

Báo cáo OOP – LAB 05

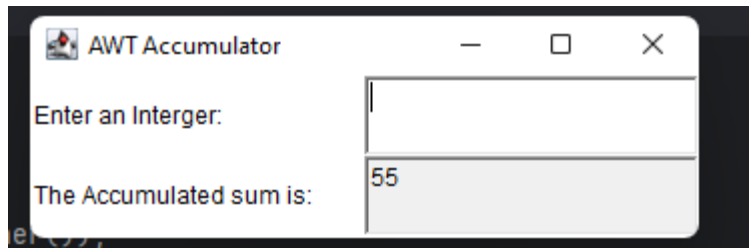
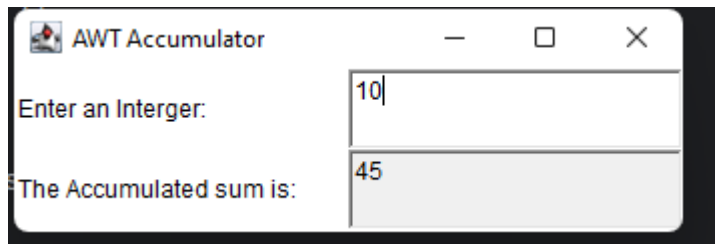
Họ và Tên: Trần Khánh Quỳnh – 20225762

1. Swing Component

1.1 AWTAccumulator

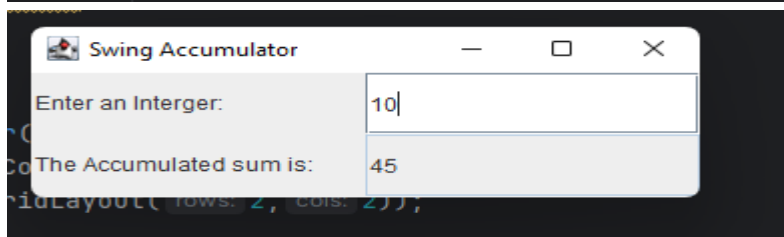
```
AWTAccumulator.java x
1 package swing;
2 > import ...
8 public class AWTAccumulator extends Frame{
    5 usages
9     private TextField tfInput;
    4 usages
10    private TextField tfOutput;
    2 usages
11    private int sum = 0;
12
    1 usage
13    public AWTAccumulator(){
14        setLayout(new GridLayout( rows: 2, cols: 2));
15        add(new Label( text: "Enter an Integer: "));
16
17        tfInput = new TextField( columns: 10);
18        add(tfInput);
19        tfInput.addActionListener(new tfInputListener());
20
21        add(new Label( text: "The Accumulated sum is: "));
22
23        tfOutput = new TextField( columns: 10);
24        tfOutput.setEditable(false);
25        add(tfOutput);
26
27        setTitle("AWT Accumulator");
28        setSize( width: 350, height: 120);
29        setVisible(true);
30    }
31
32 > public static void main(String[] args){ new AWTAccumulator(); }
    1 usage
35    private class tfInputListener implements ActionListener {
36        @Override
37        public void actionPerformed(ActionEvent evt){
38            int numberIn = Integer.parseInt(tfInput.getText());
39            sum += numberIn;
40            tfInput.setText("");
41            tfOutput.setText(sum + " ");

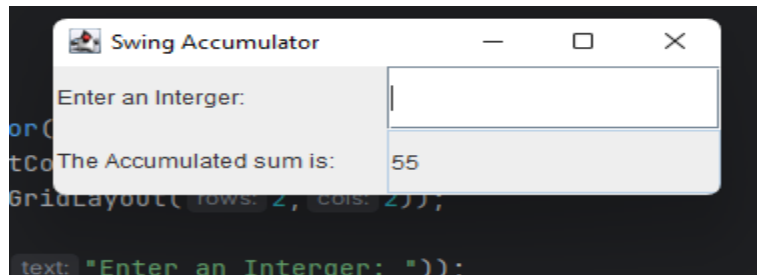
```



1.2 SwingAccumulator

```
© AWTAccumulator.java    © SwingAccumulator.java ×
1 package swing;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 new *
7 public class SwingAccumulator extends JFrame {
8     5 usages
9     private JTextField tfInput;
10    4 usages
11    private JTextField tfOutput;
12    2 usages
13    private int sum = 0;
14    1 usage new *
15    public SwingAccumulator() {
16        Container cp = getContentPane();
17        cp.setLayout(new GridLayout( rows: 2, cols: 2));
18
19        cp.add(new JLabel( text: "Enter an Interger: "));
20        tfInput = new JTextField( columns: 10);
21        cp.add(tfInput);
22        tfInput.addActionListener(new tfInputListener());
23
24        cp.add(new JLabel( text: "The Accumulated sum is: "));
25        tfOutput = new JTextField( columns: 10);
26        tfOutput.setEditable(false);
27        cp.add(tfOutput);
28
29        setTitle("Swing Accumulator");
30        setSize( width: 350, height: 120);
31        setVisible(true);
32    }
33
34    new *
35    public static void main(String[] args){
36        new SwingAccumulator();
37    }
38 }
```





2. Organizing Swing components without Layout Managers

```

© AWTAccumulator.java  © SwingAccumulator.java  © NumberGrid.java x
1  package swing;
2
3  import java.awt.*;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.ActionListener;
6  import javax.swing.*;
7
8  new *
9  public class NumberGrid extends JFrame {
10     6 usages
11     private JButton[] btnNumbers = new JButton[10];
12     3 usages
13     private JButton btnDelete, btnReset;
14     9 usages
15     private JTextField tfDisplay;
16
17     1 usage new *
18     public NumberGrid() {
19         tfDisplay = new JTextField();
20         tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
21         JPanel panelButtons = new JPanel(new GridLayout( rows: 4, cols: 3));
22         addButtons(panelButtons);
23
24         Container cp = getContentPane();
25         cp.setLayout((new BorderLayout()));
26         cp.add(tfDisplay, BorderLayout.NORTH);
27         cp.add(panelButtons, BorderLayout.CENTER);
28         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29         setTitle("Number Grid");
30         setSize( width: 200, height: 200);
31         setVisible(true);
32     }
33
34     1 usage new *
35     void addButtons(JPanel panelButtons) {
36         ButtonListener btnListener = new ButtonListener();
37         for (int i=1; i<=9; i++) {
38             btnNumbers[i] = new JButton( text: "" + i);

```



3. Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen

```
StoreScreen.java x
1  package aims.screen;
2
3  import javax.swing.*;
4  import aims.cart.Cart;
5  import aims.media.Media;
6  import aims.store.Store;
7  import aims.screen.MediaStore;
8  import aims.media.*;
9  import java.awt.*;
10 import java.awt.event.ActionEvent;
11 import java.awt.event.ActionListener;
12 import java.util.ArrayList;
13 import java.util.List;
14
15 public class StoreScreen extends JFrame {
16
17     5 usages
18     private Store store;
19     10 usages
20     private Container cp;
21     3 usages
22     private Cart cart;
23
24     1 usage
25     JPanel createNorth() {
26         JPanel north = new JPanel();
27         north.setLayout(new BorderLayout(north, BorderLayout.Y_AXIS));
28         north.add(createMenuBar());
29         north.add(createHeader());
30         return north;
31     }
32
33     1 usage
34     JMenuBar createMenuBar() {
```

```

JMenuItem addCD = new JMenuItem(text: "Add CD");
smUpdateStore.add(addCD);
addCD.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { AddCDStoreScreen popUp = new AddCDStoreScreen(store); }
});

JMenuItem addDVD = new JMenuItem(text: "Add DVD");
smUpdateStore.add(addDVD);
addDVD.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { AddDVDStoreScreen popUp = new AddDVDStoreScreen(store); }
});
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));
menuBar.add(menu);
return menuBar;
}

