Assignment 2

Case Studies

1. Formulating the Problem

1.1 Problem Description

Design and implement a Java program that creates a GUI that will allow a user to reserve seats at a new movie theater.

1.2 Verbalization

What is the goal?

Create a GUI that will allow a user to reserve seats

What are the givens?

Price for the ticket

Total available seats

1.3 Information Elicitation

Goal

To collect user's order.

Store an order.

Print the order for user's review.

Givens

Price for the ticket

Total available seats

Unknowns

None

Conditions

None

2. Planning the Solution

2.1 Solution Strategy

Create observable array lists which will store prices, movies, and seats. Create Order Handler class which will be implementing Action Event interface and subscribe it when the user clicks a button.

2.2 Goal Decomposition

Sub-goal 1

Get data from the user.

Sub-goal 2

Listen to the event.

Calculate the order cost

Sub-goal 3

Display data.

2.3 Resources

Relevant formulas

subtotal = (cost*topCost);

total = (subtotal* TAXRATE) + subtotal;

2.4 Data Organization and Description

Input (givens):

Name	Description	Origin	Used in Sub-goal #
Full Name	User's full name	User	1
Price	Price to the ticket	User	1
Movie type	Collection of movies available	User	1
Seats	Number of seats available	User	1

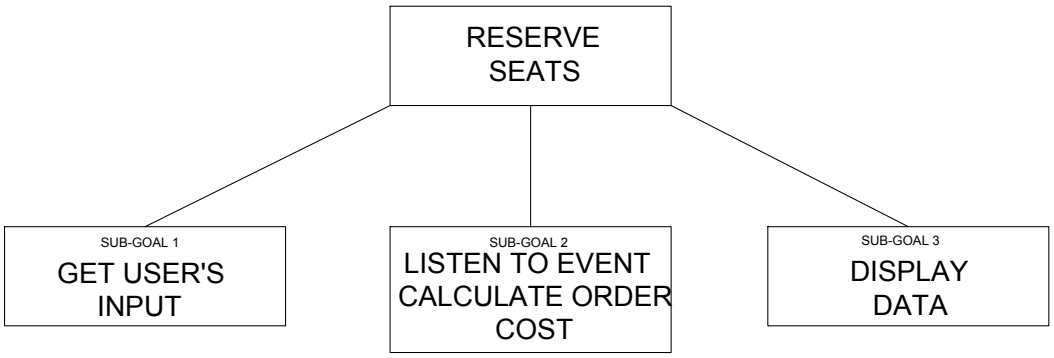
Output (unknowns):

Name	Description	Origin	Used in Sub-goal #
Full Name	User's full name	Screen	1
Price	Price to the ticket	Screen	1
Movie type	Collection of movies available	Screen	1
Seats	Number of seats available	Screen	1

3. Designing the Solution

3.1 Structure Chart

First Level Decomposition



The first level decomposition includes three main goals of this program.

1. Get user's input
2. Listen to the event and get the data from combo box.Calculate the ticket cost
3. Display data.

Goal Refinement

Sub-goal 1

Get data from the user.

Sub-goal 1.1

Create Movie class that includes all required fields.

Sub-goal 1.2

Create OrderHandler class that includes all required methods.

Sub-goal 2

Listen to the event and calculate order cost

Sub-goal 2.1

Implement Event Handler

Sub-goal 2.2

Create Handle method that is responsible for the logic.

