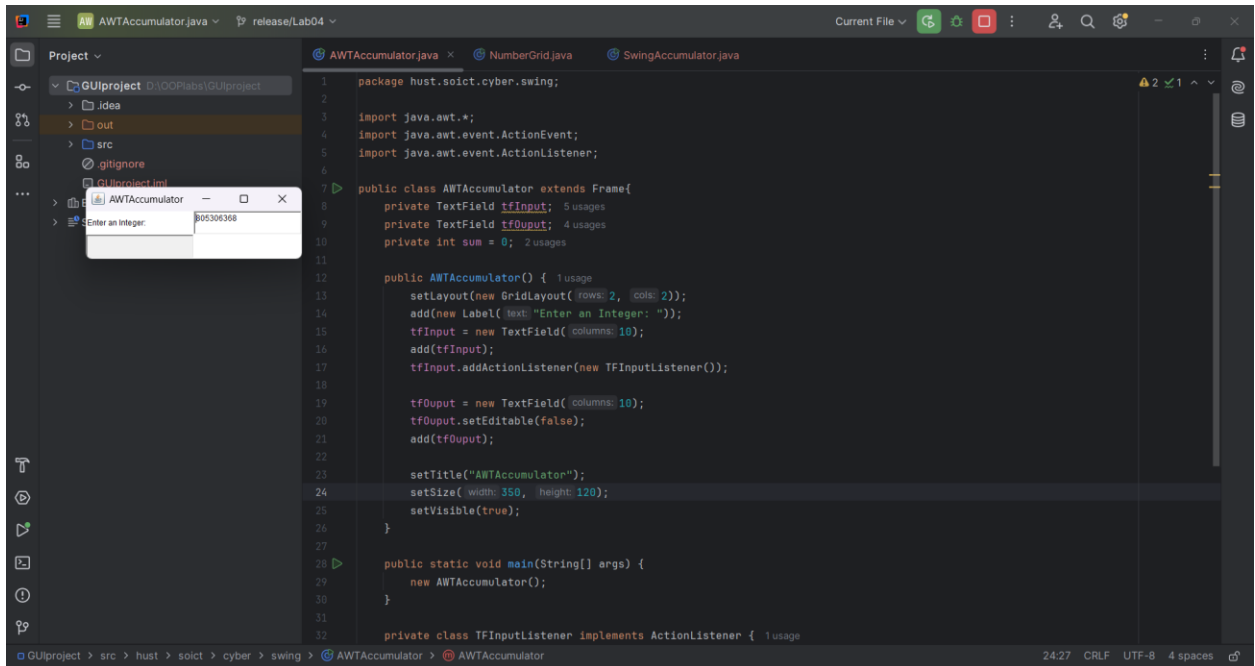


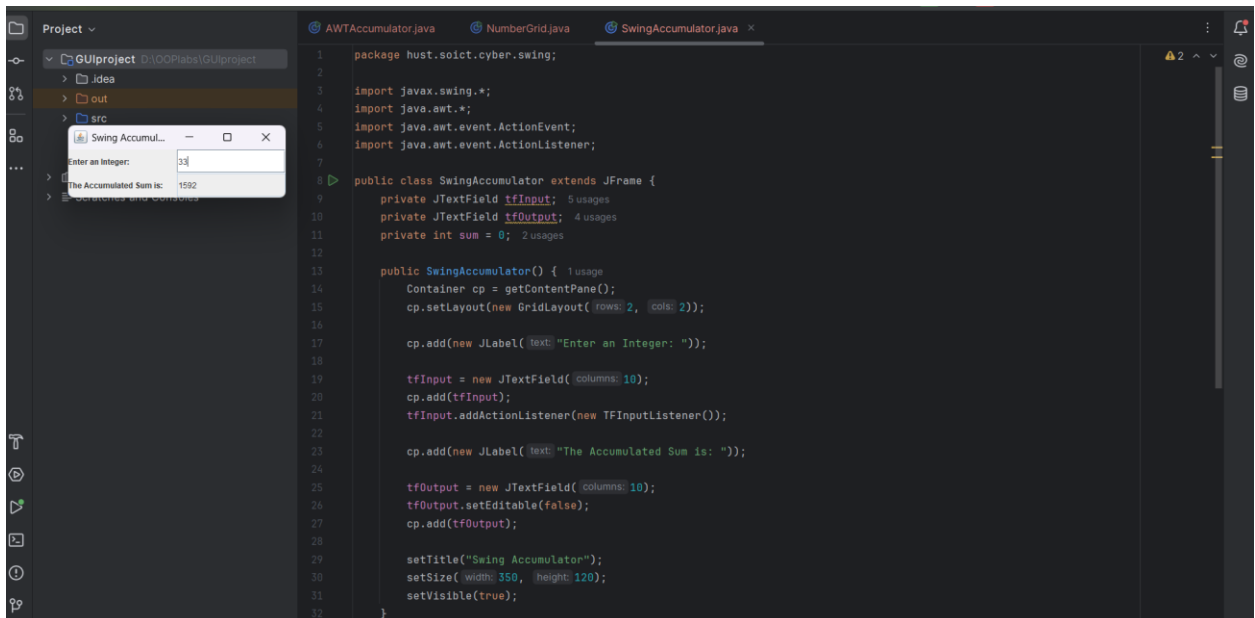
Report

1. Swing

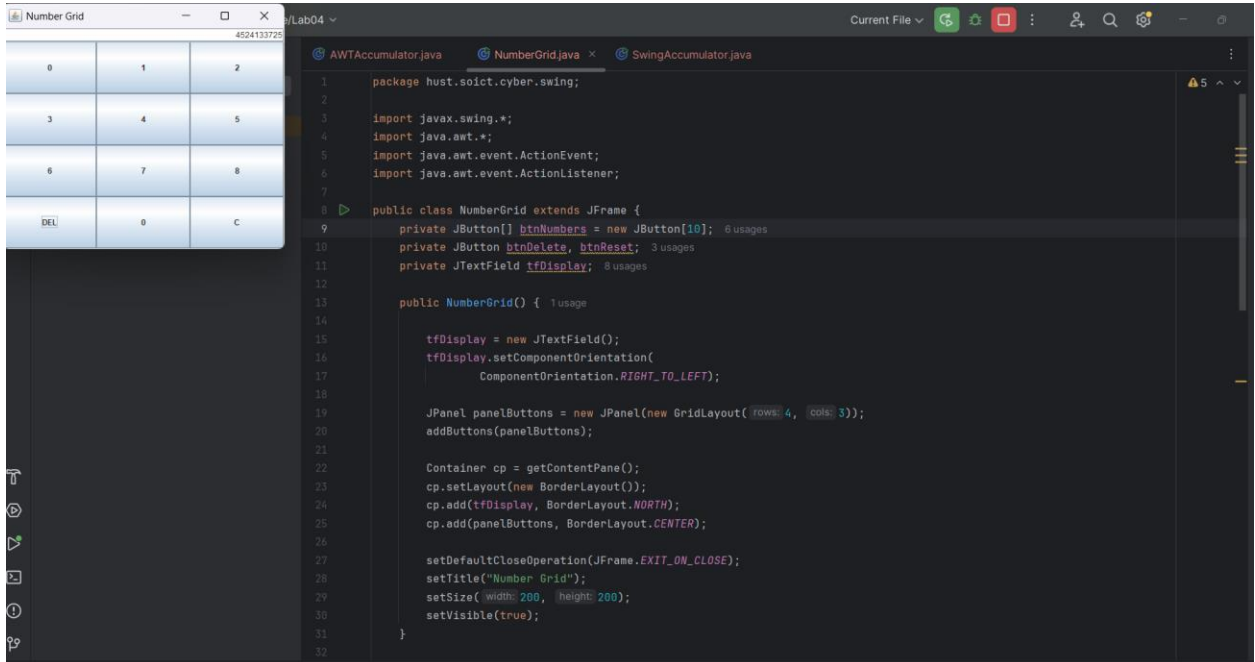
1.1 AWTAccumulator



1.2. SwingAccumulator

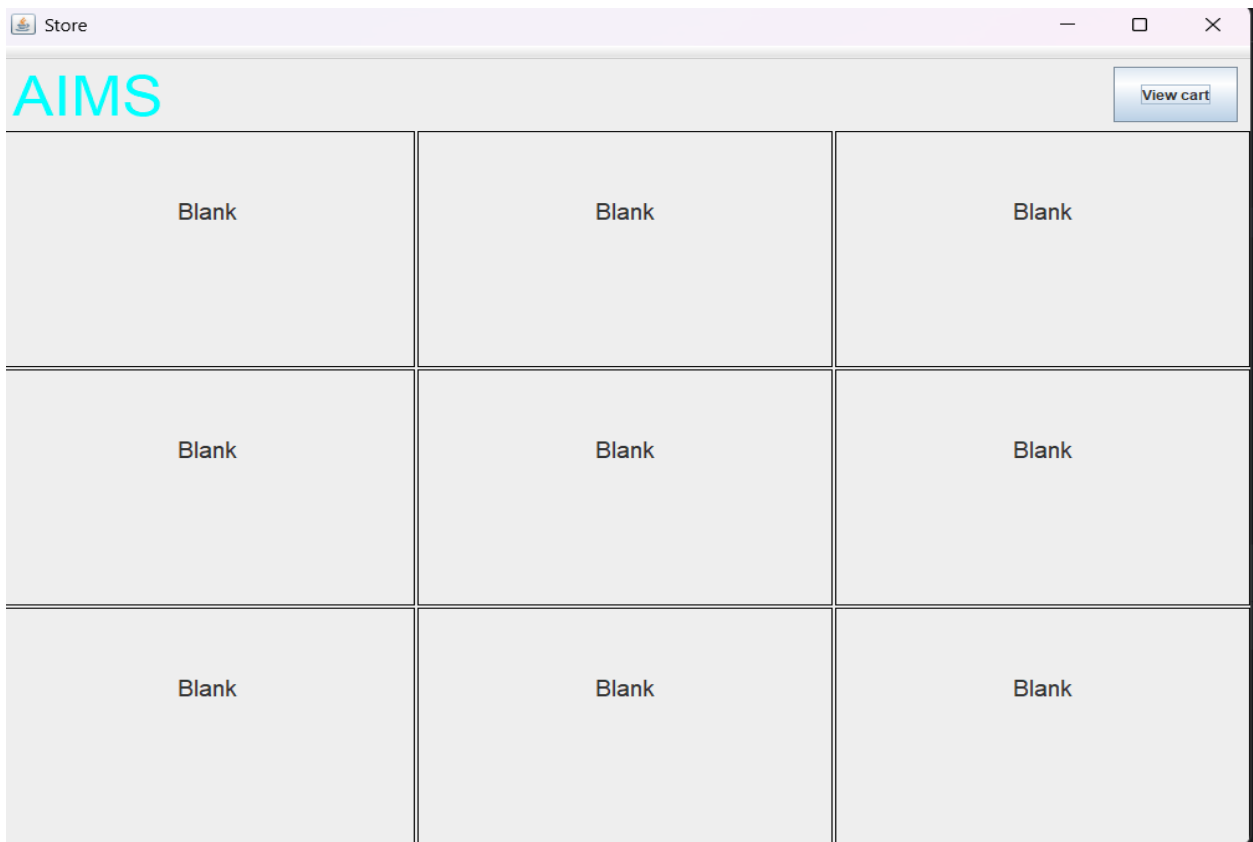


2. Organizing Swing components with Layout Managers



3. Create a graphical user interface for AIMS with Swing

3.1. View Store Screen



3.2 StoreScreen class

```
10
11 public class StoreScreen extends JFrame { 3 usages
12     private Store store; 3 usages
13     private Cart cart; 3 usages
14
15     JPanel createNorth() { 1 usage
16         JPanel north = new JPanel();
17         north.setLayout(new BorderLayout(north, BorderLayout.Y_AXIS));
18         north.add(createMenuBar());
19         north.add(createHeader());
20         return north;
21     }
22
23     JMenuBar createMenuBar() { 1 usage
24         JMenu menu = new JMenu("Options");
25
26         JMenu smUpdateStore = new JMenu("Update Store");
27         smUpdateStore.add(new JMenuItem("Add Book"));
28         smUpdateStore.add(new JMenuItem("Add CD"));
29         smUpdateStore.add(new JMenuItem("Add DVD"));
30
31         menu.add(smUpdateStore);
32         menu.add(new JMenuItem("View store"));
33         menu.add(new JMenuItem("View cart"));
34
35         JMenuBar menuBar = new JMenuBar();
36         menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
37
38         return menuBar;
39     }
40 }
```

```

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;
}