```

1 usage

```

JPanel createHeader() {
    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);
}

```

```

private class AddBookStoreScreen extends JFrame {
    1 usage
    public AddBookStoreScreen(Store store) {
        this.setLayout(new GridLayout( rows: 4, cols: 2, hgap: 5, vgap: 5));
        this.add(new JLabel( text: "Enter title: "));
        TextField title = new TextField( columns: 10);
        this.add(title);
        this.add(new JLabel( text: "Enter category: "));
        TextField category = new TextField( columns: 10);
        this.add(category);
        this.add(new JLabel( text: "Enter cost: "));
        TextField cost = new TextField( columns: 10);
        this.add(cost);
        this.setTitle("Add Book");
        this.setSize( width: 300, height: 150);
        JButton addButton = new JButton( text: "Add");
        this.add(addButton);
        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                Book book = new Book(2, title.getText(), category.getText(), Float.parseFloat(cost.getText()));
                store.addMedia(book);
                cp.add(createCenter(), BorderLayout.CENTER);
                cp.revalidate();
                title.setText("");
                category.setText("");
                cost.setText("");
            }
        });

        this.setVisible(true);
    }
}

```



```

private class AddCDStoreScreen extends JFrame {
    1 usage
    public AddCDStoreScreen(Store store) {
        this.setLayout(new GridLayout( rows: 7, cols: 2, hgap: 5, vgap: 5));
        this.add(new JLabel( text: "Enter title: "));
        TextField title = new TextField( columns: 10);
        this.add(title);
        this.add(new JLabel( text: "Enter category: "));
        TextField category = new TextField( columns: 10);
        this.add(category);
        this.add(new JLabel( text: "Enter cost: "));
        TextField cost = new TextField( columns: 10);
        this.add(cost);
        this.add(new JLabel( text: "Enter artist: "));
        TextField artist = new TextField( columns: 10);
        this.add(artist);
        this.setTitle("Add CD");
        this.add(new JLabel( text: "Number of tracks: "));
        TextField numberOfTracks = new TextField( columns: 10);
        this.add(numberOfTracks);
        this.pack();
        JButton turnInBtn = new JButton( text: "Add");
        this.add(turnInBtn);
        turnInBtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                CompactDisc dvd = new CompactDisc(2, title.getText(), category.getText(),
                    Float.parseFloat(cost.getText()), artist.getText(), new ArrayList<Track>());
                store.addMedia(dvd);
                cp.add(createCenter(), BorderLayout.CENTER);
                cp.revalidate();
                title.setText("");
                category.setText("");
            }
        });
    }
}

```

```

private class AddDVDStoreScreen extends JFrame {
    1 usage
    public AddDVDStoreScreen(Store store) {
        this.setLayout(new GridLayout( rows: 4, cols: 2, hgap: 5, vgap: 5));
        this.add(new JLabel( text: "Enter title: "));
        TextField title = new TextField( columns: 10);
        this.add(title);
        this.add(new JLabel( text: "Enter category: "));
        TextField category = new TextField( columns: 10);
        this.add(category);
        this.add(new JLabel( text: "Enter cost: "));
        TextField cost = new TextField( columns: 10);
        this.add(cost);

        this.setTitle("Add DVD");
        this.setSize( width: 300, height: 100);
        JButton turnInBtn = new JButton( text: "Add");
        this.add(turnInBtn);
        turnInBtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                DigitalVideoDisc dvd = new DigitalVideoDisc(2, title.getText(), category.getText(),
                    Float.parseFloat(cost.getText()));
                store.addMedia(dvd);
                cp.add(createCenter(), BorderLayout.CENTER);
                cp.revalidate();
                title.setText("");
                category.setText("");
                cost.setText("");
            }
        });
        this.setVisible(true);
    }
}

```

```
1 usage
JPanel createHeader() {
    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 cart1 = new JButton(text: "View cart");
    cart1.setPreferredSize(new Dimension(width: 100, height: 50));
    cart1.setMaximumSize(new Dimension(width: 100, height: 50));
    cart1.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) { new CartScreen(cart).setVisible(true); }
    });
    header.add(Box.createRigidArea(new Dimension(width: 10, height: 10)));
    header.add(title);
    header.add(Box.createHorizontalGlue());
    header.add(cart1);
    header.add(Box.createRigidArea(new Dimension(width: 10, height: 10)));
    return header;
}

4 usages
JPanel createCenter() {
    JPanel center = new JPanel();
    center.setLayout(new GridLayout(rows: 3, cols: 3, hgap: 2, vgap: 2));
    List<Media> mediaStore = store.getItemsInStore();
    for (Media media : mediaStore) {
        MediaStore cell = new MediaStore(media, cart);
        center.add(cell);
    }
}
```

3.2 Create class MediaStore

```

1 package aims.screen;
2 > import ...
8
3 messages 2 related problems
9 public class MediaStore extends JPanel{
    1 usage
10     private Media media;
    1 usage 2 related problems
11 @
12     public MediaStore(Media media, Cart myCart) {
13         this.media = media;
14         this.setLayout(new BoxLayout(target: this, BoxLayout.Y_AXIS));
15
16         JLabel title = new JLabel(media.getTitle());
17         title.setFont(new Font(title.getFont().getName(),Font.PLAIN, size: 20));
18         title.setAlignmentX(CENTER_ALIGNMENT);
19
20         JLabel cost = new JLabel("" + media.getCost() + "$");
21         cost.setAlignmentX(CENTER_ALIGNMENT);
22
23         JPanel container = new JPanel();
24         container.setLayout(new FlowLayout(FlowLayout.CENTER));
25         JButton btnAdd = new JButton(text: "Add to cart");
26         container.add(btnAdd);
27         btnAdd.addActionListener(new ActionListener() {
28             @Override
29             public void actionPerformed(ActionEvent e) {
30                 myCart.addMedia(media);
31                 JOptionPane.showMessageDialog(
32                     parentComponent: null,
33                     message: "Media \" + media.getTitle() + "\" has been added to the cart successfully!",
34                     title: "Add to Cart",
35                     JOptionPane.INFORMATION_MESSAGE //
36             );
37         });
38     }
39 }