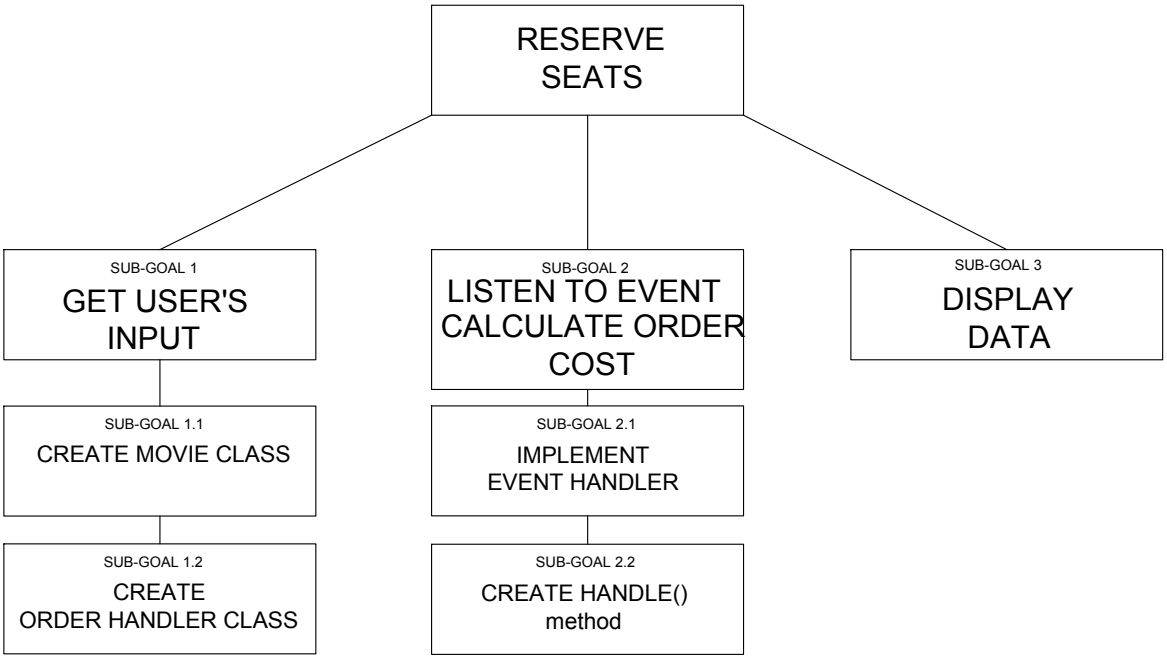
Sub-goal 3

Display results.

Sub-goal 3.1

Create Main method to display to display order.

Second Level Decomposition



The second level decomposition displays more detailed process. The first sub goal consists 2 sub-goals: creation of 2 classes. The second sub-goal features implementing an event handler interface and creating handle method. The third sub-goal focuses on printing all the data on the screen.

3.2 Module and Data Specifications

Name: Prompt user to enter a full name.
Input: Full name of user
Output: None

Logic: Store user's name in String name variable.

Name: Display for user to choose data: price, movie, and seat as drop downs. (combobox)
Input: User's choice of price, movie, and seats
Output: None
Logic:Store user's data in an observable lists.
Name: Provide user with 'place order' or 'cancel order' buttons if he/she wants to reserve or cancel the order.
Input: Depends of users .
Output: Depends of users .
Logic: If user presses "Place Order" then the program will print the order summary. If user presses "Cancel Order" then all the input boxes will be zero out.

3.3 Algorithm

Logic

1.0 Display GUI of movie ticket ordering system.
2.0 Get values from user's input.
3.0 Capture user inputs in array lists and variables.
4.0 Calculate the total order cost
5.0 Display results

Algorithm Description

Create and display a graphical user interface and collect user data. Create the event handler class to listen for user's interaction. Capture users input and store them in array lists. Calculate the order total cost and display it on the screen as an order summary.

