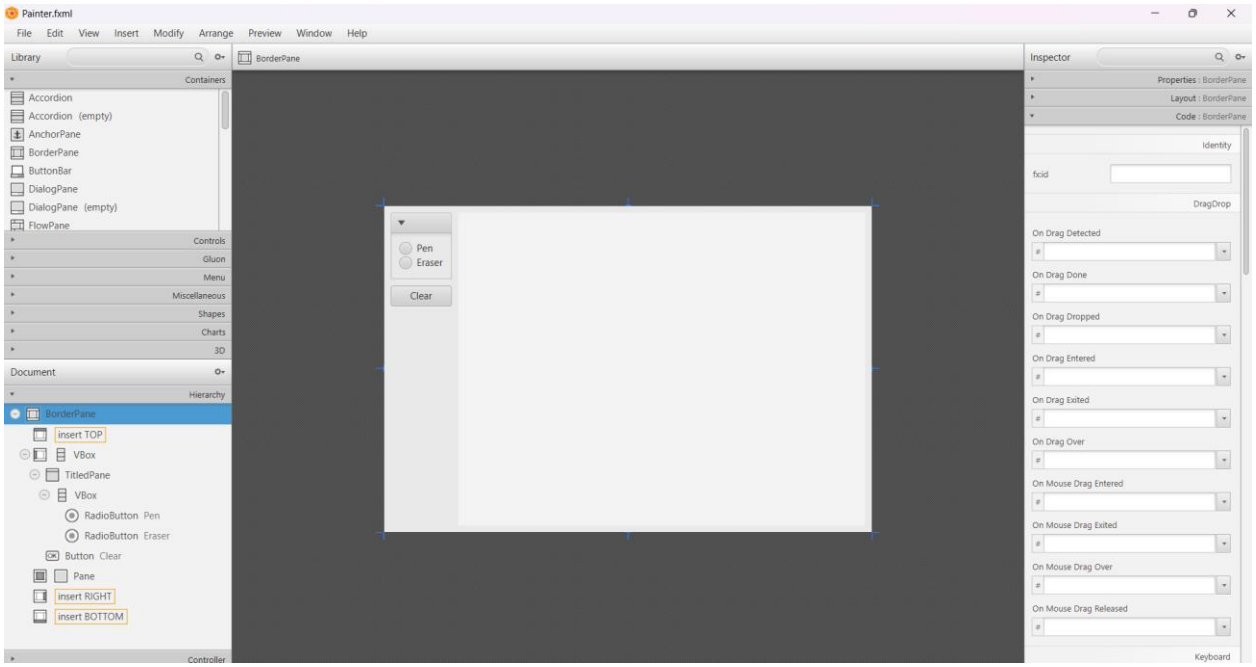
On Drag Entered:

On Drag Exited:

On Drag Over:

On Mouse Drag Entered:

On Mouse Drag Exited

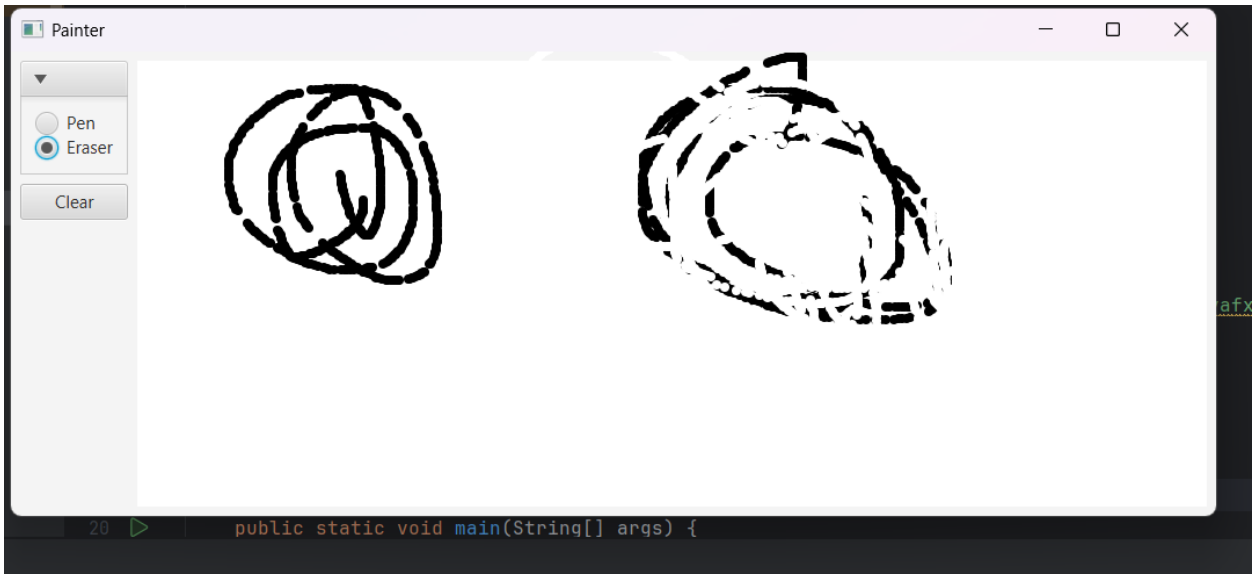



```
<?xml version="1.0" encoding="UTF-8"?>

<?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="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/23.0.
    <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>
```

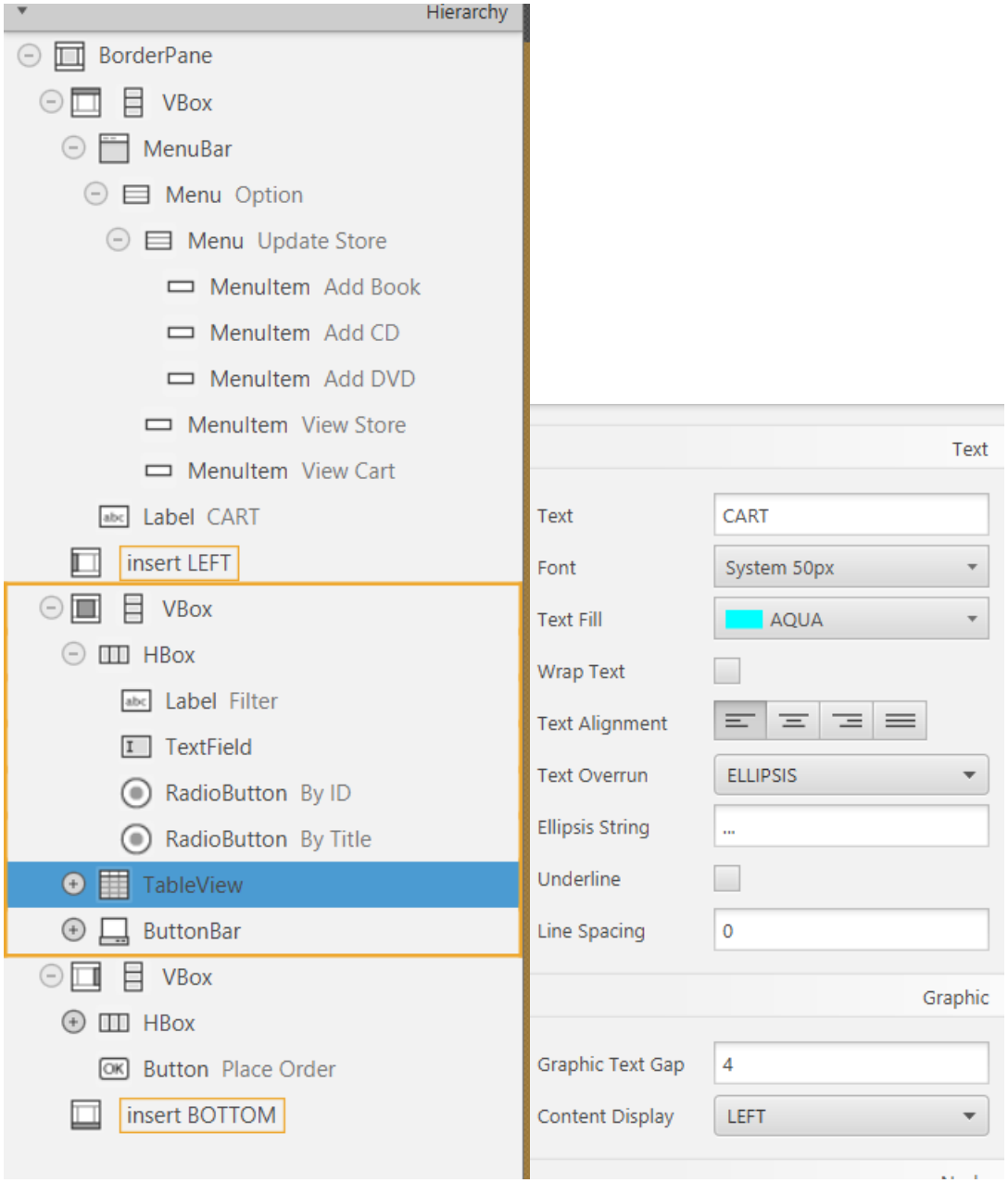
```java
package hust.soict.cyber.javafx;

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

public class Painter extends Application {

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

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

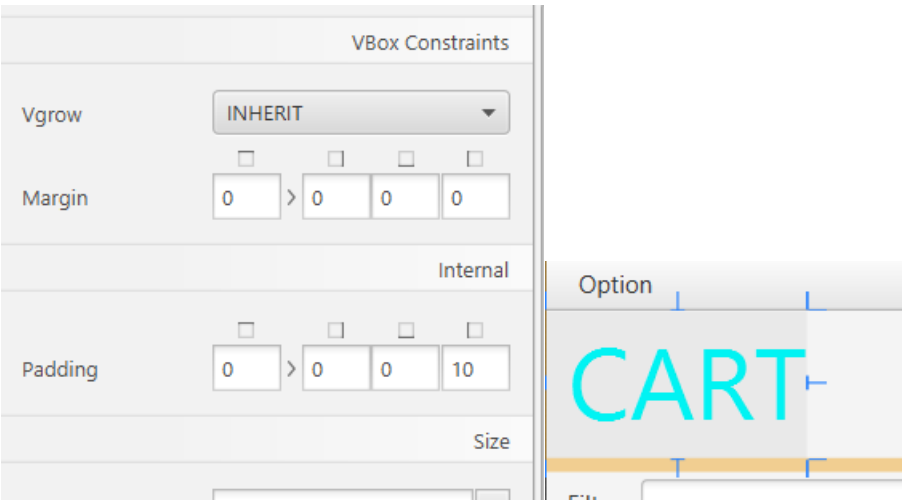
```java
public class PainterController {  1 usage
    private int color;  4 usages