```

```

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

```
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());
        }
    }
}
```

```
        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

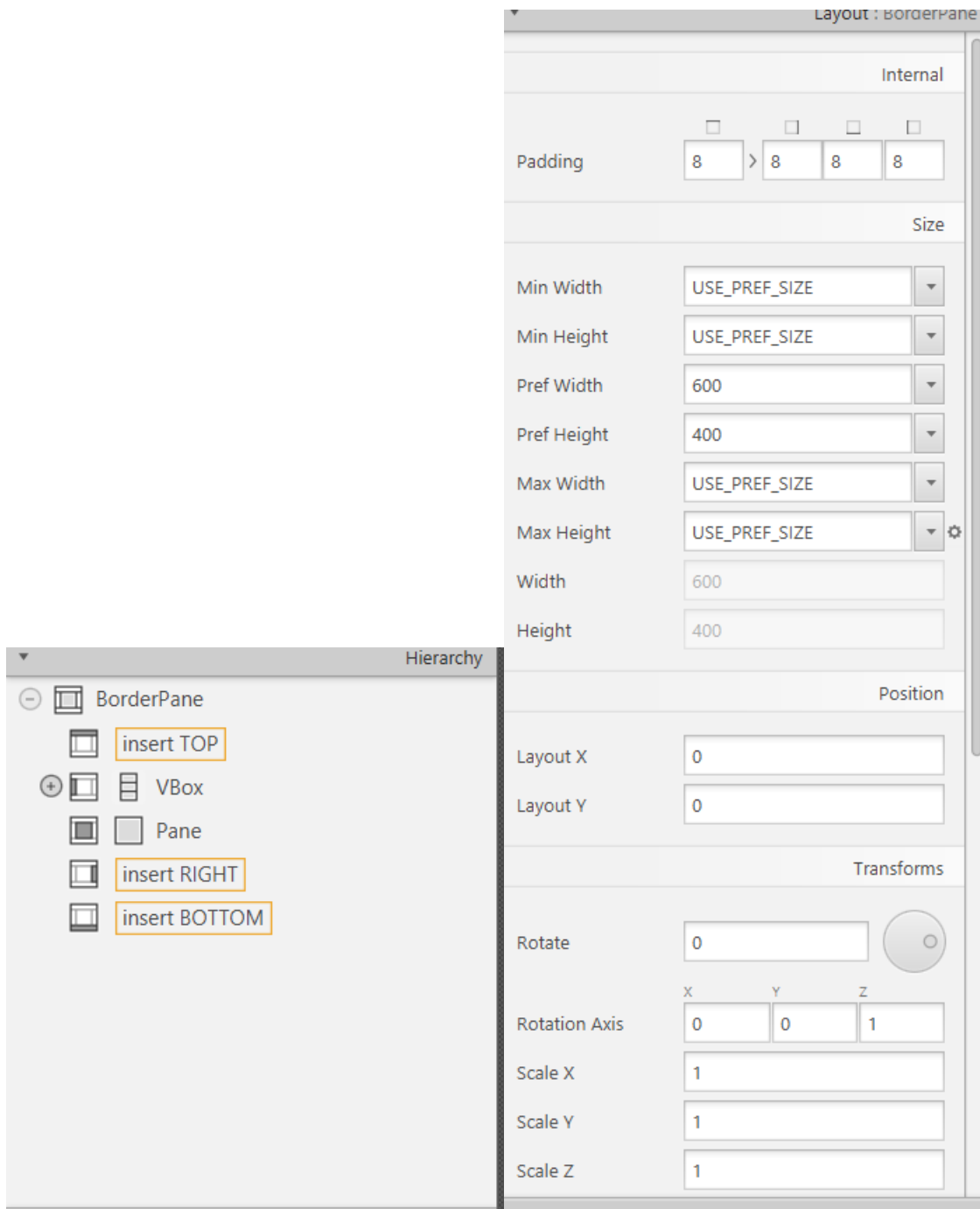
```
private class PlayButtonListener implements MouseListener{ 1 usage
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}

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