4. Translation

4.1 Source Code

```
1 package asst2;
2
3 //=====================================================
4 // Name       :  Tsagan Garyaeva
5 // SID        :  31483539
6 // Course     :  IT-114
7 // Section    :  452
8 // Instructor :  Maura Deek
9 // T.A        :
10 //=====================================================
11 //=====================================================
12 // Assignment # :  2
13 // Date         :  02/24/2019
14 //=====================================================
15 //=====================================================
16
17
18
19
20 import javafx.application.Application;
21 import javafx.collections.FXCollections;
22 import javafx.collections.ObservableList;
23 import javafx.event.ActionEvent;
24 import javafx.event.EventHandler;
25 import javafx.geometry.Insets;
26 import javafx.geometry.Pos;
27 import javafx.scene.Scene;
28 import javafx.scene.control.*;
29 import javafx.scene.image.Image;
30 import javafx.scene.image.ImageView;
31 import javafx.scene.layout.*;
32 import javafx.scene.text.Font;
33 import javafx.stage.Stage;
34 import javafx.scene.control.ToggleGroup;
35 import javafx.scene.layout.HBox;
36
37 public class Movie extends Application{
38
39   private ComboBox<String> prices;
40   private ComboBox<String> movies;
41   private ComboBox<String> seats;
42
43
44   TextArea order;
45   TextField name = new TextField();
46   ToggleGroup delGroup = new ToggleGroup();
47   private Button orderit, clearit;
48   private Label lb_order;
```

```
49
50
51
52  private ObservableList<String> price =FXCollections.observableArrayList ("Adult-
$10.50",
53  "Child-$5.25");
54
55  private ObservableList<String> movie =FXCollections.observableArrayList ("HOW TO
TRAIN YOUR DRAGON: THE HIDDEN WORLD",
56  "ALITA: BATTLE ANGEL","FIGHTING WITH MY FAMILY", "ISN'T IT ROMANTIC");
57
58  private ObservableList<String> seat =FXCollections.observableArrayList ("1", "2", "3",
"4", "5");
59
60
61  // vars
62  String result3 ="";
63  double topCost;
64  double cost;
65  double total;
66  double subtotal;
67  final double TAXRATE = 0.07 ;
68  double tax;
69  int TOTAL_SEATS = 300;
70
71
72  @SuppressWarnings({ "unchecked", "rawtypes" })
73  public void start(Stage window) throws Exception {
74
75    // area to place the various components
76    BorderPane pane = new BorderPane();
77
78    //VBox with movies types
79    VBox movies_pane = new VBox(30);
80    movies = new ComboBox(movie);
81    movies.setValue("Movie: ");
82    movies.setMaxWidth(300);
83    movies_pane.getChildren().add(movies);
84
85
86
87
88    // VBox with prices
89    VBox prices_pane = new VBox(30);
90    prices = new ComboBox(price);
91    prices.setValue("Price:");
92    prices.setMaxWidth(300);
93    prices_pane.getChildren().add(prices);
94
```

```
95
96      // VBox with seats
97      VBox seats_pane = new VBox(30);
98      seats = new ComboBox(seat);
99      seats.setValue("Seat: ");
100      seats.setMaxWidth(300);
101      seats_pane.getChildren().add(seats);
102
103
104
105
106      // main pane
107      HBox main = new HBox(20);
108      main.getStyleClass().add("main");
109      main.getChildren().addAll(prices_pane,movies_pane, seats_pane);
110      main.setAlignment(Pos.TOP_CENTER);
111      main.setPadding(new Insets(10, 10, 10, 10));
112      main.setStyle("-fx-background-color:#7408d8; -fx-text-fill:#7408d8;-fx-font-size: 17;-fx-
padding: 10px;");
113
114
115
116
117
118      // Summary pane
119      VBox order_pane = new VBox(10);
120      lb_order = new Label("Summary of your order: ");
121      lb_order.setStyle("-fx-text-fill:#ffc87b; -fx-text-fill:#f9fafc; -fx-font-size: 17;-fx-padding:
