

ContentPane.java

```
1 package project;
2
3 import java.awt.CardLayout;
4 import java.awt.Color;
5 import java.awt.Dimension;
6 import java.awt.Image;
7 import java.awt.event.ActionListener;
8 import java.util.Map;
9
10 import javax.swing.BorderFactory;
11 import javax.swing.BoxLayout;
12 import javax.swing.ImageIcon;
13 import javax.swing.JButton;
14 import javax.swing.JLabel;
15 import javax.swing.JPanel;
16 import javax.swing.JScrollPane;
17 import javax.swing.JTabbedPane;
18 import javax.swing.JTable;
19 import javax.swing.border.BevelBorder;
20 import javax.swing.border.Border;
21 import javax.swing.table.TableModel;
22
23
24 /* This is one of the Main User Interface component , This is displayed as Content
   Pane in the User Interface
25 * This is one of the main constituent part of the View Component(Main User
   Interface)
26 * This displays the Multiple Forms Interface using a Tabbed Pane and also display
   the
27 * tabular list of data (Table) for Participant entity.
28 * It contains a Tool bar of Buttons on the top of the Pane and it allows the
   users to interacts with
29 * the interface through its buttons */
30
31 public class ContentPane extends JPanel {
32
33     private static final long serialVersionUID = 1L;
34     JPanel contentPanel;
35     JPanel toolPanel; //Tool Panel where the buttons are
   displayed
36     JTabbedPane tabbedPane; //This contains all the forms to be
   displayed
37     TableModel model; // Tabular list data
38     JTable table; // Tabular list to be displayed
39     Map<String,GenericForm> forms; // Forms to be displayed
40     /*All the Buttons in the Tool Bar */
41     JButton formSaveButton,tableExpandButton,formExpandButton,resetButton;
42     String acl;
43     Border emptyBorder=BorderFactory.createEmptyBorder(5, 5, 5, 5);
44     Border compoundBorder=BorderFactory.createCompoundBorder(new CustomBorder(2,
   Color.BLUE), emptyBorder);
45
46     /* The Forms to be embedded into the Tabbed Pane are passed as Parameter*/
47     public ContentPane(Map<String,GenericForm> forms,String acl){
48         super();
```

ContentPane.java

```

49     this.forms=forms;
50     this.acl=acl;
51     setLayout(new CardLayout(0, 0));
52     setBackground(new Color(186,219,241));
53     setBorder(compoundBorder);
54     createGUI();                                     //Render all the
components.
55     }
56
57     /* This will render all the components such as Tabbed Pane,Table,Tool Panel */
58     public void createGUI(){
59         renderTabbedPane();
60         renderTable();
61         contentPanel=new JPanel();
62         contentPanel.setBackground(new Color(186,219,241));
63         contentPanel.setLayout(new BoxLayout(contentPanel, BoxLayout.Y_AXIS));
64         toolPanel=renderToolPanel();
65         contentPanel.add(toolPanel);
66         contentPanel.add(tabbedPane);
67         contentPanel.add(new JScrollPane(table));
68         add(getHomeLogo(),"home");
69         add(contentPanel,"FormTable");
70     }
71
72     /* Get the Home Page Logo*/
73     public JLabel getHomeLogo(){
74         ImageIcon logo = new ImageIcon("logo.jpg");
75         JLabel logoLabel = new JLabel(logo);
76         //JLabel logoLabel = new JLabel("logo");
77         return logoLabel;
78     }
79
80     /*This will render the Tabbed Pane where the forms are displayed */
81     public void renderTabbedPane(){
82         tabbedPane=new JTabbedPane();
83         tabbedPane.setBorder(BorderFactory.createLineBorder(Color.GRAY,1));
84         tabbedPane.setTabPlacement(JTabbedPane.RIGHT);
85         /* Going through the collection of forms and add into the Tabbed Pane
excluding
86         * the Search and Logon forms */
87         for (String key : forms.keySet()){
88             if (!(key.equalsIgnoreCase("Logon") ||key.equalsIgnoreCase("Search")
89             )){
90                 GenericForm form=forms.get(key);
91                 JScrollPane sPane=new JScrollPane(form);
92                 sPane.setBorder(null);
93                 tabbedPane.add(key, sPane);
94             }
95         }
96
97         /* This will render the Tool Panel with all the Buttons */
98         public JPanel renderToolPanel(){
99             JPanel toolPanel=new JPanel();
100            toolPanel.setBorder(BorderFactory.createBevelBorder(BevelBorder.RAISED));

```

ContentPane.java

```
101     toolPanel.getInsets().top=0;
102     toolPanel.getInsets().bottom=0;
103     toolPanel.setMaximumSize(new Dimension(1200, 80));
104     ImageIcon icon = new ImageIcon("save-96.png","Save");
105     Image img=icon.getImage().getScaledInstance(20, 20, ALLBITS);
106
107     formSaveButton=new JButton(new ImageIcon(img));
108     formSaveButton.setActionCommand("FormSave");
109
110     formExpandButton=new JButton("Form Expand");
111     formExpandButton.setActionCommand("Form Expand");
112
113     tableExpandButton=new JButton("Table Expand");
114     tableExpandButton.setActionCommand("Table Expand");
115
116     resetButton=new JButton("Reset");
117     resetButton.setActionCommand("Reset");
118     if (!acl.equalsIgnoreCase("Reader")){
119         toolPanel.add(formSaveButton);
120     }
121     toolPanel.add(formExpandButton);
122     toolPanel.add(tableExpandButton);
123     toolPanel.add(resetButton);
124     return toolPanel;
125 }
126
127 /* Render the Tabular list (Table) */
128 public void renderTable(){
129     table=new Table();
130 }
131
132 /* Getters and Setters */
133 public JPanel getContentPane() {
134     return contentPanel;
135 }
136
137 public TableModel getModel() {
138     return model;
139 }
140
141 public void setModel(TableModel model) {
142     this.model = model;
143 }
144
145 public JTable getTable() {
146     return table;
147 }
148
149 public void setTable(JTable table) {
150     this.table = table;
151 }
152
153 public JTabbedPane getTabbedPane() {
154     return tabbedPane;
155 }
```

ContentPane.java

```
156
157     public JPanel getToolPanel() {
158         return toolPanel;
159     }
160     /* End Of Getters and Setters */
161
162     /* This allow the other components to register their Handlers(or Listeners)
with the buttons in the
163     * Tool Panel. Any button click on the Tool Panel trigger this registerd
handlers to respond */
164     public void addActionListener(ActionListener listener){
165         formSaveButton.addActionListener(listener);
166         tableExpandButton.addActionListener(listener);
167         formExpandButton.addActionListener(listener);
168         resetButton.addActionListener(listener);
169     }
170
171
172
173 }
174
```