```
// 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



Code : Pane

Identity

fx:iddrawingAreaPane

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

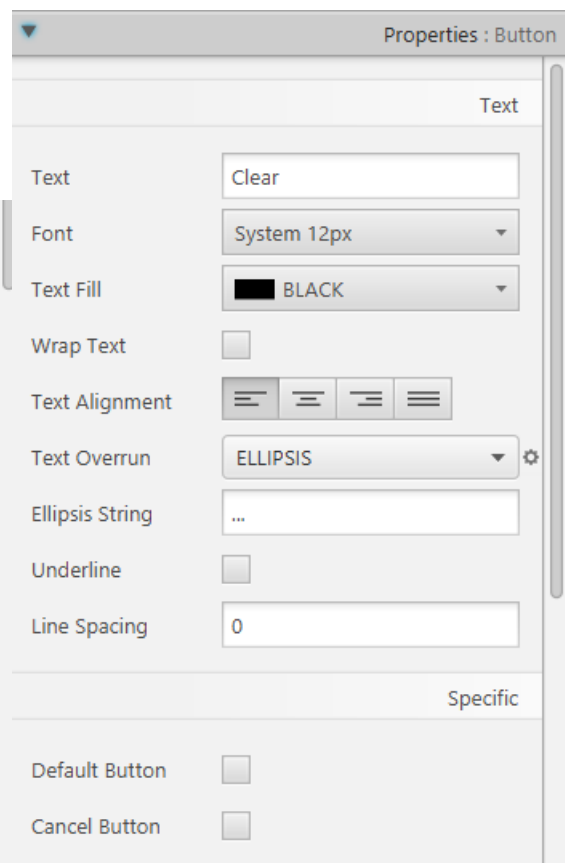
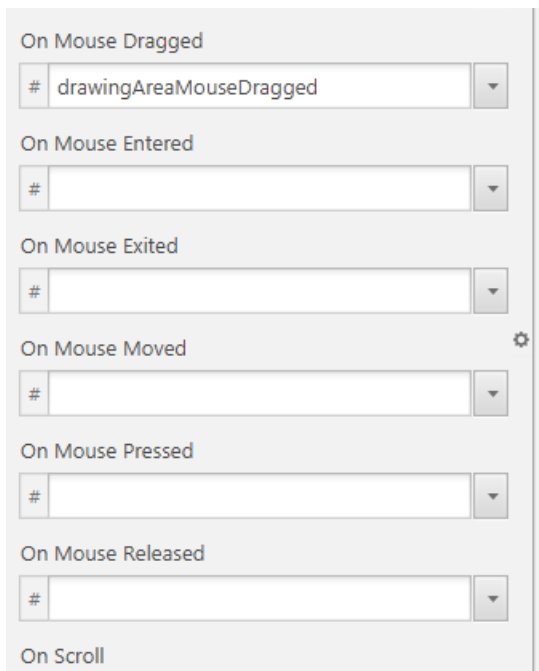
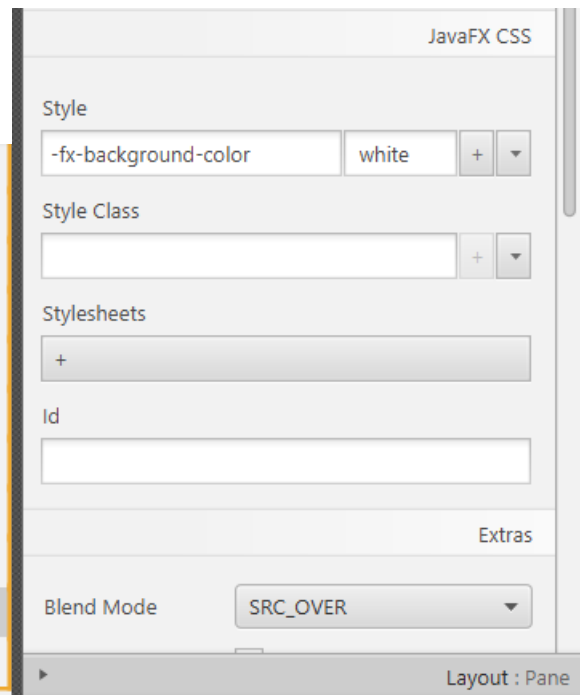
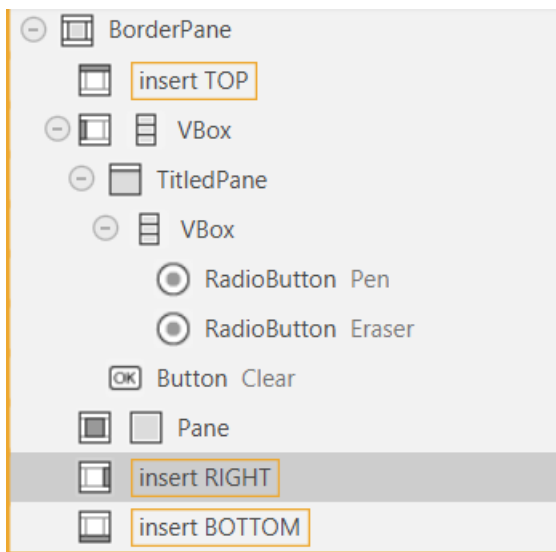
Layout X