10px;");
122      order = new TextArea();
123      order.setEditable(false);
124      order.setPrefColumnCount(10);
125      order.setMinWidth(700);
126      order.setMinHeight(200);
127      order_pane.getChildren().addAll(lb_order, order);
128
129
130      HBox order_pane2 = new HBox(10);
131      order_pane2.getChildren().addAll(order_pane);
132      order_pane2.setAlignment(Pos.TOP_CENTER);
133
134      // Bottom section with buttons
135      HBox btn_pane = new HBox(20);
136      btn_pane.setAlignment(Pos.BOTTOM_CENTER);
137      orderit = new Button("Place Order");
138      clearit = new Button("Cancel Order");
139      btn_pane.getChildren().addAll(orderit, clearit);
140      btn_pane.setPadding(new Insets(10, 10, 10, 10));
141
```

```
    //VBox with orderpane and buttons
    VBox b_section = new VBox(10);
    b_section.setPadding(new Insets(10, 10, 10, 10));
    b_section.getChildren().addAll(order_pane2, btn_pane);
    btn_pane.setStyle("-fx-background-color:#7408d8; -fx-text-fill:#c44b27;-fx-font-size: 17;-fx-padding: 10px;");
    order_pane2.setStyle("-fx-background-color:#7408d8; -fx-text-fill:#f9fafc;-fx-font-size: 17;-fx-padding: 10px;");

    //form  top section
    HBox form = new HBox();
    VBox labels = new VBox();
    VBox inputs = new VBox(10);
    form.setAlignment(Pos.TOP_CENTER);
    form.setStyle("-fx-background-color:#7408d8; -fx-text-fill:#7408d8;-fx-font-size: 17;-fx-padding: 10px;");
    Label text  = new Label("Xscape Theatres");
    text.setFont(new Font("Arial",30));
    text.setStyle("-fx-text-fill: #7408d8");
    Label lblform  = new Label("Welcome!!!");
    lblform.setStyle("-fx-text-fill:#7408d8; -fx-font-size: 20px;");
    VBox welcome = new VBox(10);
    welcome.getChildren().addAll(text, lblform);
    welcome.setAlignment(Pos.TOP_CENTER);


    // define width limits
    name.setMinWidth(350);
    Label lblname= new Label("Enter Your Name: ");
    lblname.setStyle(" -fx-text-fill:#f9fafc");
    name.setPromptText("Name");
    labels.getChildren().add(lblname);

    inputs.getChildren().add(name);
    form.getChildren().addAll(labels, inputs);
    form.setPadding(new Insets(10, 10, 10, 10));

    VBox f_form = new VBox();
    f_form.setPadding(new Insets(10, 10, 10, 10));
    f_form.getChildren().addAll(welcome, form);



    // Subscribe for when the user clicks the buttons
    OrderHandler oh = new OrderHandler();
    orderit.setOnAction(oh);
    clearit.setOnAction(oh);
    clearit.setOnAction(e -> clear());
```

```java


    // Add all to the main pane
    pane.setTop( f_form );
    pane.setBottom(b_section);
    pane.setCenter(main);

    // the main window
    Scene scene = new Scene(pane, 900, 600);
    window.setTitle("Xscape Theatres");
    window.setScene(scene);
    window.show();

  }

  // Cancel the order method
  public void clear() {
    order.setText("");
    name.clear();
    TOTAL_SEATS = 300;
    prices.setValue("Pick a Ticket Price");
    movies.setValue("Pick a Movie");
    seats.setValue("Pick a Seat");
  }



  class OrderHandler implements EventHandler<ActionEvent> {

    public void handle(ActionEvent e) {
      // validate name
      String name_n = name.getText();


      if (name_n.isEmpty())
      {
        name.setText("Enter your name");
        name.setStyle("-fx-text-fill: red; -fx-font-size: 16;");
        order.setText("Please, fill the form above.");
        order.setStyle("-fx-text-fill: red; -fx-font-size: 16;");
        return;
      }

       // if event occurs
      if (e.getSource() == orderit) {
        // price
        String result = prices.getValue();
        // movies types
        String result2 = movies.getValue();
```