    @FXML
    private Pane drawingAreaPane;

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

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

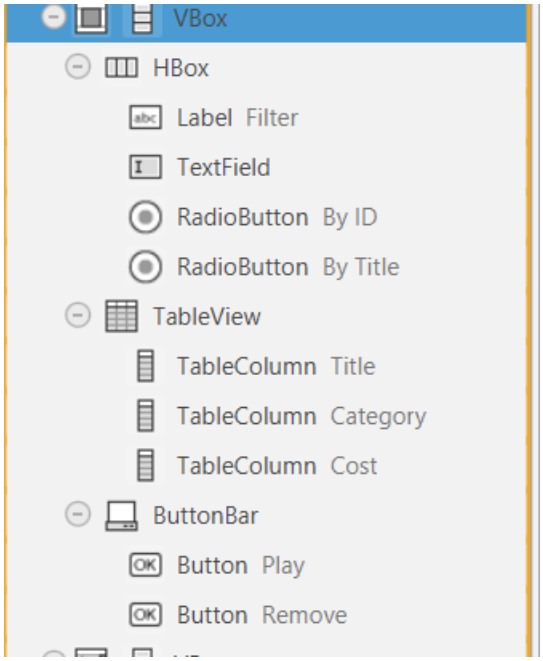
```java
    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        if (color == 1 && event.getX()>=0) {
            Circle newCircle = new Circle(event.getX(), event.getY(), radius: 4.0, Color.BLACK);
            drawingAreaPane.getChildren().add(newCircle);
        }else if(color == 0 &&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

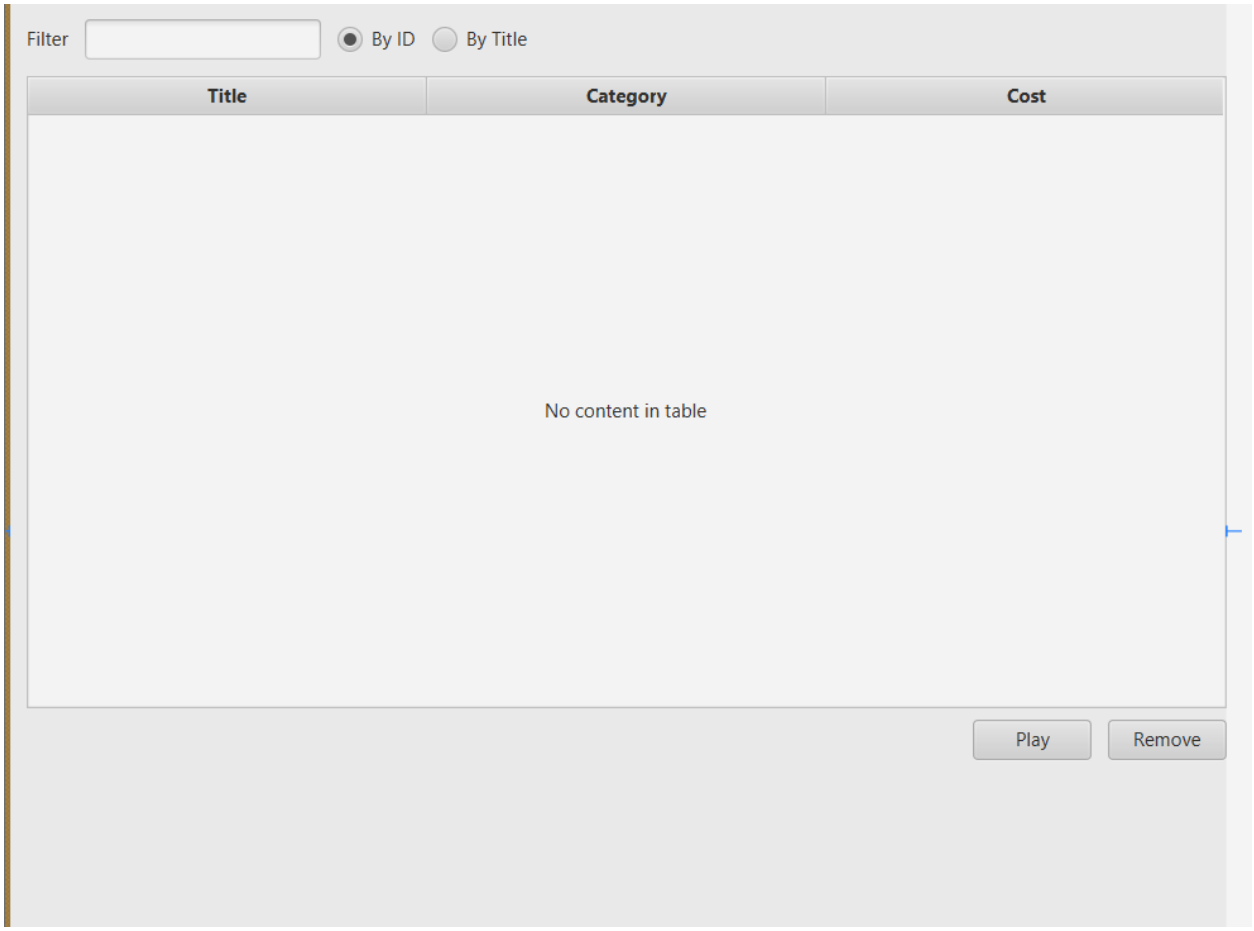


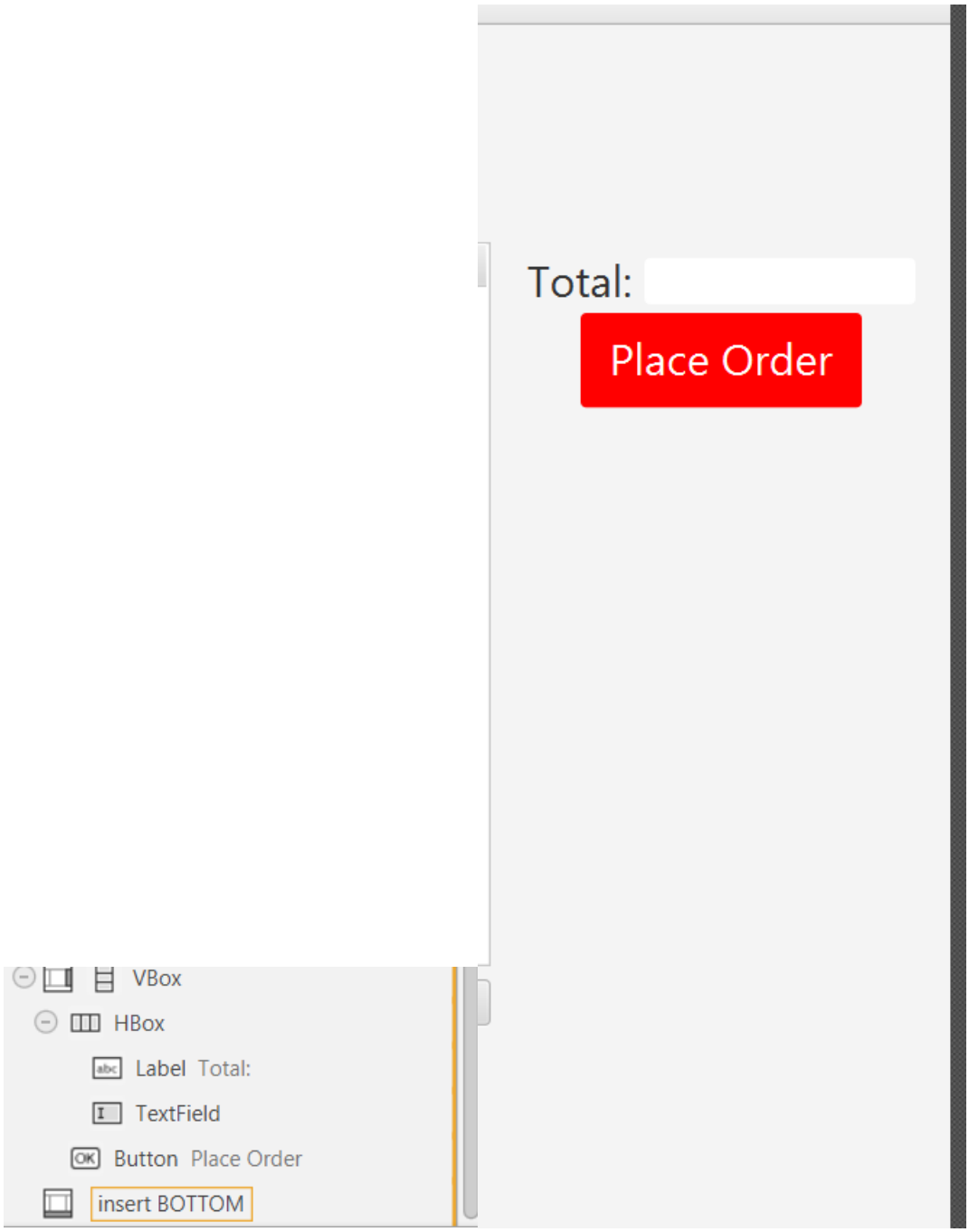
5. Setting up the View Cart Screen with ScreenBuilder


Hierarchy
BorderPane
VBox
MenuBar
Menu Option
Menu Update Store
MenuItem Add Book
MenuItem Add CD
MenuItem Add DVD
MenuItem View Store
MenuItem View Cart
Label CART
insert LEFT
VBox
HBox
Label Filter
TextField
RadioButton By ID
RadioButton By Title
TableView
ButtonBar
VBox
HBox
Button Place Order
insert BOTTOM
Text
Text
CART
Font
System 50px
Text Fill
AQUA
Wrap Text
Text Alignment
Text Overrun
ELLIPSIS
Ellipsis String
...
Underline
Line Spacing
0
Graphic
Graphic Text Gap
4
Content Display
LEFT


VBox Constraints
Vgrow
INHERIT
Margin
0
0
0
0
Internal
Padding
0
0
0
10
Size
Option
CART


VBox
HBox
Label Filter
TextField
RadioButton By ID
RadioButton By Title
TableView
TableColumn Title
TableColumn Category
TableColumn Cost
ButtonBar
Button Play
Button Remove


Filter
By ID
By Title
Title
Category
Cost
No content in table
Play
Remove


Total:
Place Order
VBox
HBox
Label Total:
TextField
Button Place Order
insert BOTTOM

6. Integrating JavaFX into Swing application – The JFXPanel class