```

```

39         if(media instanceof Playable) {
40             JButton playBtn= new JButton( text: "Play");
41             playBtn.addActionListener(new ActionListener() {
42                 @Override
43                 public void actionPerformed(ActionEvent e) {
44                     JDialog playDialog = createPlayDialog(media);
45                     playDialog.setVisible(true);
46                     playDialog.setSize( width: 300, height: 200);
47                     playDialog.pack();
48                 }
49             });
50             container.add(playBtn);
51         }
52         this.add(Box.createVerticalGlue());
53         this.add(title);
54         this.add(cost);
55         this.add(Box.createVerticalGlue());
56         this.add(container);
57         this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
58     }
59
60     1 usage
61     @ static JDialog createPlayDialog(Media media) {
62         JDialog playDialog = new JDialog();
63         Container container = playDialog.getContentPane();
64         playDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
65         container.setLayout(new BoxLayout(container,BoxLayout.Y_AXIS));
66         container.add(Box.createRigidArea(new Dimension( width: 10, height: 10)));
67         if(media instanceof DigitalVideoDisc dvd) {
68             container.add(new JLabel("Playing DVD:" + dvd.getTitle()));
69             container.add(new JLabel("DVD length:" + dvd.getLength() + " min"));
70         } else if (media instanceof CompactDisc cd) {
71             container.add(new JLabel("Title: " + cd.getTitle()));
72             container.add(new JLabel("Artist: " + cd.getArtist()));

```

3.3 Result

AIMS

View cart

J97
26.5\$

Add to cart

Play

EDM
32.25\$

Add to cart

Play

OOP
20.5\$

Add to cart

Play

Tran Khanh Quynh
25.5\$

Add to cart

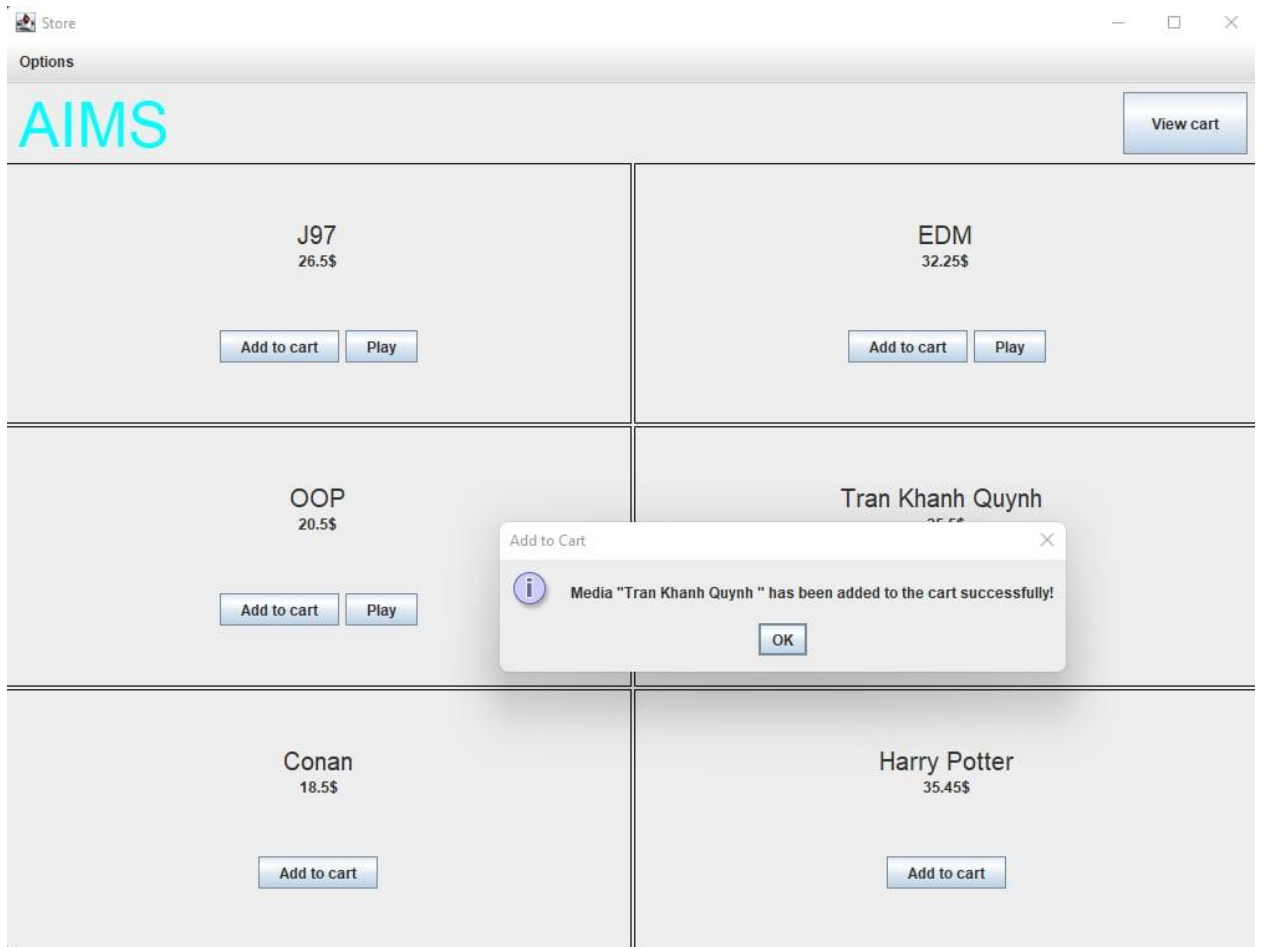
Play

Conan
18.5\$

Add to cart

Harry Potter
35.45\$

Add to cart



=

AIMS

[View cart](#)

J97
26.5\$

[Add to cart](#)[Play](#)

EDM
32.25\$

[Add to cart](#)[Play](#)

OOP
20.5\$

[Add to cart](#)[Play](#)

Tran Khanh Quynh
25.5\$

[Add to cart](#)[Play](#)

Conan
18.5\$

[Add to cart](#)

Harry Potter
35.45\$

[Add to cart](#)

Play J97

Title: J97
Artist: Various artist
Play: Thien Ly oi. Length: 7 min
Play: Thien An. Length: 4 min
Play: 5Milion. Length: 5 min