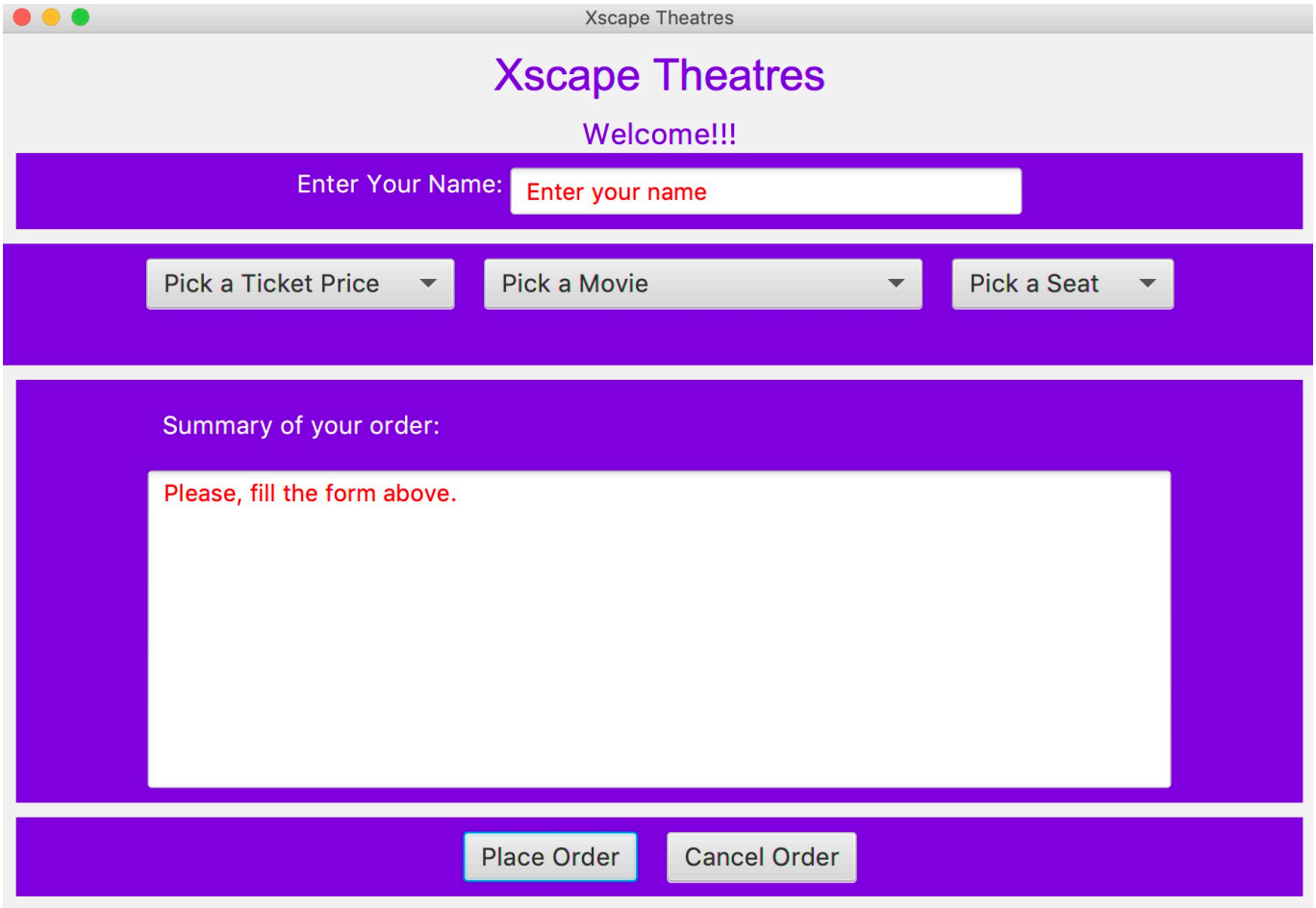
```
237         // movies seats
238         String result3 = seats.getValue();
239
240
241       // Cost of tickets
242       if (prices.getValue().equals("Adult-$10.50")){
243           cost = 10.50;
244         }
245       else if(prices.getValue().equals("Child-$5.25")){
246           cost = 5.25;
247         }
248       if(result3.equals(null)){
249           result3= "";
250         }
251         //Cost of seats
252       if (result3.contains("1")){
253           topCost=1;
254           TOTAL_SEATS-=1;
255         }
256       if (result3.contains("2")){
257            topCost=2;
258           TOTAL_SEATS-=2;
259         }
260       if (result3.contains("3")){
261           topCost=3;
262           TOTAL_SEATS-=3;
263         }
264       if (result3.contains("4")){
265            topCost=4;
266           TOTAL_SEATS-=4;
267         }
268       if (result3.contains("5")){
269           topCost=5;
270           TOTAL_SEATS-=5;
271         }
272
273
274       if (result3.contains("")){
275           order.setText("Please, pick a seat.");
276         }
277
278
279         //System.out.println(cost);
280
281         subtotal = (cost*topCost);
282         total = (subtotal* TAXRATE) + subtotal;
283         total = Math.round(total);
284
285         order.setText(
```