```java
public class CartScreen extends JFrame {  1 usage
    private Cart cart;  1 usage
    private Store store;  no usages


    public CartScreen(Cart cart, Store store) {  1 usage
        super();
        this.cart = cart;

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

        this.setTitle("Cart");
        this.setVisible(true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/hust/soict/cybersec/aims/screen/cart.fxml"));
                    CartScreenController controller = new CartScreenController(cart, store);
                    loader.setController(controller);
                    Parent root = (Parent) loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
        this.setSize( width: 1024,  height: 768);
    }
}
```

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

Identity

fx:id tblMedia

Main

On Sort

#

DragDrop

Identity

fx:id colMediaTitle

Edit

On Edit Start

#

On Edit Commit

#

On Edit Cancel

#

Identity

fx:id colMediaCategory

Edit

On Edit Start

#

On Edit Commit

#

On Edit Cancel

#

Identity

fx:id colMediaCost

Edit

On Edit Start

#

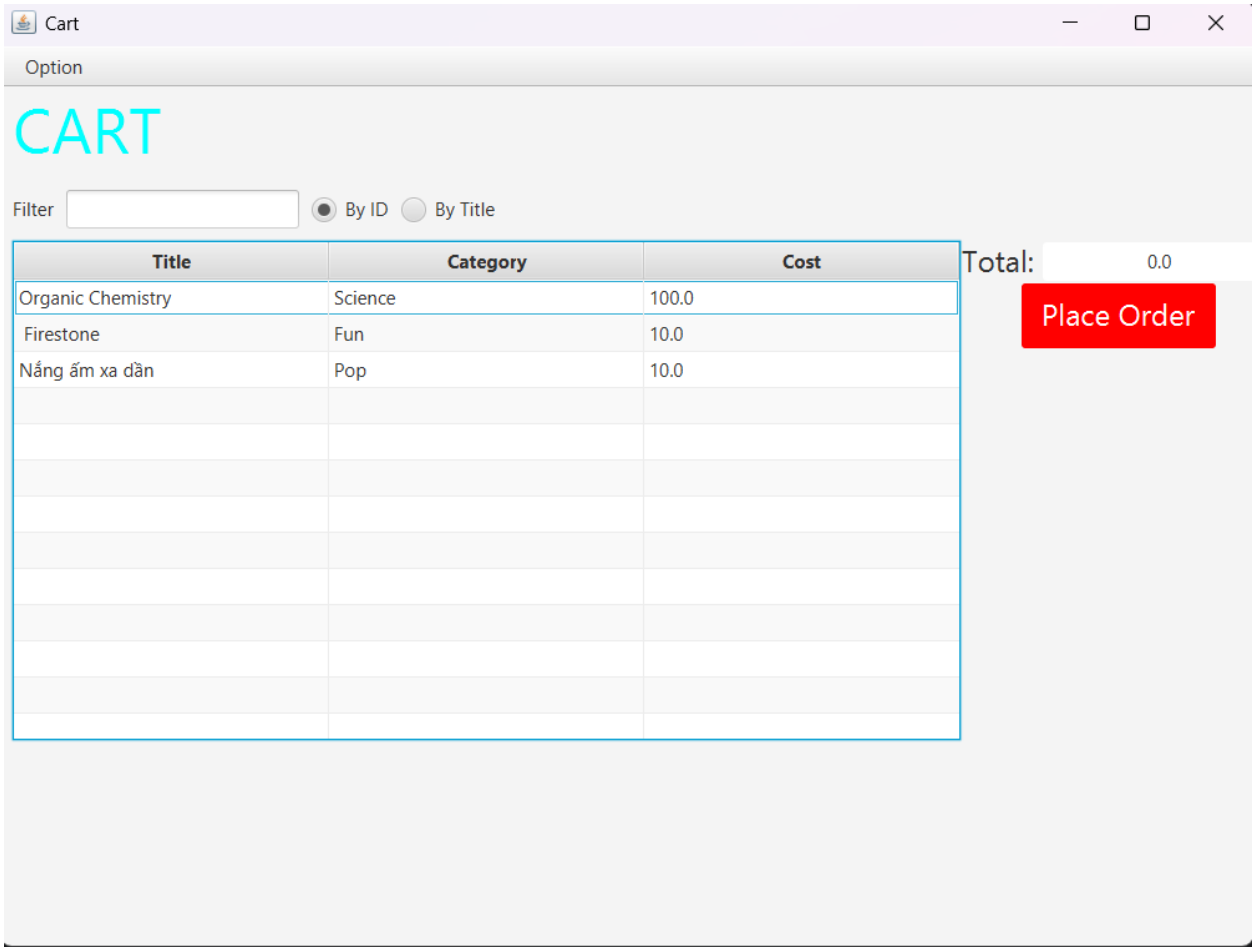
On Edit Commit

#

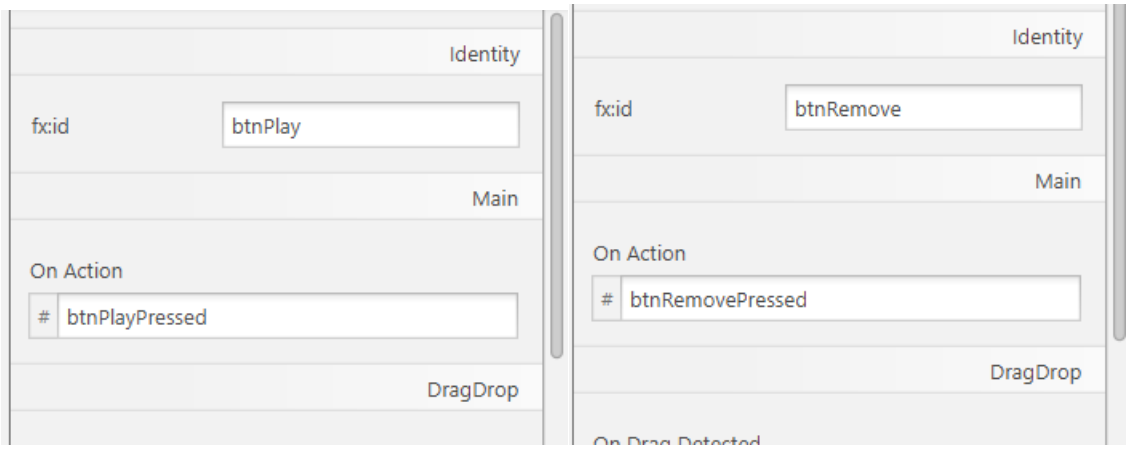
On Edit Cancel

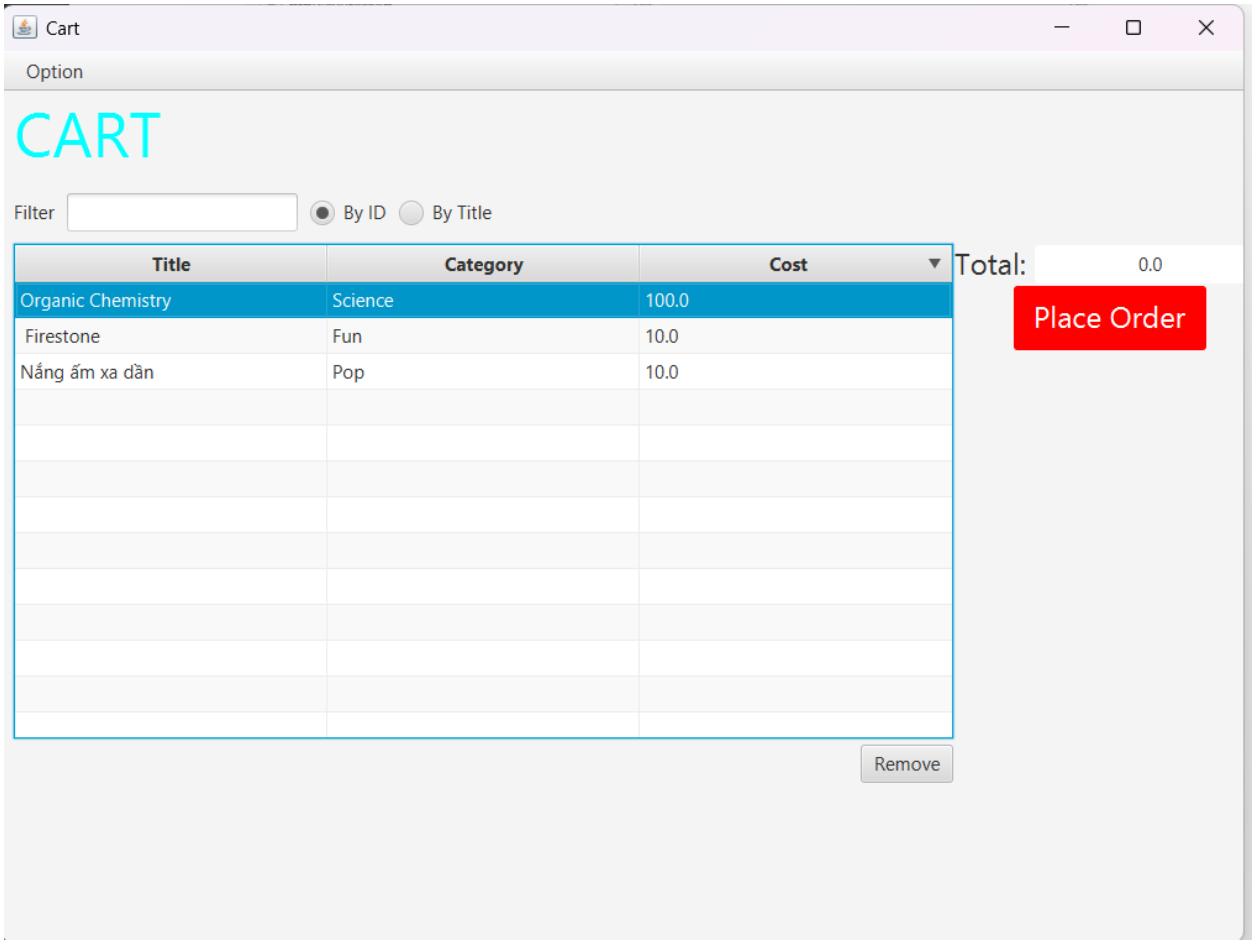
#

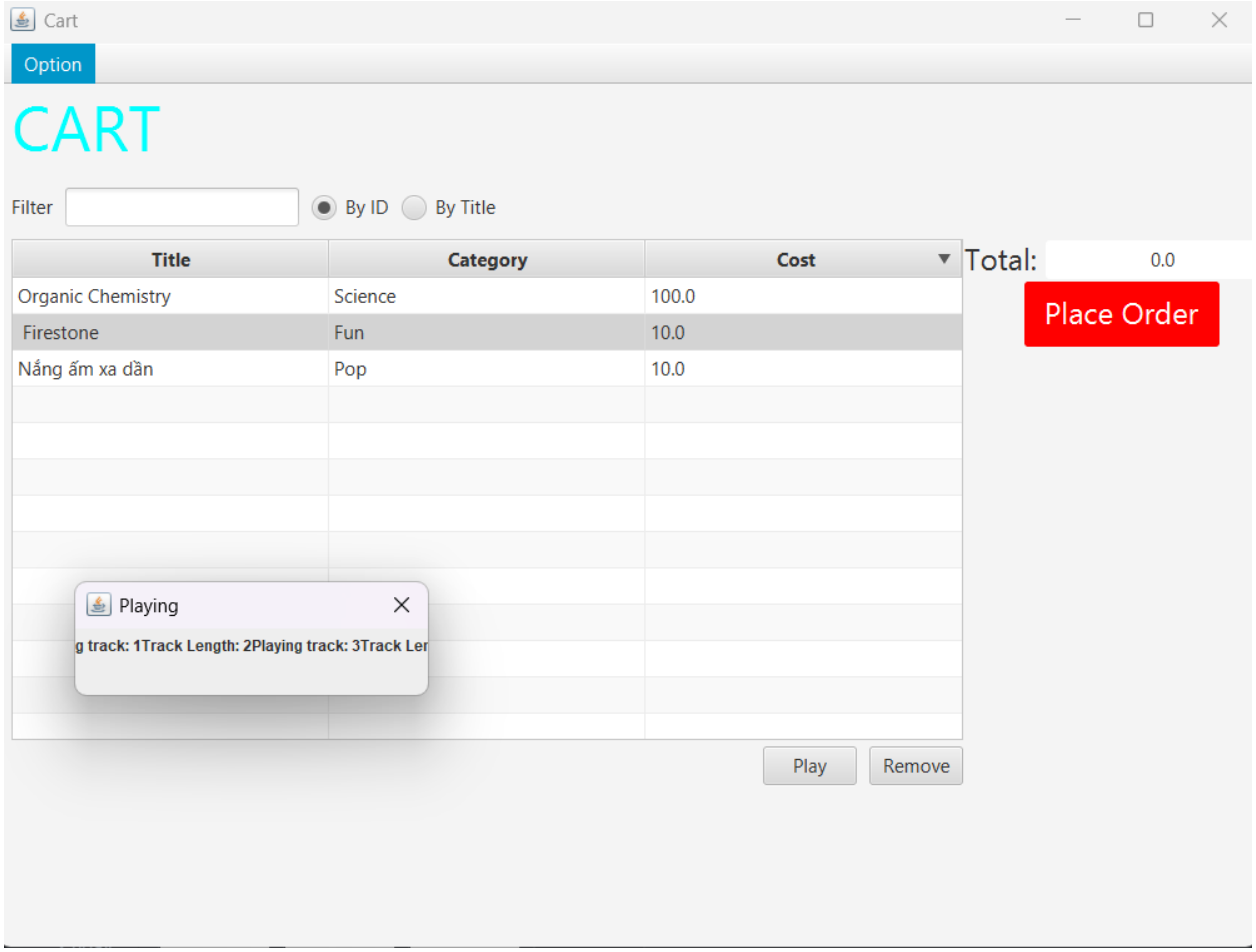
- Update itemsOrdered to ObservableList



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



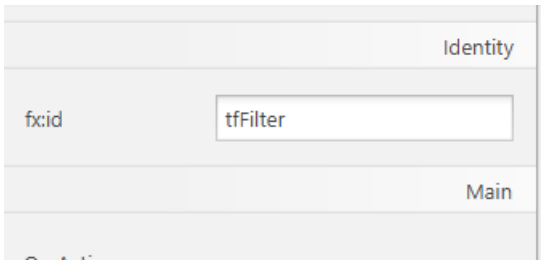

Cart
Option
CART
Filter
By ID
By Title
Title
Category
Cost
Organic Chemistry
Science
100.0
Firestone
Fun
10.0
Nắng ấm xa dần
Pop
10.0
Total:
0.0
Place Order
Remove

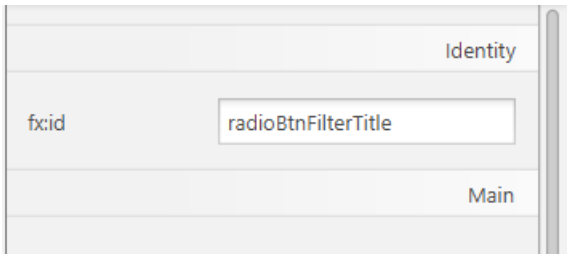


9. Deleting a media