CART

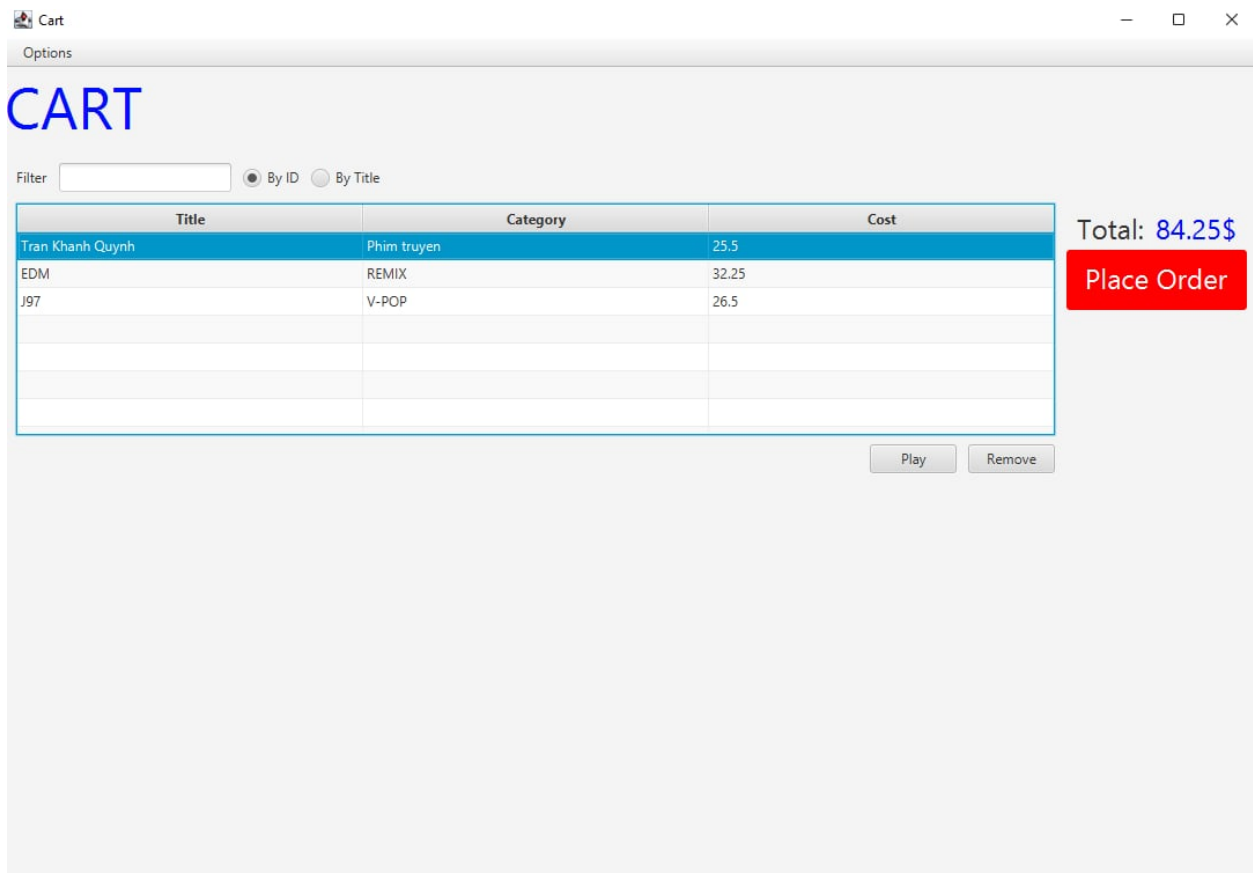
Filter

☒ By ID
 ☐ By Title

Title	Category	Cost
OOP	Curious	20.5
EDM	REMIX	32.25
Tran Khanh Quynh	Phim truyen	25.5

Total: 78.25\$

Place Order



4. JavaFX API

4.1. Create class Painter

```
Painter.java x
Runnable class
1  import java.util.Objects;
2
3  import java.util.Objects;
4
5  import javafx.application.Application;
6  import javafx.fxml.FXMLLoader;
7  import javafx.scene.Parent;
8  import javafx.scene.Scene;
9  import javafx.stage.Stage;
10
11  public class Painter extends Application {
12      @Override
13      public void start(Stage stage) throws Exception {
14          try {
15              Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("Painter.fxml")));
16              Scene scene = new Scene(root);
17              stage.setTitle("Painter");
18              stage.setScene(scene);
19              stage.show();
20          } catch (Exception e) {
21              System.out.println(e.toString());
22          }
23      }
24
25      public static void main(String[] args) { launch(args); }
26  }
```

4.2 Create Painter.fxml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.*?>
4  <?import javafx.scene.control.*?>
5  <?import javafx.scene.layout.*?>
6
7  <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" >
8      <top>
9          <MenuBar BorderPane.alignment="CENTER">
10              <menus>
11                  <Menu mnemonicParsing="false" text="File">
12                      <items>
13                          <MenuItem mnemonicParsing="false" text="Close" />
14                      </items>
15                  </Menu>
16                  <Menu mnemonicParsing="false" text="Edit">
17                      <items>
18                          <MenuItem mnemonicParsing="false" text="Delete" />
19                      </items>
20                  </Menu>
21                  <Menu mnemonicParsing="false" text="Help">
22                      <items>
23                          <MenuItem mnemonicParsing="false" text="About" />
24                      </items>
25                  </Menu>
26              </menus>
27          </MenuBar>
28      </top>
29  </BorderPane>
```

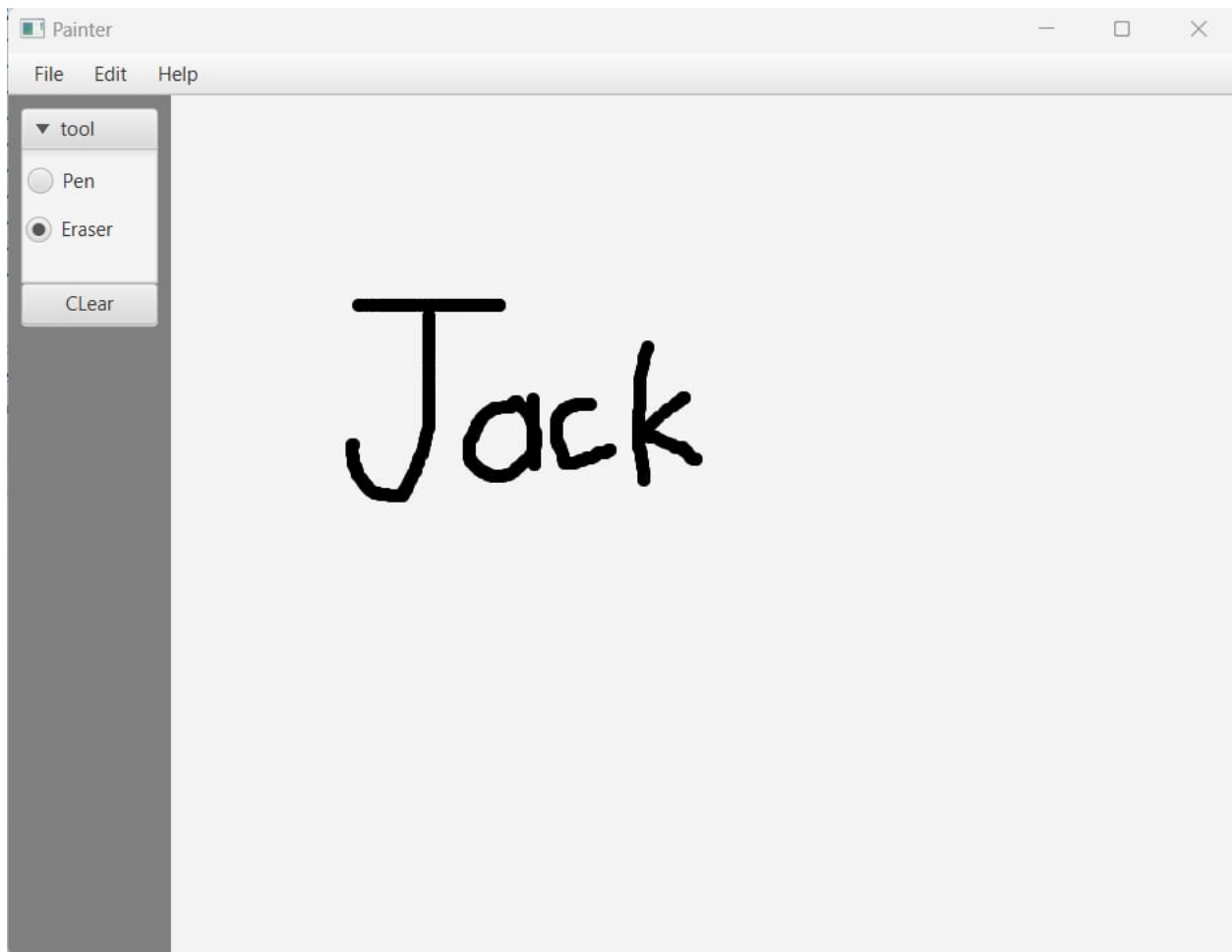
4.3 Create PainterController