0

Layout Y

0

Code : VBox



Object Properties

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 : Button

Identity

fx:id

Main

On Action

clearButtonPressed

DragDrop

On Drag Detected

#

On Drag Done

#

On Drag Dropped

#

On Drag Entered

#

On Drag Exited

#

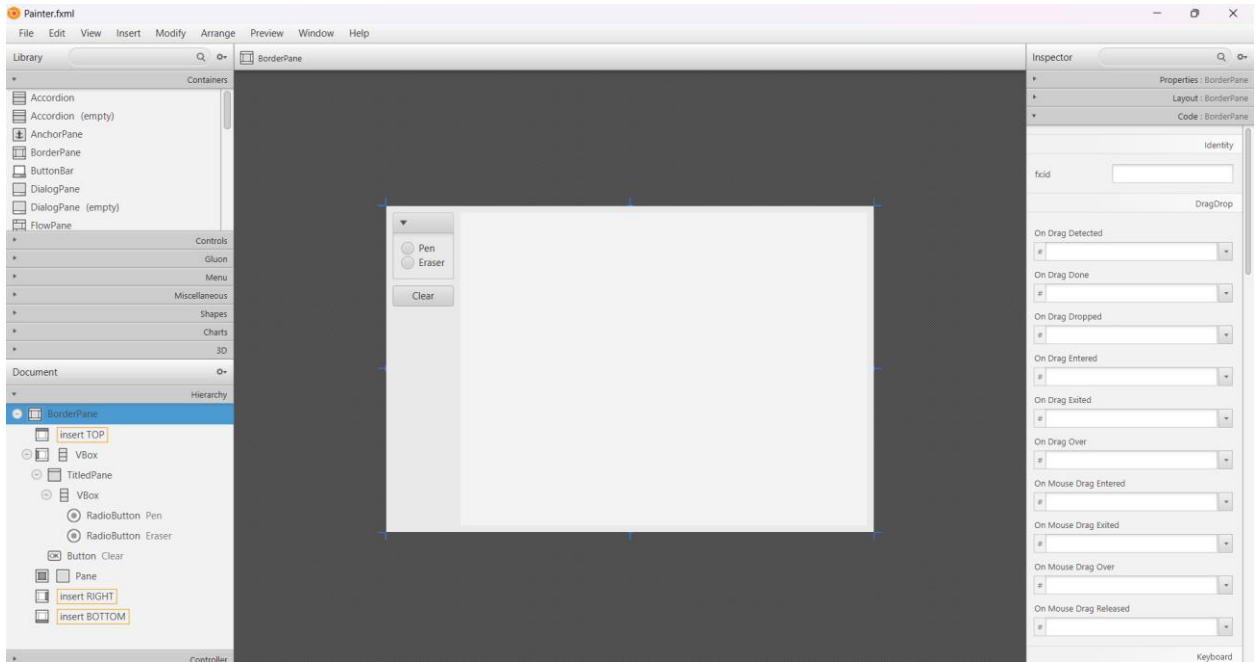
On Drag Over

#

On Mouse Drag Entered

#

On Mouse Drag Exited



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.geometry.Insets?>
4 <?import javafx.scene.control.Button?>
5 <?import javafx.scene.control.RadioButton?>
6 <?import javafx.scene.control.ToggleGroup?>
7 <?import javafx.scene.layout.BorderPane?>
8 <?import javafx.scene.layout.Pane?>
9 <?import javafx.scene.layout.VBox?>
10
11
12 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/23.0">
13   <Left>
14     <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
15       <BorderPane.margin>
16         <Insets right="8.0" />
17       </BorderPane.margin>
18       <children>
19         <TitledPane>
20           <content>
21             <VBox>
22               <children>
23                 <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Pen">
24                   <toggleGroup>
25                     <ToggleGroup fx:id="identity" />
26                   </toggleGroup>
27                 </RadioButton>
28                 <RadioButton mnemonicParsing="false" onAction="#chooseOption" text="Eraser" toggleGroup="$identity" />
29               </children>
30             </VBox>
31           </content>
32         </TitledPane>
33       </children>
34     </VBox>
35   </Left>
36   <Center>
37     <Pane />
38   </Center>
39 </BorderPane>
```

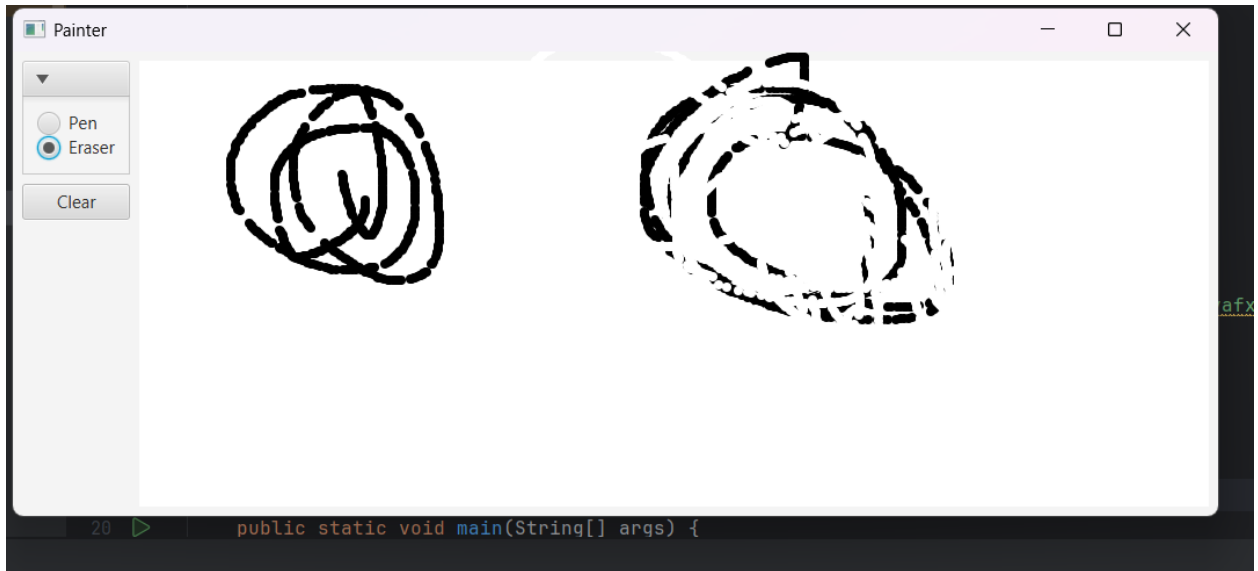
```
1 package hust.soict.cyber.javafx;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Painter extends Application {
10
11     @Override
12     public void start(Stage stage) throws Exception {
13         Parent root = FXMLLoader.load(getClass().getResource("hust/soict/cyber/javafx/Painter.fxml"));
14
15         Scene scene = new Scene(root);
16         stage.setTitle("Painter");
17         stage.setScene(scene);
18         stage.show();
19     }
20
21     public static void main(String[] args) {
22         launch(args);
23     }
24 }
```

```
13 public class PainterController { 1 usage
14     private int color; 4 usages
15
16     @FXML
17     private Pane drawingAreaPane;
18
19     @FXML
20     private ToggleGroup identity;
21
22     @FXML
23     void chooseOption(ActionEvent event) {
24         String button = ((RadioButton)event.getSource()).getText();
25         if (button.equals("Pen")) {
26             System.out.println("1");
27             color = 1;
28         } else {
29             System.out.println("0");
30             color = 0;
31         }
32     }
33
34     @FXML
35     void clearButtonPressed(ActionEvent event) {
36         drawingAreaPane.getChildren().clear();
37     }
38
39     @FXML
40     void drawingAreaMouseDragged(MouseEvent event) {
41         if (color == 1 && event.getX() >= 0) {
42             Circle newCircle = new Circle(event.getX(), event.getY(), radius: 4.0, Color.BLACK);
43             drawingAreaPane.getChildren().add(newCircle);
44         } else if (color == 0 && event.getX() >= 0) {
```