```
@FXML   // remove button  no usages
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    totalCost = cart.totalCost();
    tfTotalCost.setText(totalCost + ""); // reset the total cost
}
```

10. Filter items in cart – FilteredList

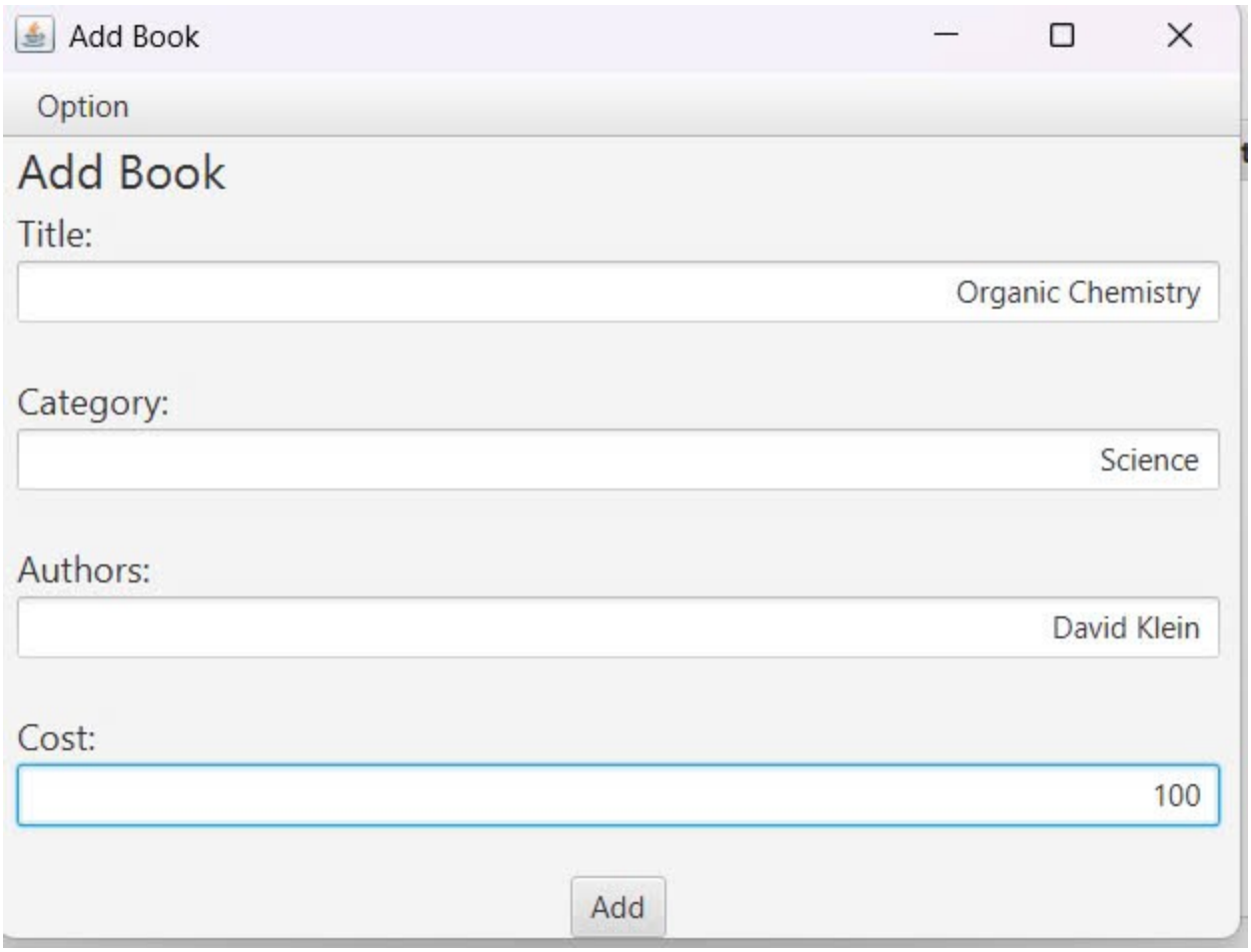


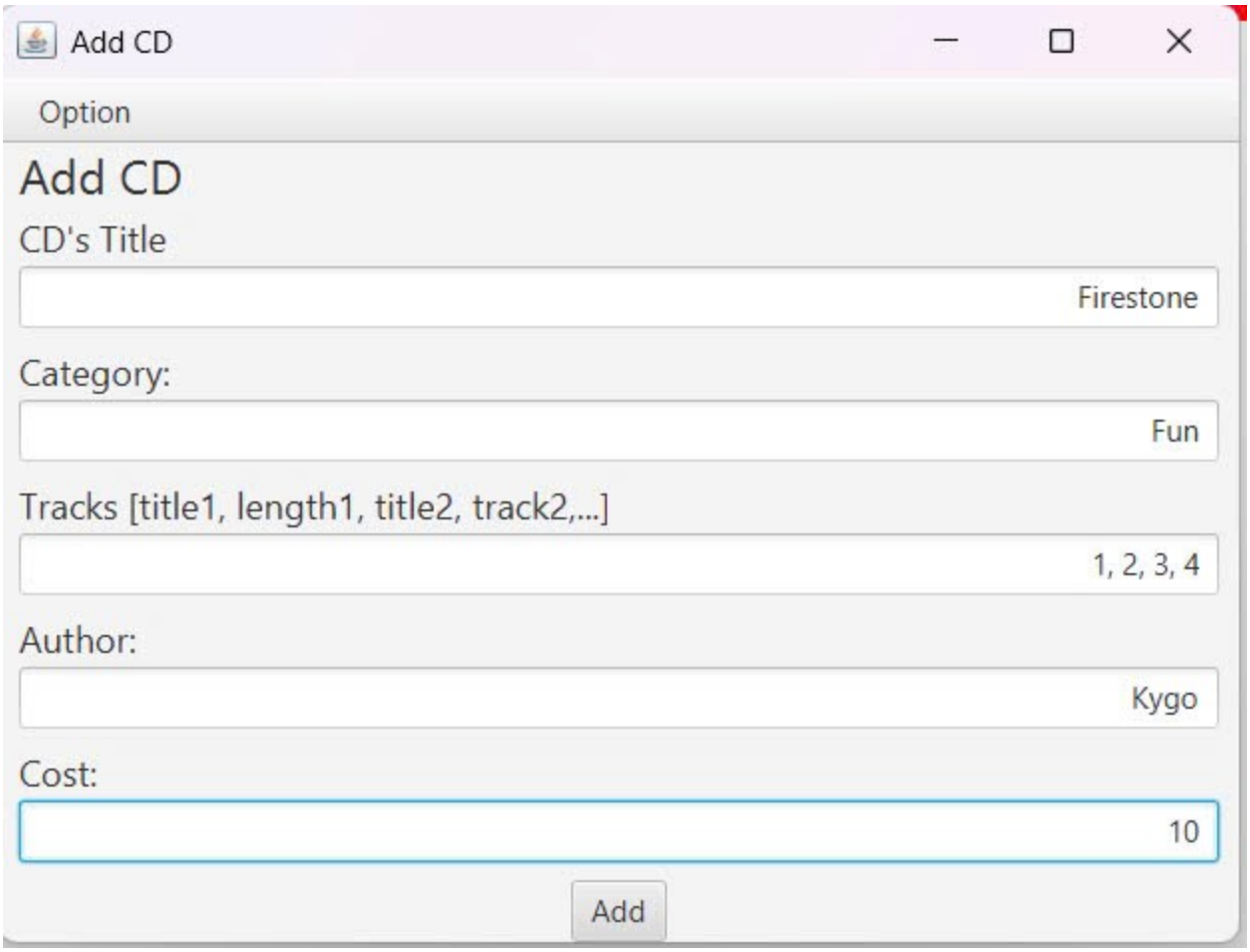


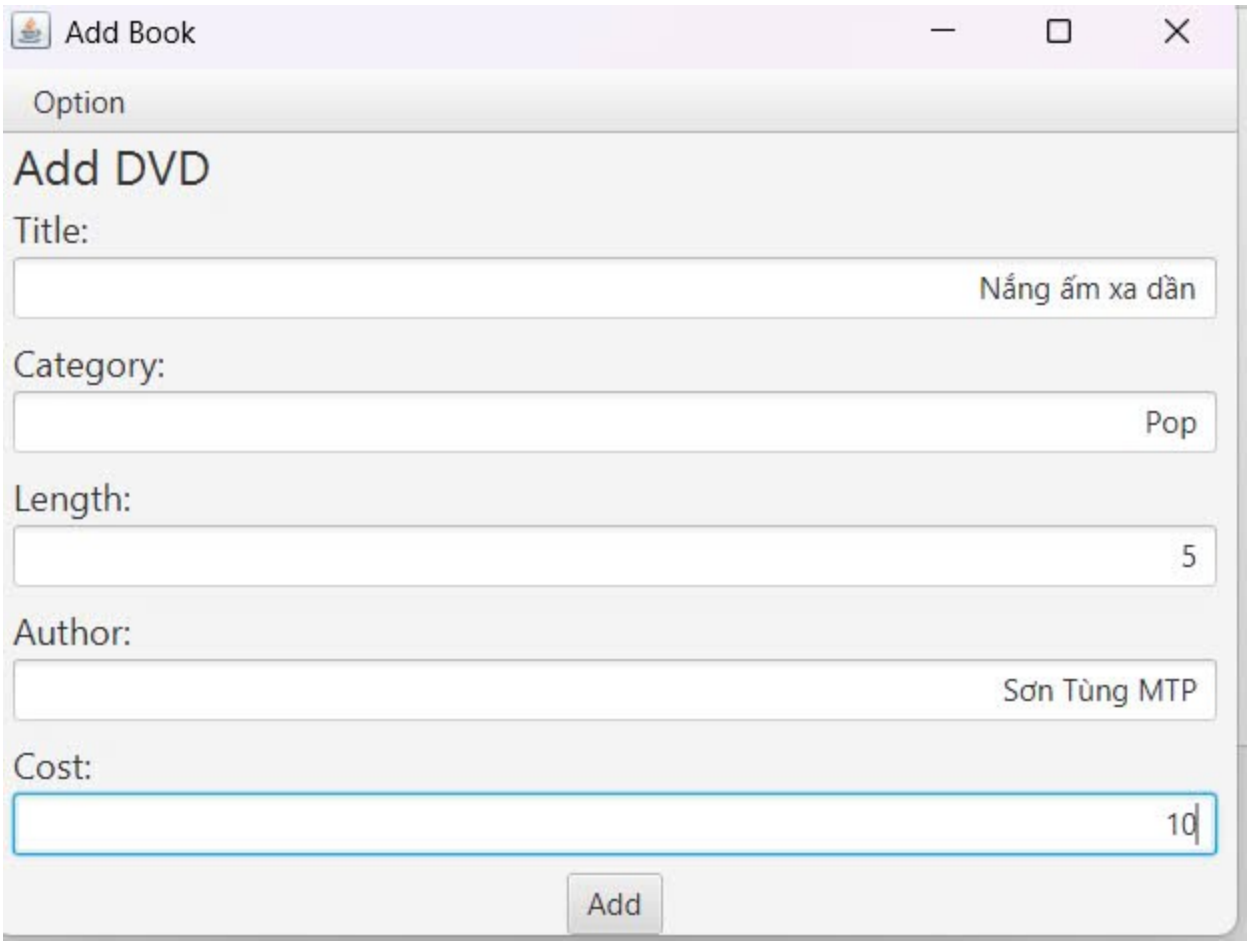