```

Painter.java PainterController.java x painter.fxml
17
18     1 usage
19     @FXML
20     private RadioButton eraserButton;
21
22     3 usages
23     @FXML
24     private AnchorPane drawingAreaPane;
25
26     no usages
27     @FXML
28     void clearButtonPressed(ActionEvent event) { drawingAreaPane.getChildren().clear(); }
29
30     no usages
31     @FXML
32     void drawAreaMouseDragged(MouseEvent event) {
33         if (penRadioButton.isSelected()) {
34             Circle circle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
35             drawingAreaPane.getChildren().add(circle);
36         } else if (eraserButton.isSelected()) {
37             Iterator<Node> iter = drawingAreaPane.getChildren().iterator();
38             while (iter.hasNext()) {
39                 Node c = iter.next();
40                 if (c instanceof Circle circle) {
41                     if (circle.getCenterX() <= event.getX() + 4 && circle.getCenterX() >= event.getX() - 4) {
42                         if (circle.getCenterY() <= event.getY() + 4 && circle.getCenterY() >= event.getY() - 4) iter.remove();
43                     }
44                 }
45             }
46         }
47     }

```

4.4 Result



5. View Cart Screen

5.1

Create cart.fxml

```
© StoreScreen.java  </> cart.fxml  </> Painter.fxml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <?import javafx.geometry.Insets?>
3  <?import javafx.scene.control.Button?>
4  <?import javafx.scene.control.ButtonBar?>
5  <?import javafx.scene.control.Label?>
6  <?import javafx.scene.control.Menu?>
7  <?import javafx.scene.control.MenuBar?>
8  <?import javafx.scene.control.MenuItem?>
9  <?import javafx.scene.control.RadioButton?>
10 <?import javafx.scene.control.TableColumn?>
11 <?import javafx.scene.control.TableView?>
12 <?import javafx.scene.control.TextField?>
13 <?import javafx.scene.control.ToggleGroup?>
14 <?import javafx.scene.layout.BorderPane?>
15 <?import javafx.scene.layout.HBox?>
16 <?import javafx.scene.layout.VBox?>
17 <?import javafx.scene.text.Font?>
18
19 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml" >
20     <top>
21         <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
22             <children>
23                 <MenuBar>
24                     <menus>
25                         <Menu mnemonicParsing="false" text="Options">
26                             <items>
27                                 <MenuItem mnemonicParsing="false" text="Update store">
28                                     <items>
29                                         <MenuItem mnemonicParsing="false" text="Add book" />
30                                         <MenuItem mnemonicParsing="false" text="Add DVD" />
31                                         <MenuItem mnemonicParsing="false" text="Add CD" />
32                                     </items>
33                                 </MenuItem>
34                                 <Menu mnemonicParsing="false" text="View store">
35                                     <items>
36                                         <MenuItem mnemonicParsing="false" text="Action 1" />
37                                     </items>
38                                 </Menu>
39                                 <Menu mnemonicParsing="false" text="View cart">
```

5.2 Create CartScreen

```
StoreScreen.java  </> cart.fxml  CartScreen.java  </> Painter.fxml
3  > import ...
14
15  1 usage new *
public class CartScreen extends JFrame {
16
17      1 usage
private Cart cart;
18
19      1 usage new *
public CartScreen(Cart cart) {
20          super();
21          this.cart = cart;
22          JFXPanel fxPanel = new JFXPanel();
23          this.add(fxPanel);
24          this.setTitle("Cart");
25          this.setVisible(true);
26          setSize( width: 1100, height: 768);
27      new *
Platform.runLater(new Runnable() {
28          new *
@Override
29  @
public void run() {
30      try {
31          FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "cart.fxml"));
32          CartScreenController controller = new CartScreenController(cart);
33          loader.setController(controller);
34          Parent root = loader.load();
35          fxPanel.setScene(new Scene(root));
36      } catch (IOException e) {
37          e.printStackTrace();
38      }
39  }
40  });
41  }
42  }
43
44
```

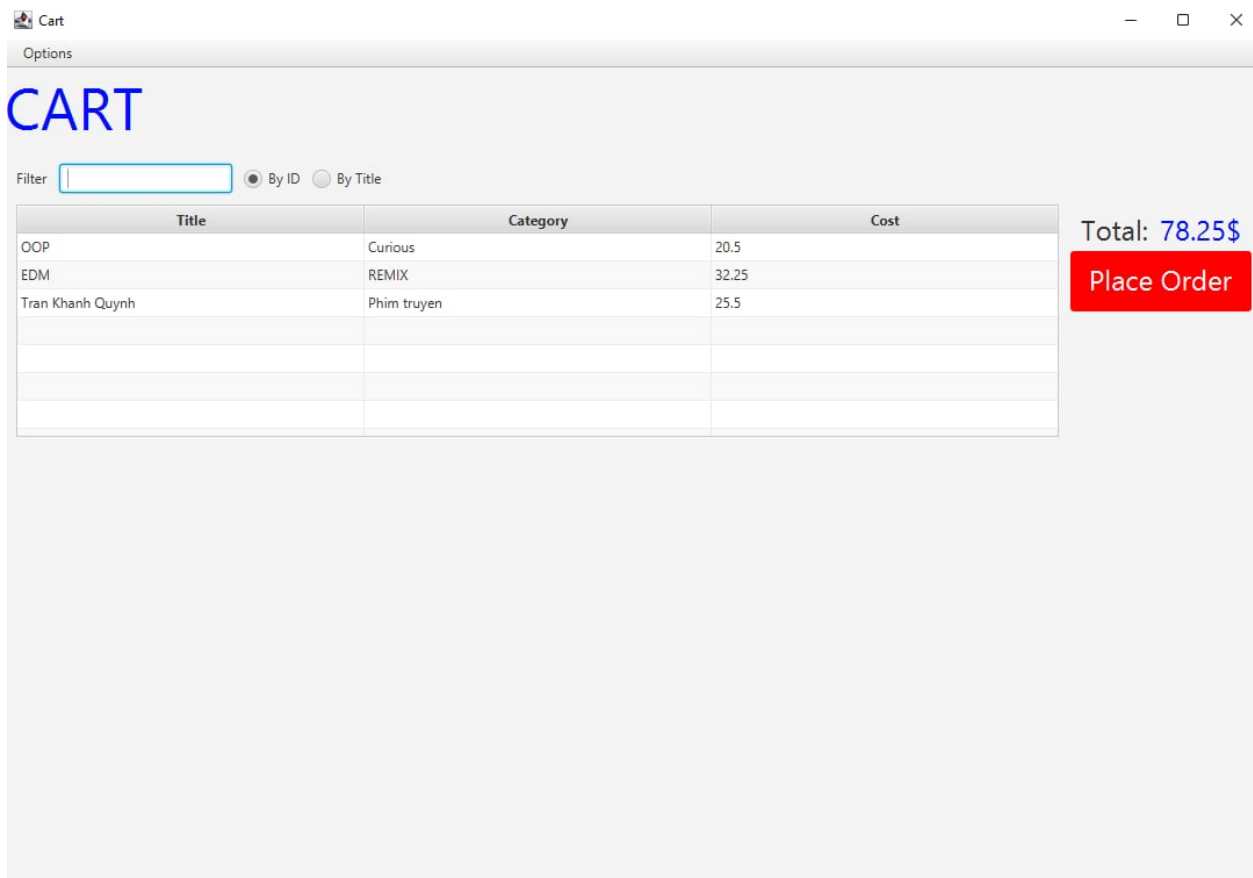
5.3 Create CartScreenController