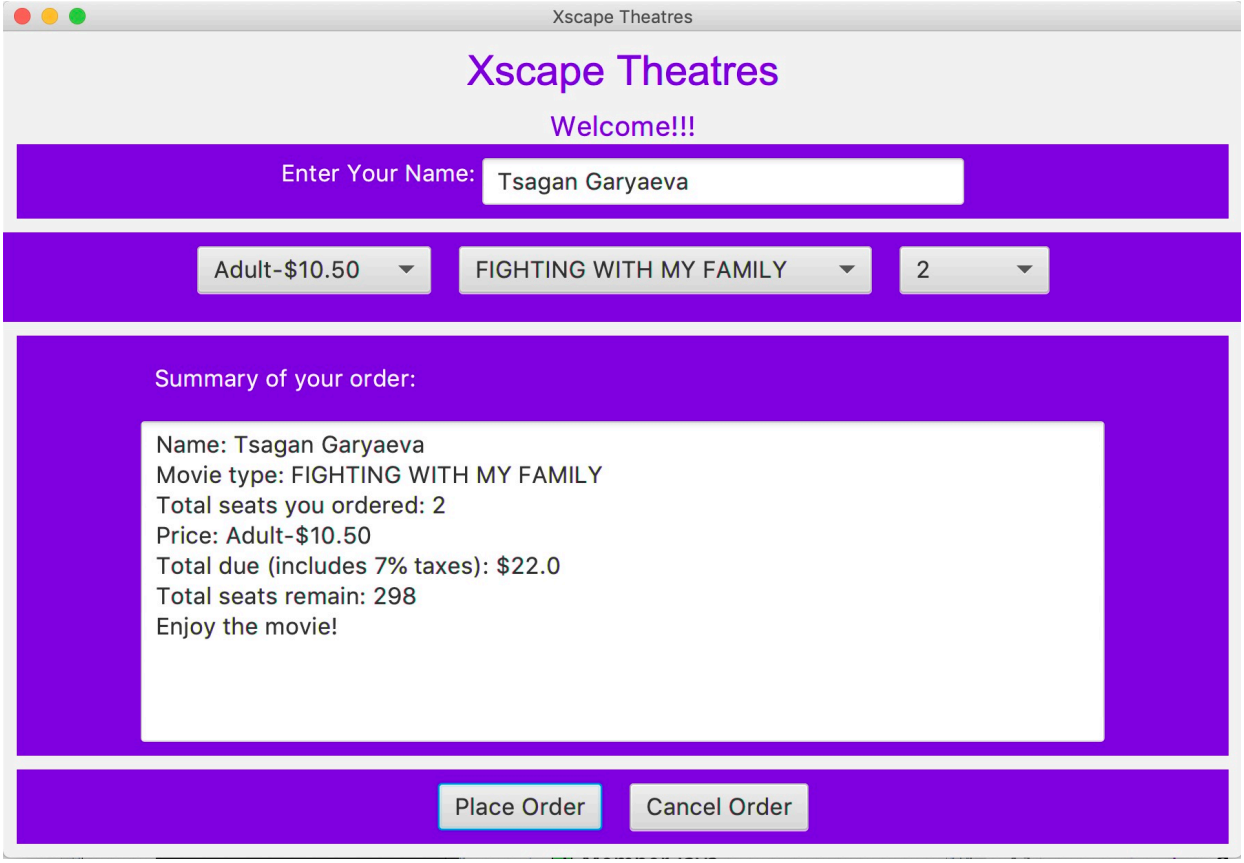
```
                        "Name: " + name.getText() + "\n" +
                        "Movie type: " + result2  + " \n"+
                        "Total seats you ordered: " + result3 + "\n" +
                        "Price: "+ result  + " \n"+
                        "Total due (includes 7% taxes): " + "$"+ total + "\n" +
                        "Total seats remain: " + TOTAL_SEATS + "\n" +
                        "Enjoy the movie!");

          } // end of logic for orderit button

        else
        {
        order.setText("Please, enter your name");
        name.clear();

        }
      }

    }


    // main method
    public static void main(String[] args) {
      Application.launch(args);
    }
  }