```java
tfFilter.textProperty().addListener(new ChangeListener<String>() {
    @Override
    public void changed (ObservableValue<? extends String> observable, String oldValue, String newValue) {
        showFilteredMedia(newValue);
    }
});
```

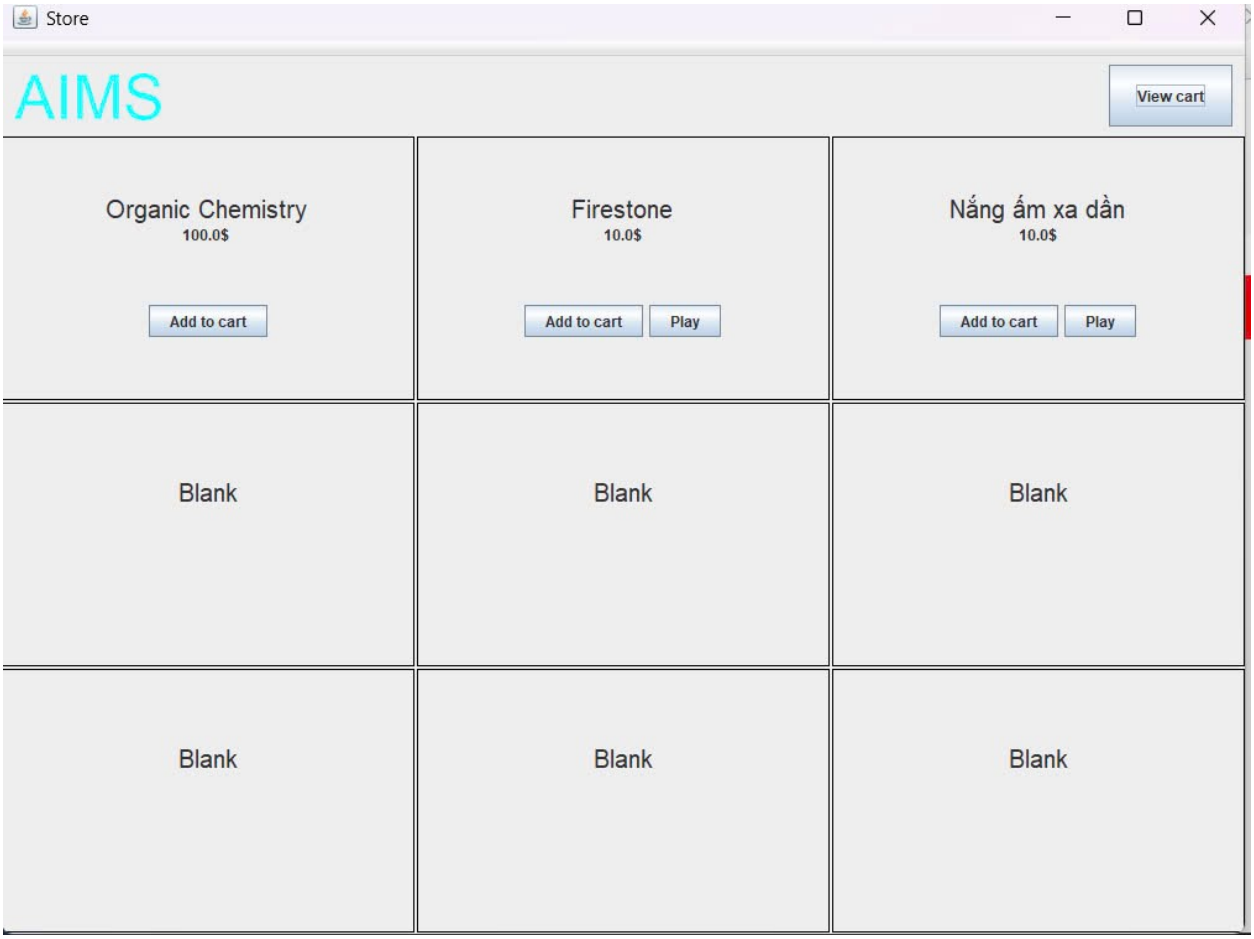
```java
void showFilteredMedia(String newValue) {  1 usage
    String valueType;
    if (radioBtnFilterID.isSelected()) {
        valueType = "id";
    } else {
        valueType = "title";
    }

    FilteredList<Media> list = new FilteredList<>(cart.getItemsOrdered(), predicate: null);
    list.setPredicate(media -> media.filterByProperty(newValue, valueType));