```

59     }
60
61     no usages new *
62     @FXML
63     private void initialize() {
64         colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media,String>(s: "title"));
65         colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media,String>(s: "category"));
66         colMediaCost.setCellValueFactory(new PropertyValueFactory<Media,Float>(s: "cost"));
67         tblMedia.setItems(FXCollections.observableList(this.cart.getItemsOrdered()));
68         tblMedia.setPlaceholder(new Label(s: "No item in cart"));
69
70         btnPlay.setVisible(false);
71         btnRemove.setVisible(false);
72
73         new *
74         btnRemove.setOnAction(new EventHandler<javaafx.event.ActionEvent>() {
75             new *
76             @Override
77             public void handle(javaafx.event.ActionEvent event) {
78                 Media media = tblMedia.getSelectionModel().getSelectedItem();
79                 cart.removeMedia(media);
80                 totalPrice.setText(Float.toString((float) cart.totalCost()) + "$");
81                 tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
82             }
83         });
84         new *
85         tfFilter.textProperty().addListener(new ChangeListener<String>() {
86             new *
87             @Override
88             public void changed(ObservableValue<? extends String> observableValue, String s, String t1) {
89                 showFilterMedia(t1);
90             }
91         });
92
93         tblMedia.getSelectionModel().selectedItemProperty().addListener(
94             new *
95             new ChangeListener<Media>() {
96                 new *
97                 @Override
98                 public void changed(ObservableValue<? extends Media> observableValue, Media media, Media t1) {
99                     updateButtonBar(t1);
100                 }
101             }
102         );
103     }

```

5.4 Result



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

6.1 Edit class CartScreenController

```
btnPlay.setOnAction(new EventHandler<ActionEvent>() {
    new *
    @Override
    public void handle(ActionEvent event) {
        JDialog playDialog = MediaStore.createPlayDialog(tblMedia.getSelectionModel().getSelectedItem());
        playDialog.setVisible(true);
        playDialog.setSize( width: 300, height: 200);
        playDialog.pack();
    }
});
```

```

1 usage new *
@FXML
void updateButtonBar(Media media) {
    btnRemove.setVisible(true);
    if(media instanceof Playable) {
        btnPlay.setVisible(true);
    } else {
        btnPlay.setVisible(false);
    }
}
}

```

6.2 Result

Cart

Options

CART

Filter

☒ By ID
 ☐ By Title

Title	Category	Cost
Tran Khanh Quynh	Phim truyen	25.5
EDM	REMIX	32.25
J97	V-POP	26.5

Total: 84.25\$

Place Order

Play

Remove

7 Deleting a media

7.1 Code