```

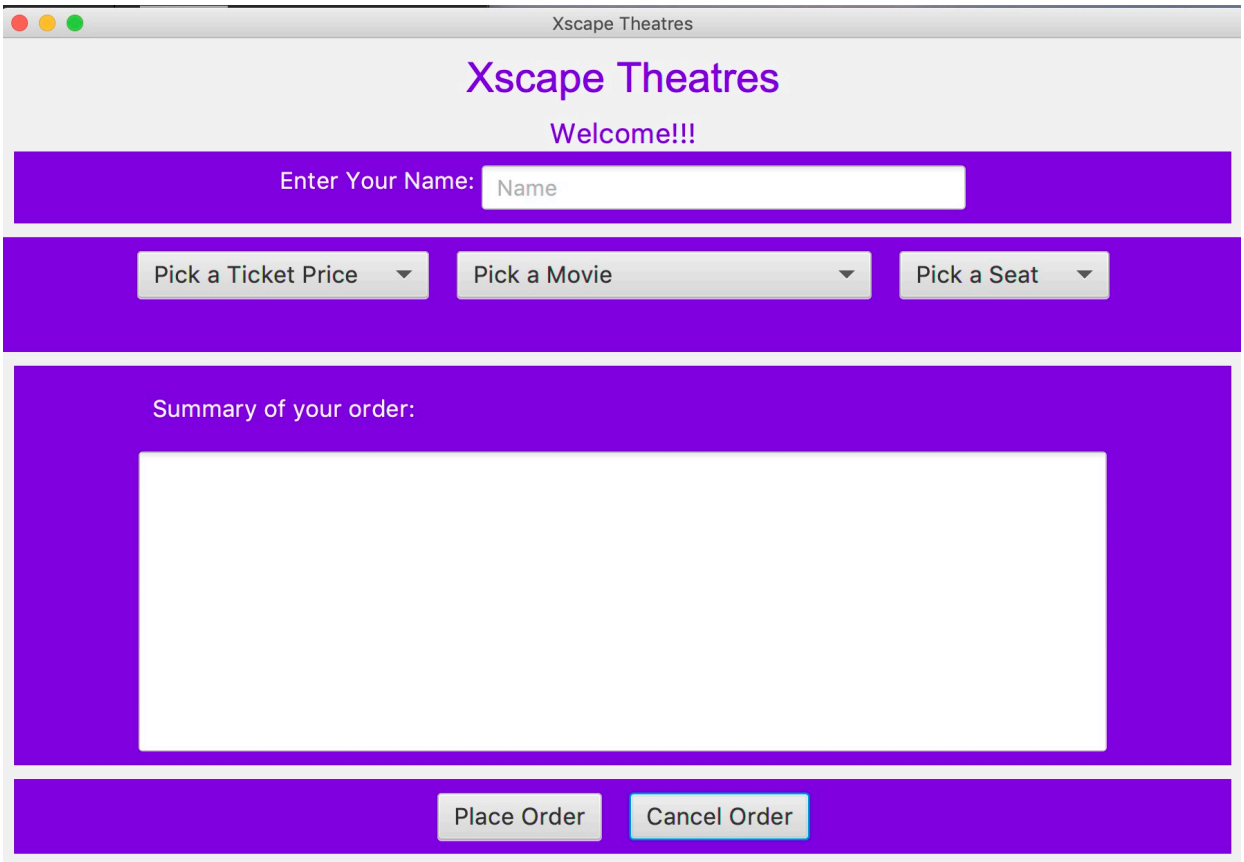
4. Solution Testing

Test the program with following data domain:
The domain range includes integers and String.

Test the program with following data:




Xscape Theatres
Xscape Theatres
Welcome!!!
Enter Your Name:
Tsagan Garyaeva
Adult-$10.50
FIGHTING WITH MY FAMILY
2
Summary of your order:
Name: Tsagan Garyaeva
Movie type: FIGHTING WITH MY FAMILY
Total seats you ordered: 2
Price: Adult-$10.50
Total due (includes 7% taxes): $22.0
Total seats remain: 298
Enjoy the movie!
Place Order
Cancel Order


Xscape Theatres
Xscape Theatres
Welcome!!!
Enter Your Name:
Name
Pick a Ticket Price
Pick a Movie
Pick a Seat
Summary of your order:
Place Order
Cancel Order