    if (cart.getItemsOrdered() != null) {
        tblMedia.setItems(list);
    }
}
```

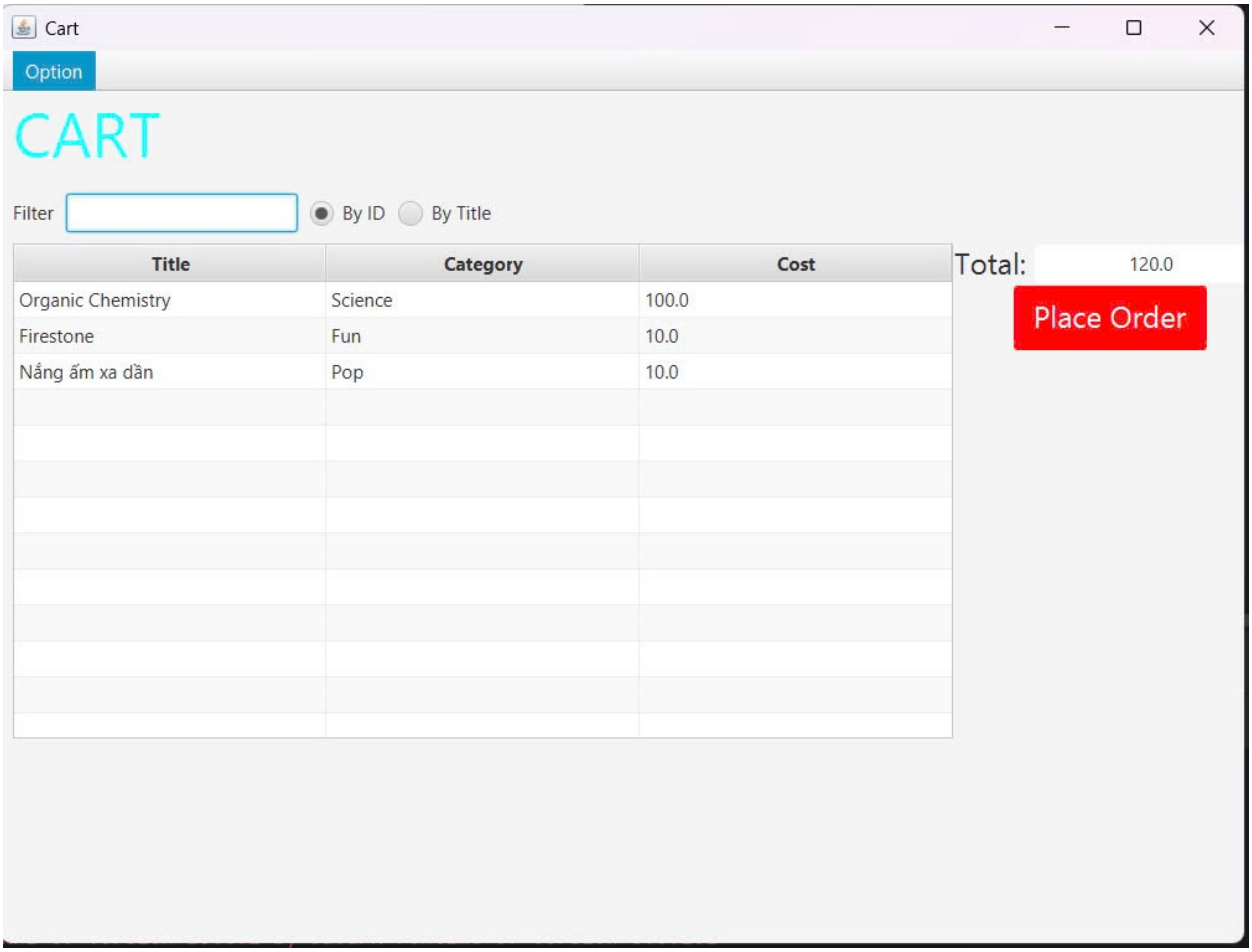
11. Complete the Aims GUI application


Add Book
Option
Add Book
Title:
Organic Chemistry
Category:
Science
Authors:
David Klein
Cost:
100
Add

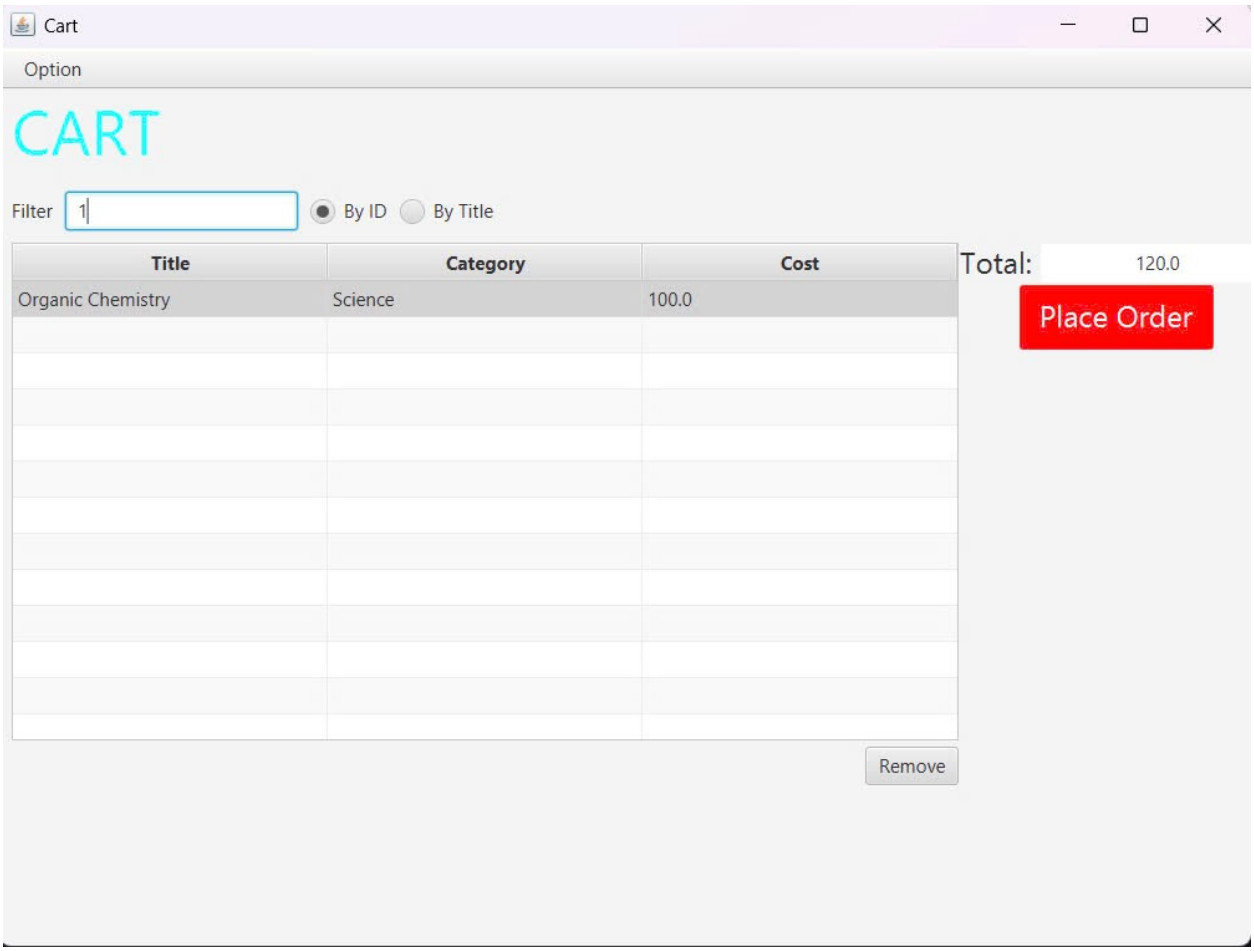

Add CD
Option
Add CD
CD's Title
Firestone
Category:
Fun
Tracks [title1, length1, title2, track2,...]
1, 2, 3, 4
Author:
Kygo
Cost:
10
Add

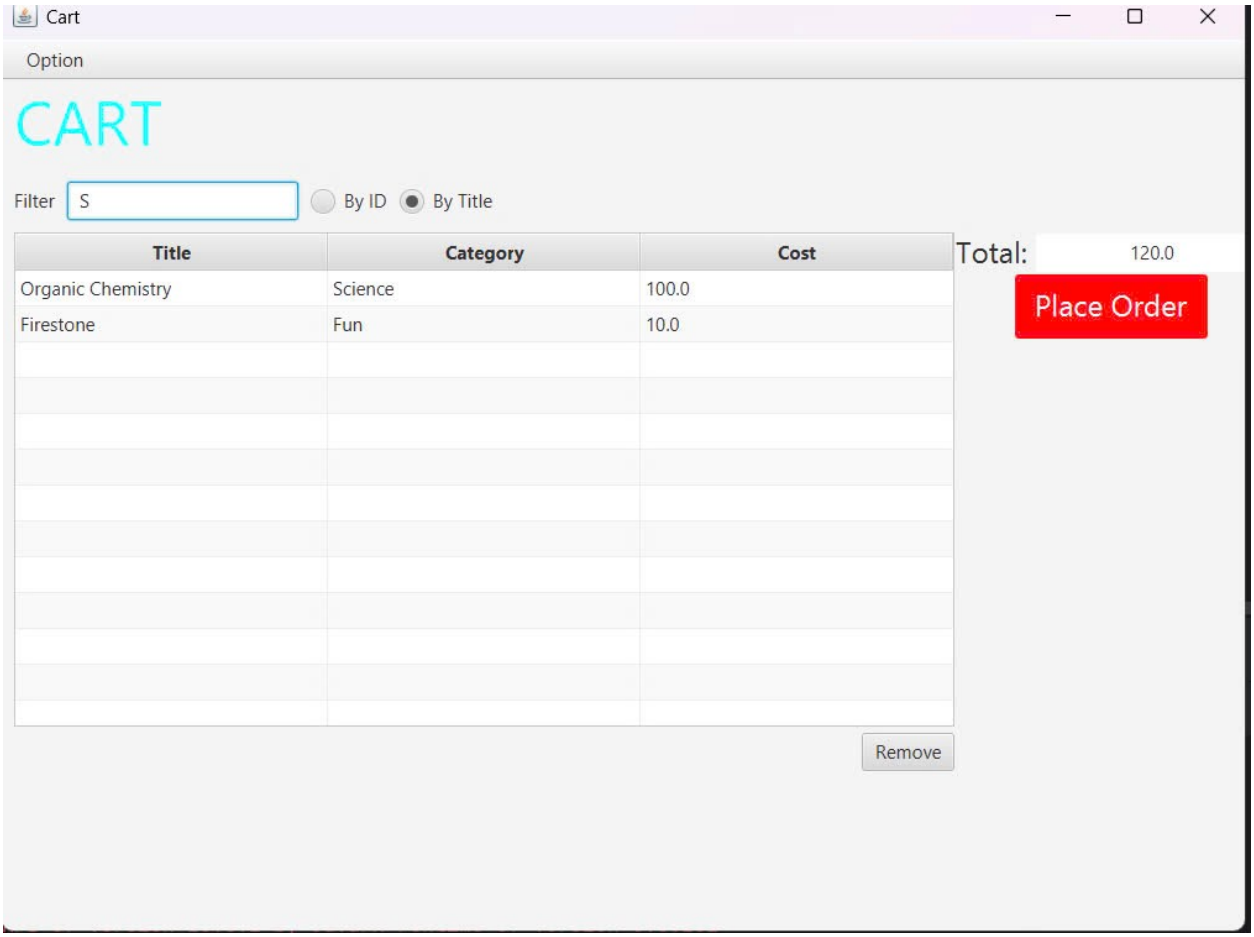

Add Book
Option
Add DVD
Title:
Nắng ấm xa dần
Category:
Pop
Length:
5
Author:
Sơn Tùng MTP
Cost:
10
Add


Store
AIMS
View cart
Organic Chemistry
100.0$
Add to cart
Firestone
10.0$
Add to cart
Play
Nắng ấm xa dần
10.0$
Add to cart
Play
Blank
Blank
Blank
Blank
Blank
Blank


Cart
Option
CART
Filter
By ID
By Title
Title
Category
Cost
Organic Chemistry
Science
100.0
Firestone
Fun
10.0
Nắng ấm xa dần
Pop
10.0
Total:
120.0
Place Order

- Search


Cart
Option
CART
Filter
1
By ID
By Title
Title
Category
Cost
Organic Chemistry
Science
100.0
Total:
120.0
Place Order
Remove



AddItemToStoreScreen code:

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

import javax.swing.JFrame;

import hust.soict.cybersec.aims.store.Store;
import hust.soict.cybersec.aims.media.Media;

public class AddItemToStore extends JFrame {  3 usages  3 inheritors
    protected Store store;
    protected String type;  2 usages

    public AddItemToStore (Store store, String mediaType) {  3 usages
        super();
        this.store = store;
        this.type = mediaType;
    }

    public void setType(String newType) {  no usages
        this.type = newType;
    }

    public Store getUpdatedStore(Media media) {  no usages
        this.store.addMedia(media);
        return store;
    }

}
```