```

btnRemove.setOnAction(new EventHandler<javafx.event.ActionEvent>() {
    new *
    @Override
    public void handle(javafx.event.ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        cart.removeMedia(media);
        totalPrice.setText(Float.toString((float) cart.totalCost()) + "$");
        tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
    }
});

```

7.2 Result

Cart

Options

CART

Filter

☒ By ID
 ☐ By Title

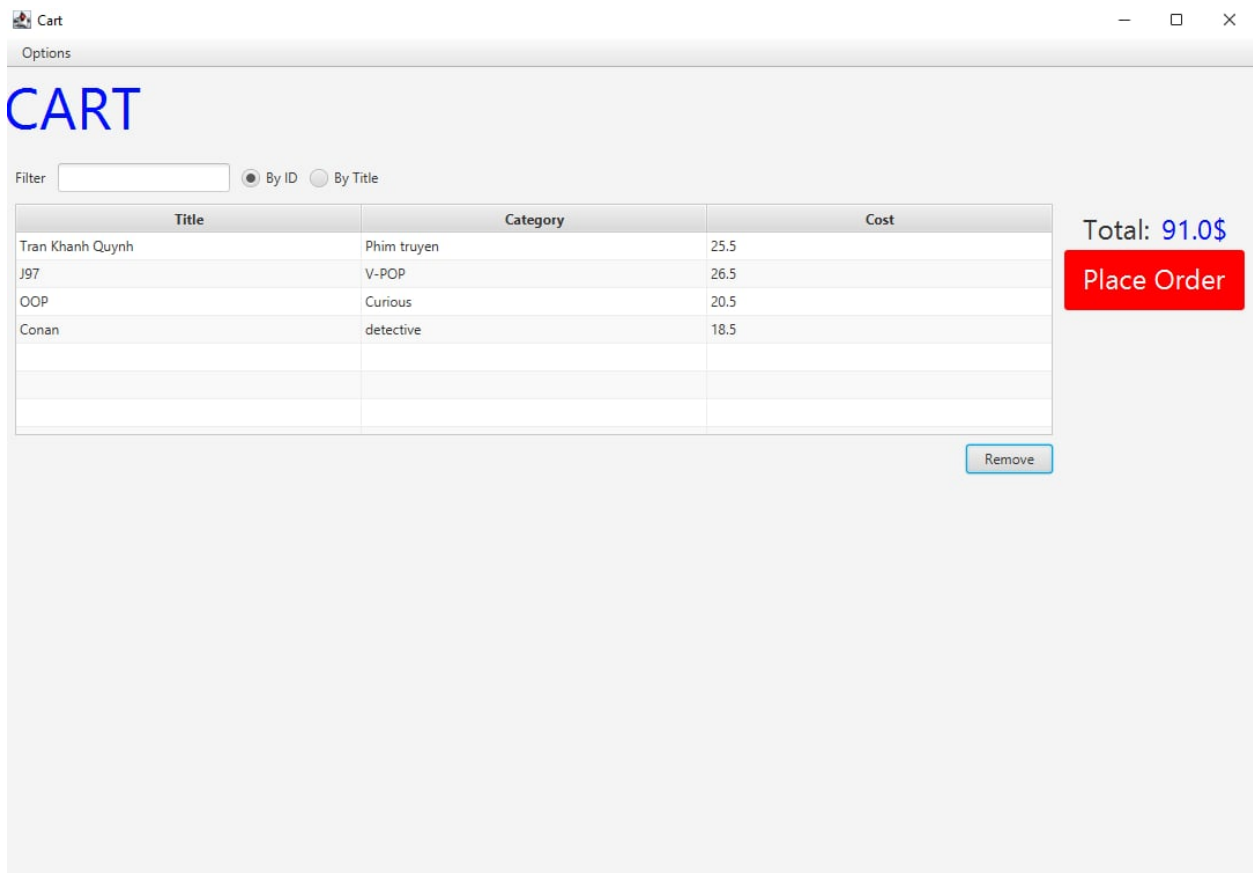
Title	Category	Cost
Tran Khanh Quynh	Phim truyen	25.5
EDM	REMIX	32.25
J97	V-POP	26.5
OOP	Curious	20.5
Conan	detective	18.5

Total: 104.75\$

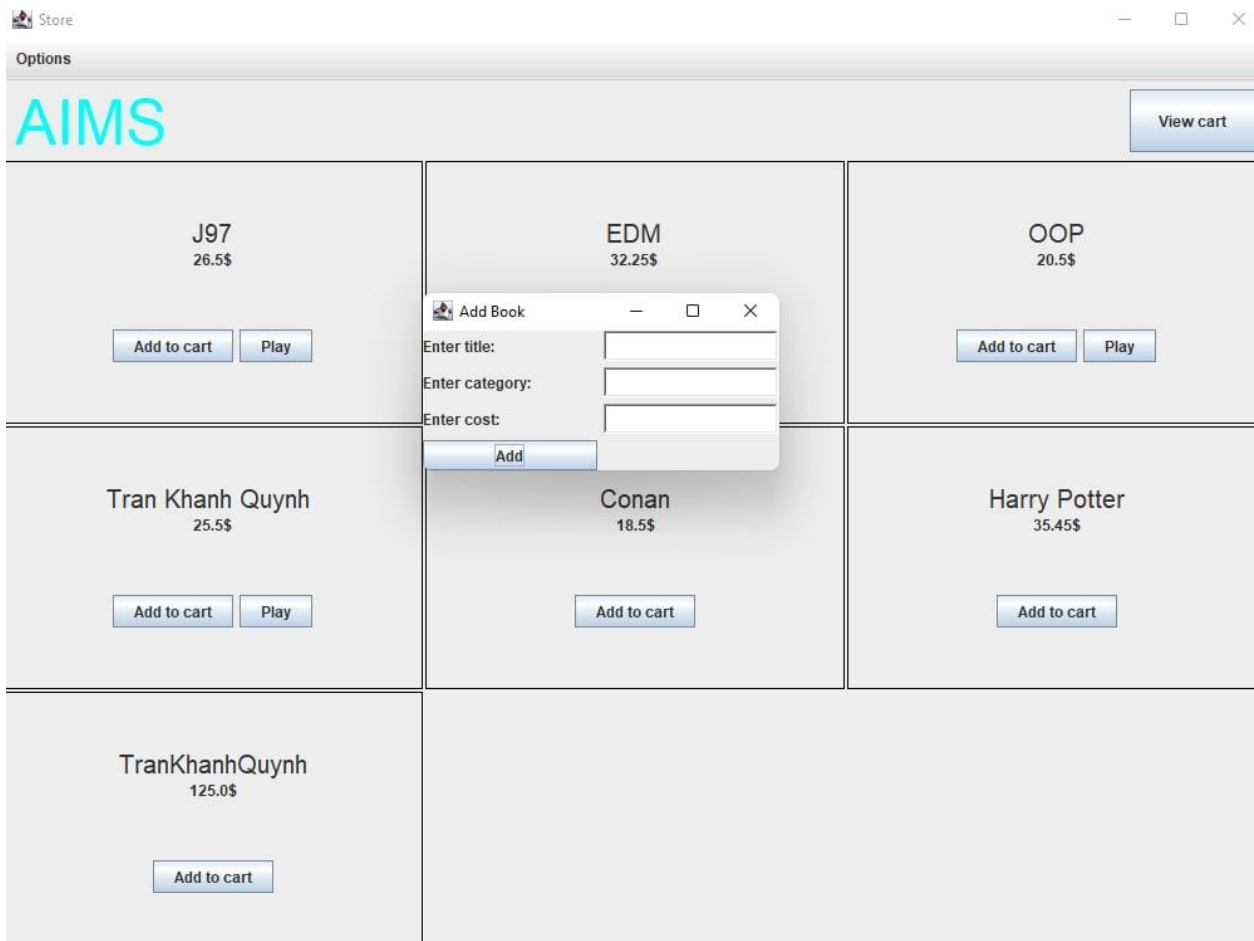
Place Order

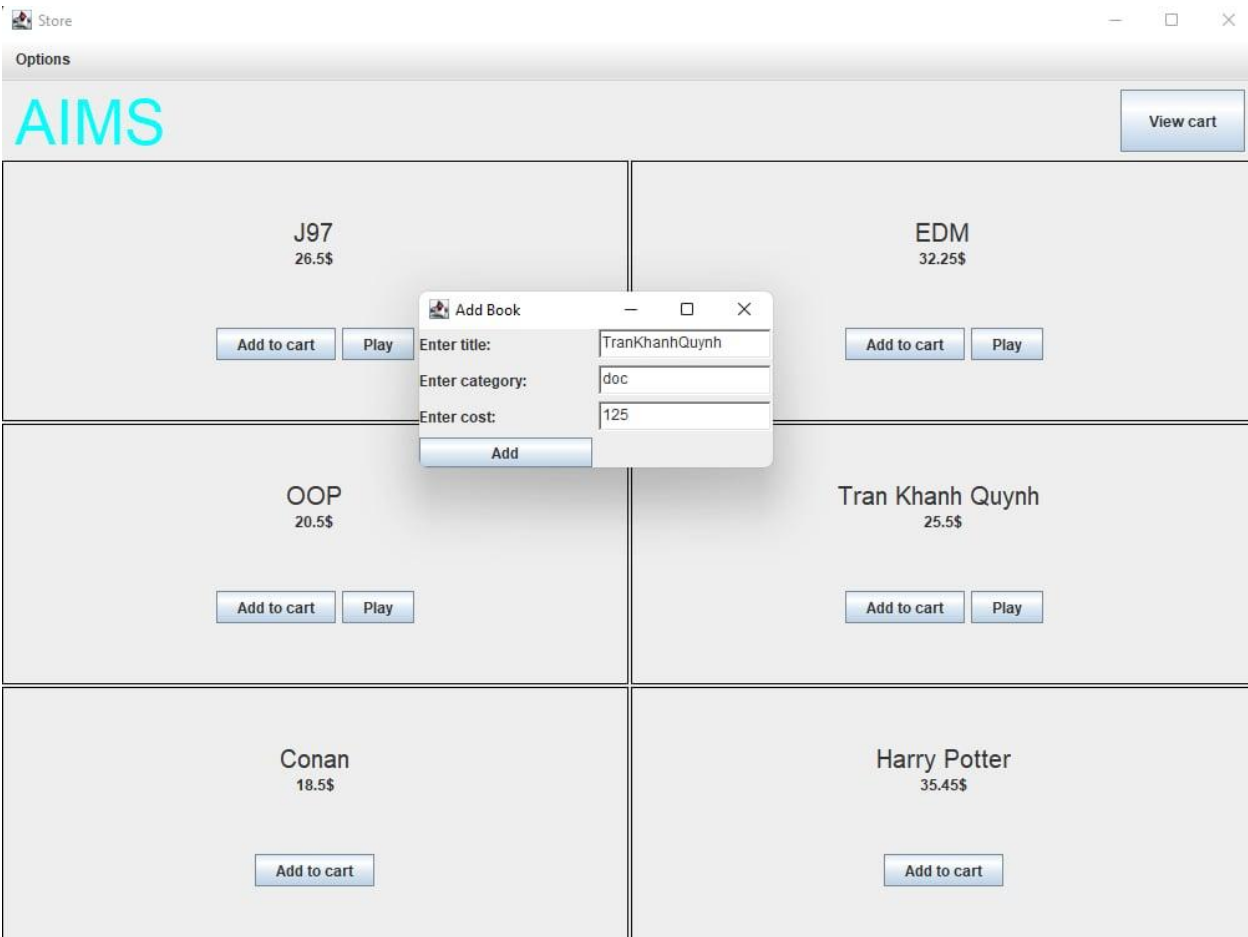
Play

Remove



8 Complete the Aims GUI application





AIMS

View cart

J97
26.5\$

Add to cart

Play

EDM
32.25\$

Add Book

Enter title:

Enter category:

Enter cost:

Add

OOP
20.5\$

Add to cart

Play

Tran Khanh Quynh
25.5\$

Add to cart

Play

Conan
18.5\$

Add to cart

Harry Potter
35.45\$

Add to cart

TranKhanhQuynh
125.0\$

Add to cart

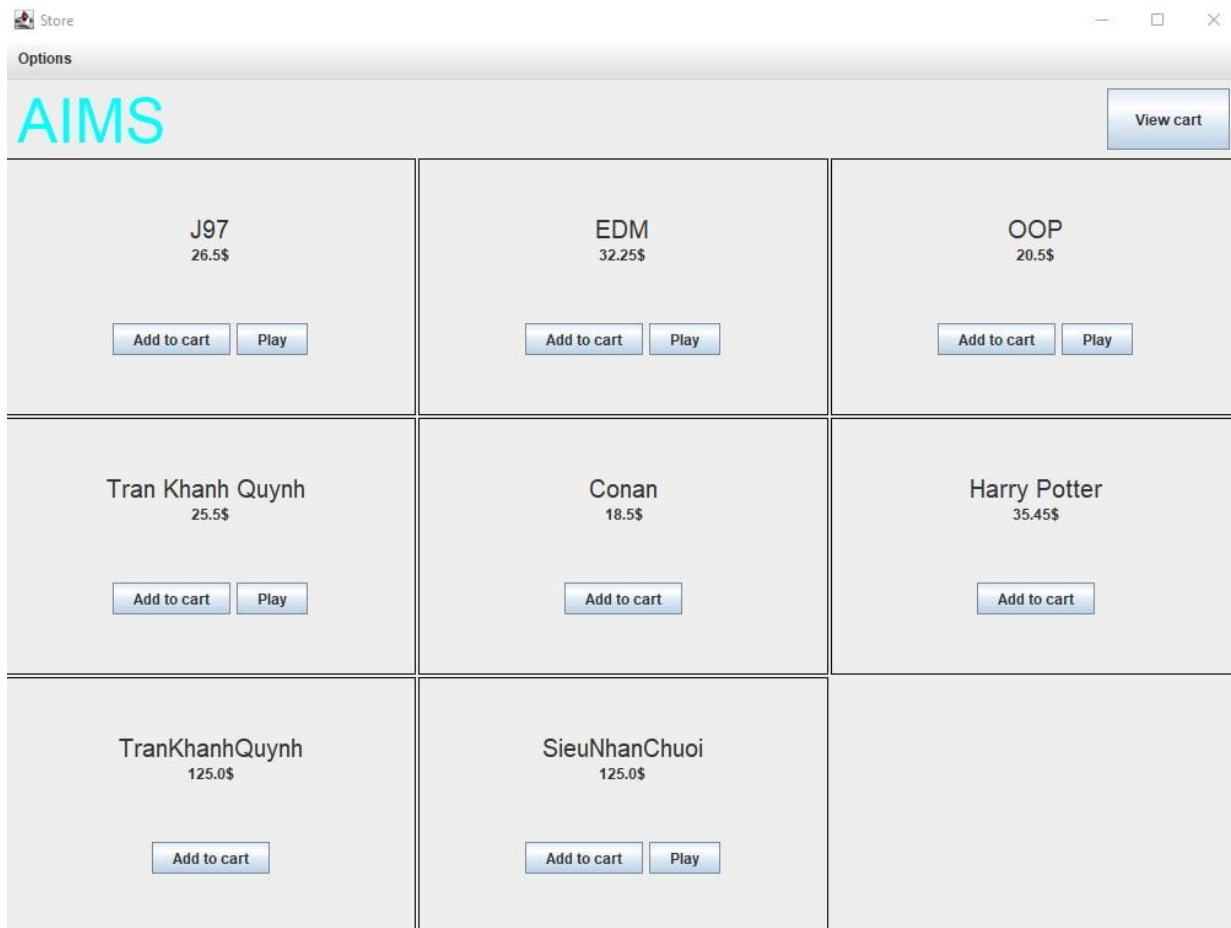
Store

Options

AIMS

View cart