```

@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

abc

Label CART

insert LEFT

VBox

HBox

abc

Label Filter

TextField

RadioButton By ID

RadioButton By Title

TableView

ButtonBar

VBox

HBox

OK

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

>

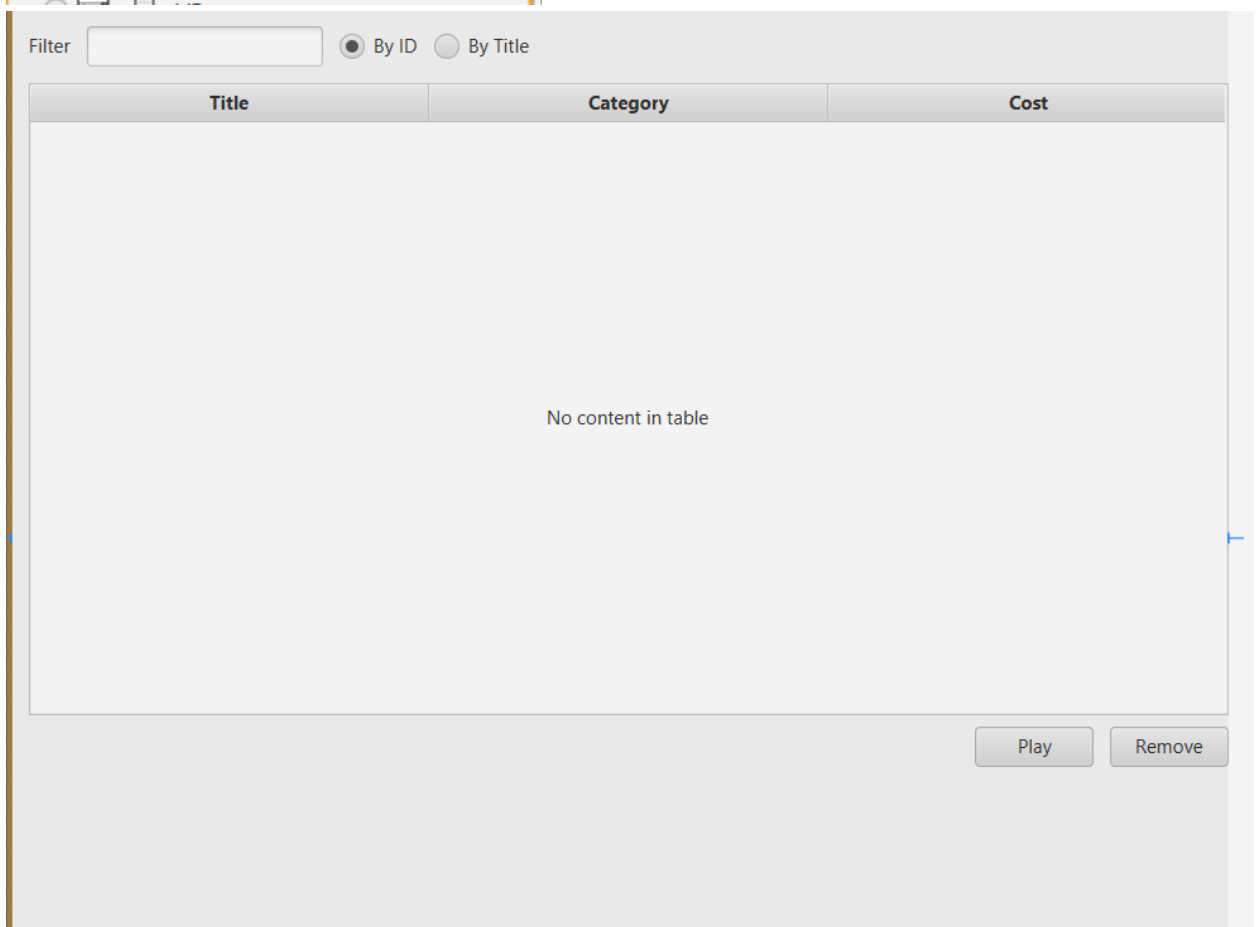
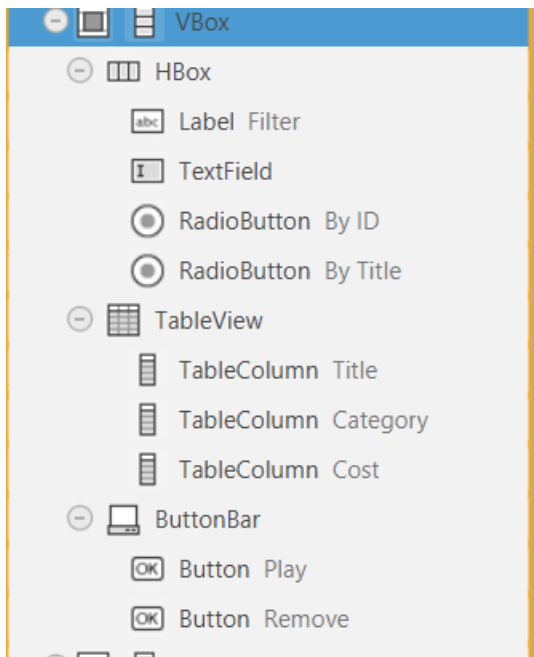
0

0

10

Size





Total:

Place Order

[-] [icon] [icon] VBox

[-] [icon] HBox

[abc] Label Total:

[I] TextField

[OK] Button Place Order

[icon] insert BOTTOM

6. Integrating JavaFX into Swing application – The JFXPanel class

```
13
14 public class CartScreen extends JFrame { 1 usage
15     private Cart cart; 1 usage
16     private Store store; no usages
17
18
19     public CartScreen(Cart cart, Store store) { 1 usage
20         super();
21         this.cart = cart;
22
23         JFXPanel fxPanel = new JFXPanel();
24         this.add(fxPanel);
25
26         this.setTitle("Cart");
27         this.setVisible(true);
28         Platform.runLater(new Runnable() {
29             @Override
30             public void run() {
31                 try {
32                     FXMLLoader loader = new FXMLLoader(getClass().getResource(
33                         name = "/hust/spict/cybersec/aims/screen/cart.fxml"));
34                     CartScreenController controller = new CartScreenController(cart, store);
35                     loader.setController(controller);
36                     Parent root = (Parent) loader.load();
37                     fxPanel.setScene(new Scene(root));
38                 } catch (IOException e) {
39                     e.printStackTrace();
40                 }
41             }
42         });
43         this.setSize( width: 1024, height: 768);
44     }
```