AddBookToStoreScreen code:

```java
public class AddBookToStoreScreen extends AddItemToStore {  1 usage
    public AddBookToStoreScreen(Store store) {  1 usage
        super(store, mediaType: "Book");

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

        this.setTitle("Add Book");
        this.setVisible(true);

        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/hust/soict/cybersec/aims/screen/addBook.fxml"));
                    AddBookToScreenController controller = new AddBookToScreenController(store);
                    loader.setController(controller);
                    Parent root = (Parent) loader.load();
                    fxPanel.setScene (new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });

        this.setSize( width: 640,  height: 480);
    }

    public Store getUpdatedStore() {  no usages
        return store;
    }
```

```java
private class AddBookToScreenController {  2 usages
    @FXML private TextField titleBook;  2 usages
    @FXML private TextField authorsBook;  2 usages
    @FXML private TextField categoryBook;  2 usages
    @FXML private TextField costBook;  2 usages
    @FXML private Button btnAddBook;  1 usage


    public AddBookToScreenController(Store _store) {  1 usage
        super();
        store = _store;
    }

    @FXML  no usages
    private void intitialize() {
        btnAddBook.setVisible(true);
        titleBook.setVisible(true);
        authorsBook.setVisible(true);
        categoryBook.setVisible(true);
        costBook.setVisible(true);

    }

    @FXML  no usages
    private void btnAddBookPressed() {
        String title = titleBook.getText();
        String[] authors = authorsBook.getText().split( regex: ",");
        String category = categoryBook.getText();
        try {
            Float cost = Float.parseFloat(costBook.getText());
            if (cost >= 0.0) {
```

AddCompactDiscToStoreScreen code:

```java
public class AddCompactDiscToStoreScreen extends AddItemToStore {  1 usage
    public AddCompactDiscToStoreScreen(Store store) {  1 usage

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

        this.setTitle("Add CD");
        this.setVisible(true);

        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/hust/soict/cybersec/aims/screen/addCD.fxml"));
                    AddCompactDiscToScreenController controller = new AddCompactDiscToScreenController(store);
                    loader.setController(controller);
                    Parent root = (Parent) loader.load();
                    fxPanel.setScene (new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
        this.setSize( width: 640,  height: 480);
    }

    private class AddCompactDiscToScreenController {  2 usages

        @FXML private TextField titleCD;  2 usages
        @FXML private TextField authorCD;  2 usages
        @FXML private TextField categoryCD;  2 usages
        @FXML private TextField costCD;  2 usages
```

```java
public AddCompactDiscToScreenController(Store _store) {  1 usage
    super();
    store = _store;
}

@FXML  no usages
private void initialize() {
    btnAddCD.setVisible(true);
    titleCD.setVisible(true);
    authorCD.setVisible(true);
    categoryCD.setVisible(true);
    costCD.setVisible(true);
    tracksCD.setVisible(true);

}

@FXML  no usages
private void btnAddCDPressed() {
    String title = titleCD.getText();
    String author = authorCD.getText();
    String category = categoryCD.getText();
    try {

        ArrayList<Track> TrackListCD = new ArrayList<Track>();

        Float cost = Float.parseFloat(costCD.getText());
        String[] tracks = tracksCD.getText().split( regex: ", ");
        Integer length = 0;
        for (int i = 0; i <= tracks.length - 1; i += 2) {
            String track_title = tracks[i];
```

```java
private void btnAddCDPressed() {
    String category = categoryCD.getText();
    try {

        ArrayList<Track> TrackListCD = new ArrayList<Track>();

        Float cost = Float.parseFloat(costCD.getText());
        String[] tracks = tracksCD.getText().split( regex: ", ");
        Integer length = 0;
        for (int i = 0; i <= tracks.length - 1; i += 2) {
            String track_title = tracks[i];
            Integer track_length = Integer.parseInt(tracks[i+1].replace( target: " ",  replacement: ""));
            if (track_length <= 0){
                throw new IllegalArgumentException(); // exception handle
            }

            length += track_length;
            TrackListCD.add(new Track(track_title, track_length));
        }

        Integer curr_id = store.getItemsInStore().size() + 1;
        CompactDisc cd = new CompactDisc(curr_id, title, category, length , author, cost);
        for (Track t: TrackListCD) {
            cd.addTrack(t);
        }
        store.addMedia(cd);

        setVisible(false);

    } catch (IllegalArgumentException iae) {
```

AddDigitalVideoDiscToStoreScreen code:

```java
public class AddDigitalVideoDiscToStoreScreen extends AddItemToStore {  1 usage
    public AddDigitalVideoDiscToStoreScreen(Store store) {  1 usage
        super(store, mediaType: "DVD");

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

        this.setTitle("Add Book");
        this.setVisible(true);

        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/hust/soict/cybersec/aims/screen/addDVD.fxml"));
                    AddDigitalVideoDiscToScreenController controller = new AddDigitalVideoDiscToScreenController(store);
                    loader.setController(controller);
                    Parent root = (Parent) loader.load();
                    fxPanel.setScene (new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
        this.setSize( width: 640,  height: 480);
    }

    private class AddDigitalVideoDiscToScreenController {  2 usages

        @FXML private TextField titleDVD;  2 usages
        @FXML private TextField authorDVD;  2 usages
        @FXML private TextField categoryDVD;  2 usages
```

```java
        @FXML private TextField costDVD;  2 usages
        @FXML private TextField lengthDVD;  2 usages
        @FXML private Button btnAddDVD;  1 usage


        public AddDigitalVideoDiscToScreenController(Store _store) {  1 usage
            super();
            store = _store;
        }

        @FXML  no usages
        private void initialize() {
            btnAddDVD.setVisible(true);
            titleDVD.setVisible(true);
            authorDVD.setVisible(true);
            categoryDVD.setVisible(true);
            costDVD.setVisible(true);
            lengthDVD.setVisible(true);

        }

        @FXML  no usages
        private void btnAddDVDPressed() {
            String title = titleDVD.getText();
            String author = authorDVD.getText();
            String category = categoryDVD.getText();
            try {
                Float cost = Float.parseFloat(costDVD.getText());
                if (cost <= 0.0){
                    throw new IllegalArgumentException();
```

```java
    @FXML  no usages
    private void btnAddDVDPressed() {
        String title = titleDVD.getText();
        String author = authorDVD.getText();
        String category = categoryDVD.getText();
        try {
            Float cost = Float.parseFloat(costDVD.getText());
            if (cost <= 0.0){
                throw new IllegalArgumentException();
            }
            Integer length = Integer.parseInt(lengthDVD.getText());
            if (length <= 0) {
                throw new IllegalArgumentException();
            }
            Integer curr_id = store.getItemsInStore().size() + 1;
            store.addMedia(new DigitalVideoDisc(curr_id, title, category, length , author, cost));

            setVisible(false);
        } catch (IllegalArgumentException iae) {
            JOptionPane.showMessageDialog( parentComponent: null,  message: "Error: Wrong format of input");;
        }
    }
  }
}
```

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

```java
public void addMedia(Media m) throws LimitExceededException {
    if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
        itemsOrdered.add(m);

    } else {
        throw new LimitExceededException("ERROR");
    }
}
```

13. Create a class which inherits from Exception

```java
package hust.soict.cybersec.aims.exception;

public class PlayerException extends Exception {  21 usages

    public PlayerException() {  no usages
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message) {  3 usages
        super(message);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(Throwable cause) {  no usages
        super(cause);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message, Throwable cause) {  no usages
        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
    }

}
```

CD

```java
    public String play() throws PlayerException{  2 usages  new *
        if (this.getLength() > 0) {
            String playing = "";
            for (Track track: tracks) {
                try {
                    playing = playing + track.play() + '\n';
                } catch (PlayerException e) {
                    throw e;
                }
            } return playing;
        } else {
            throw new PlayerException("ERROR: CD length is non-positive");
        }
    }
```

DVD

```java
    public String play() throws PlayerException {  2 usages  new *
        if (this.getLength() > 0) {
            return "Playing DVD: " + this.getTitle() + "\n DVD Length: " + this.getLength();
        } else {
            System.err.println("ERROR: DVD length is non-positive");
            throw new PlayerException("ERROR: DVD length is non-positive");
        }
    }

}
```

Track

```java
    public String play () throws PlayerException{  2 usages  new *
        if (this.getLength() > 0){
            return "Playing track: " + this.getTitle() + "\n" +"Track Length: " + this.getLength();
        } else {
            System.err.println("Error: Track length is non-positive");
            throw new PlayerException("ERROR: Track length is non-positive");
        }
    }
```

14. Update the Aims class

```java
        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.");
                }
```

```java
    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
public boolean equals(Object o) {  Kainic0804 *
    if (o instanceof Media) {
        Media m = (Media) o;
        return this.getTitle().equals(m.getTitle());
    } else {
        return false;
    }
}
```

17. Update Aims class diagram