<div>J97</div> <div>26.5\$</div> <div>Add to cart</div> <div>Play</div>	<div>EDM</div> <div>32.25\$</div> <div>Add to cart</div>	<div>OOP</div> <div>20.5\$</div> <div>Add to cart</div> <div>Play</div>
<div>Tran Khanh Quynh</div> <div>25.5\$</div> <div>Add to cart</div> <div>Play</div>	<div>Add CD</div> <div>Enter title: <input type="text" value="SieuNhanChuoi"/></div> <div>Enter category: <input type="text" value="film"/></div> <div>Enter cost: <input type="text" value="125"/></div> <div>Enter artist: <input type="text" value="TranKhanhQuynh"/></div> <div>Number of tracks: <input type="text" value="1"/></div> <div>Add</div> <div>Add to cart</div>	<div>Harry Potter</div> <div>35.45\$</div> <div>Add to cart</div>
<div>TranKhanhQuynh</div> <div>125.0\$</div> <div>Add to cart</div>		



```
52  public void play() throws PlayException {  
53      if(this.getLength() < 0) {  
54          throw new PlayException("ERROR: DVD-length is non-positive!");  
55      } else {  
56          System.out.println("\ntitle: " + getTitle() + '\n' + "artist: " + getArtist() + "\n\n" + "TRacks:");  
57          for (Track song : tracks) {  
58              song.play();  
59          }  
60      }  
61  }
```

9. ClassDiagram Updated