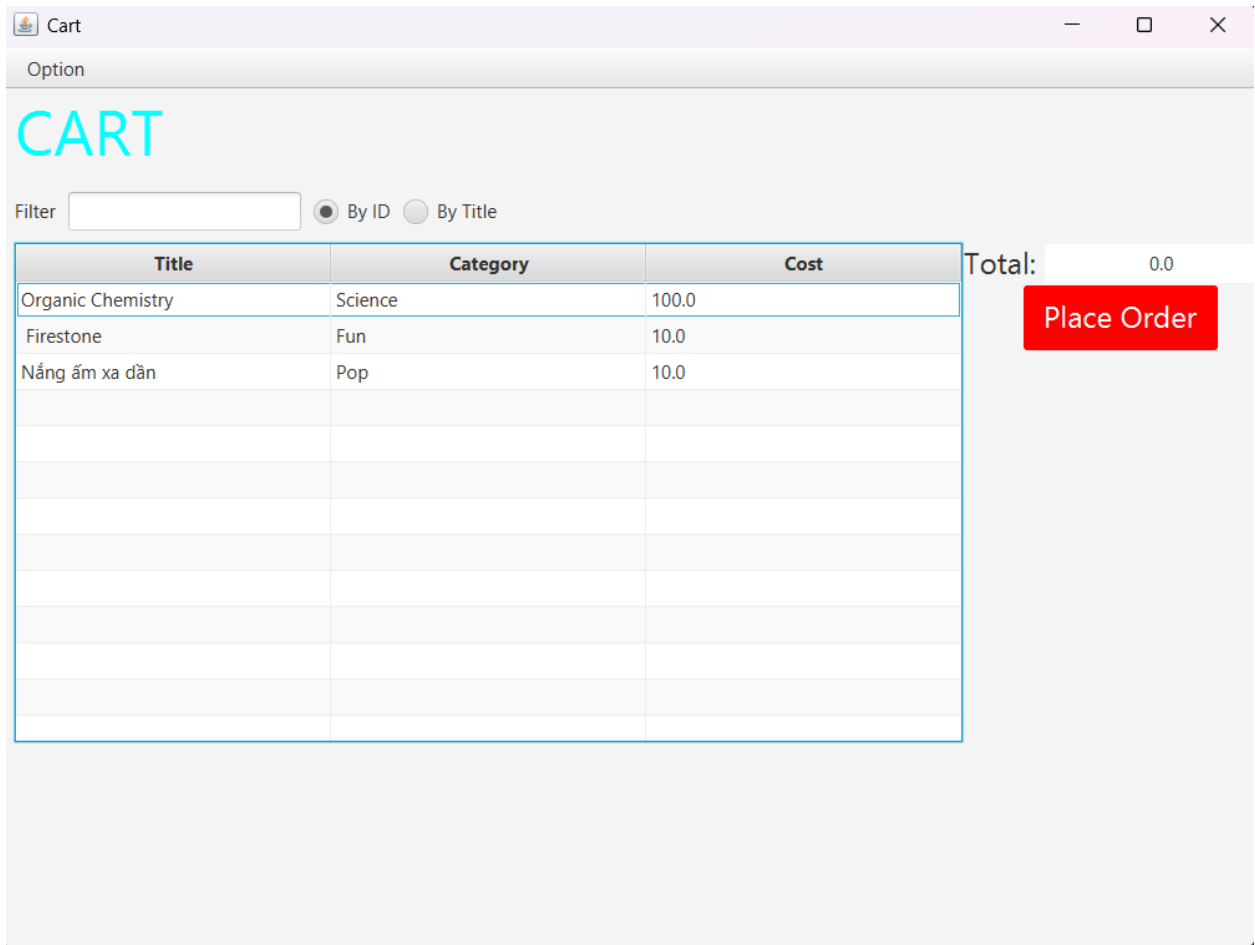
7. View the items in cart – JavaFX's data-driven UI

| | | | |
|----------------|--|---------------|--|
| | | Identity | |
| fx:id | | colMediaTitle | |
| | | Edit | |
| On Edit Start | | # | |
| On Edit Commit | | # | |
| On Edit Cancel | | # | |

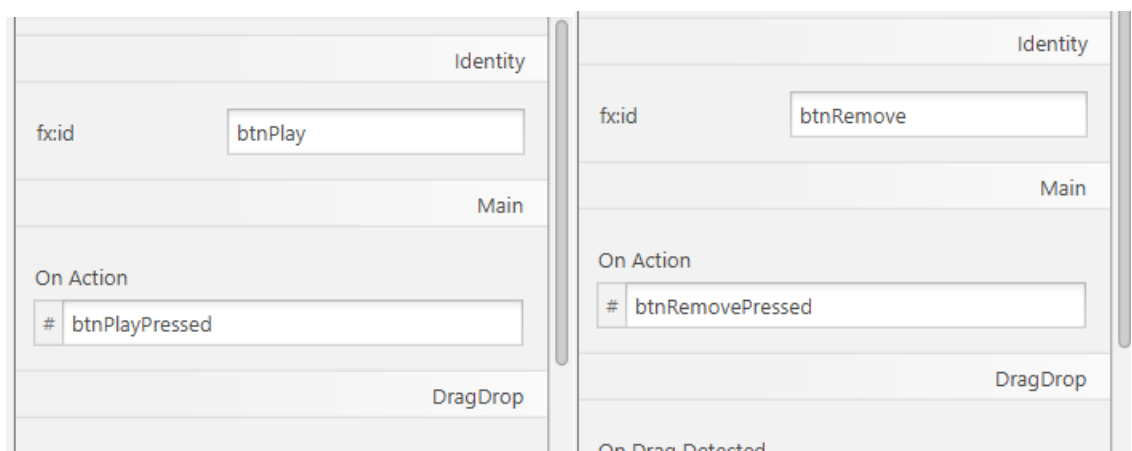
| | | | |
|----------|----------|----------|--|
| Identity | | | |
| fx:id | tblMedia | | |
| | | Main | |
| On Sort | | # | |
| | | DragDrop | |

| | | | |
|----------------|------------------|----------------|--------------|
| Identity | | Identity | |
| fx:id | colMediaCategory | fx:id | colMediaCost |
| Edit | | Edit | |
| On Edit Start | | On Edit Start | |
| # | | # | |
| On Edit Commit | | On Edit Commit | |
| # | | # | |
| On Edit Cancel | | On Edit Cancel | |
| # | | # | |

- Update itemsOrdered to ObservableList



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



Cart

Option

CART

Filter

☒ By ID

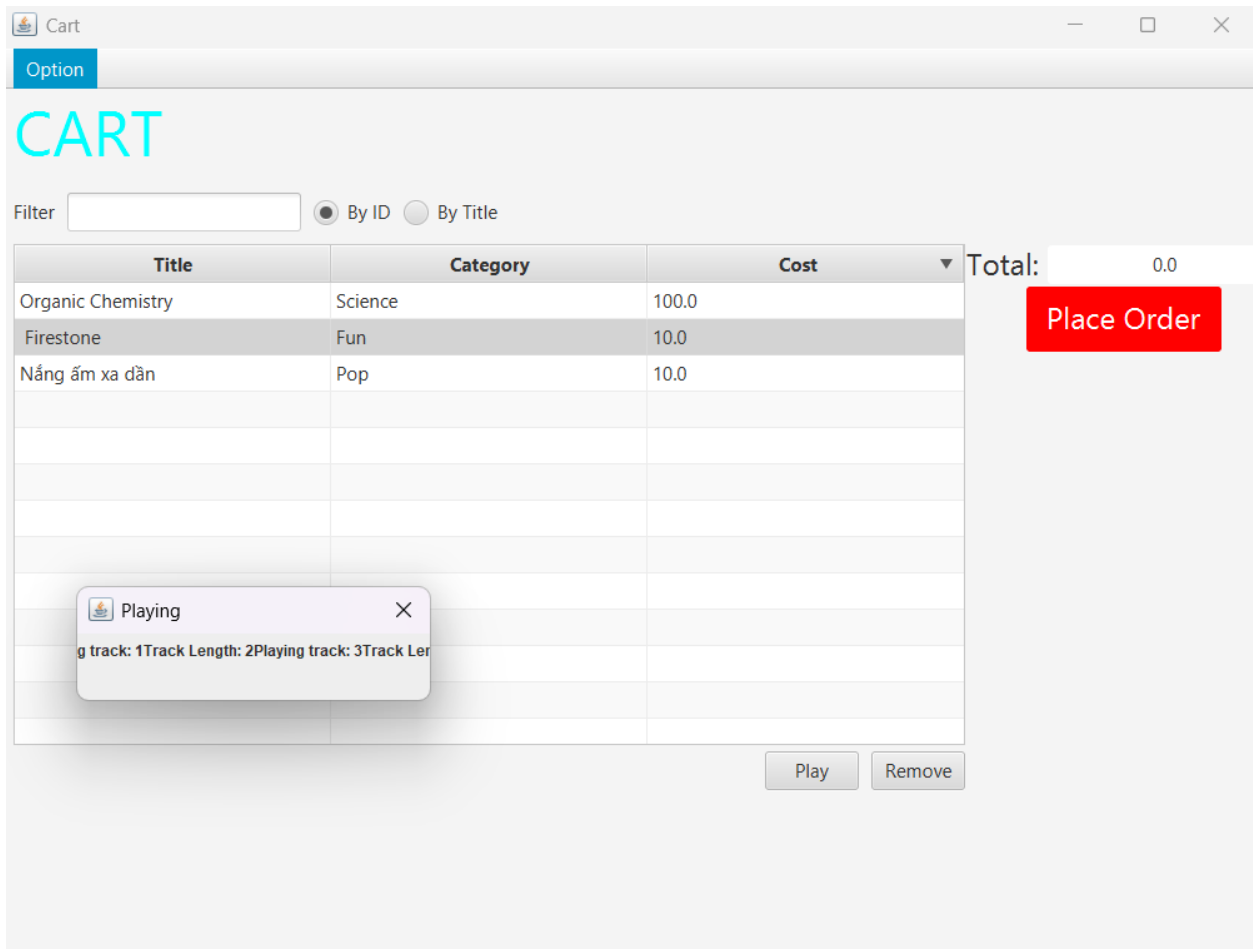
☐ By Title

| Title | Category | Cost |
|-------------------|----------|-------|
| Organic Chemistry | Science | 100.0 |
| Firestone | Fun | 10.0 |
| Nắng ấm xa dần | Pop | 10.0 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Remove

Total: 0.0

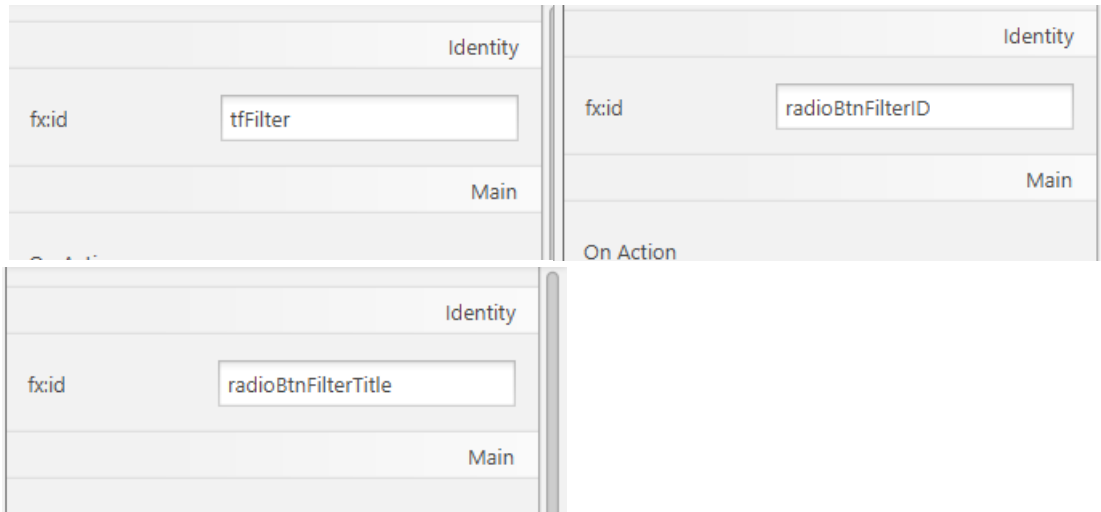
Place Order



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



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

```
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:

Category:

Authors:

Cost:

Add

 Add CD—□×

Option

Add CD

CD's Title

Category:

Tracks [title1, length1, title2, track2,...]

Author:

Cost:

Add



Add Book



Option

Add DVD

Title:

Category:

Length:

Author:

Cost:

Add

| | | |
|--|---|--|
| <div>Store</div> <div>AIMS</div> <div>View cart</div> | | |
| <div>Organic Chemistry</div> <div>100.0\$</div> <div>Add to cart</div> | <div>Firestone</div> <div>10.0\$</div> <div>Add to cart</div> <div>Play</div> | <div>Năng ẩm xa dần</div> <div>10.0\$</div> <div>Add to cart</div> <div>Play</div> |
| 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

☒ By ID

☐ By Title

| Title | Category | Cost |
|-------------------|----------|-------|
| Organic Chemistry | Science | 100.0 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Remove

Total: 120.0

Place Order

Cart

Option

CART

Filter ☐ By ID ☒ By Title

| Title | Category | Cost |
|-------------------|----------|-------|
| Organic Chemistry | Science | 100.0 |
| Firestone | Fun | 10.0 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Total: 120.0

Place Order

Remove

AddItemToStoreScreen code:

```

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:

```

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;
    }
}

```

```
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 initialize() {
        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:

```
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
```

```

public AddCompactDiscScreenController(Store _store) {
    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];

```

```

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:


```

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("/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

```

```

        @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();
                }
            }
        }
    }
}

```

```

@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, "Error: Wrong format of input");
    }
}
}

```

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

```

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

```

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

```

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

```

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

```

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

```

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.");
        }
}
}


```

```

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

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

17. Update Aims class diagram

