

```

Loading vc-cvs...
1 package caltool.schedule_ui;
2
3 import caltool.schedule.*;
4 import caltool.caltool_ui.*;
5 import mvp.*;
6 import javax.swing.*;
7 import java.awt.*;
8 import java.awt.event.*;
9
10 public class CategoriesEditor extends CalendarToolWindow {
11
12     public CategoriesEditor(Screen screen, Model model,
13         CalendarToolUI calToolUI) {
14         super(screen, model, calToolUI);
15     }
16
17     public Component compose() {
18         JPanel panel = new JPanel();
19         window.add(panel);
20         panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
21
22         panel.add(Box.createVerticalStrut(15));
23         panel.add(composeList());
24         panel.add(Box.createVerticalStrut(25));
25         panel.add(composeButtonRow1());
26         panel.add(Box.createVerticalStrut(15));
27         panel.add(composeButtonRow2());
28         panel.add(Box.createVerticalStrut(15));
29
30         window.setTitle("Edit Categories");
31         window.pack();
32
33         return window;
34     }
35
36     /**
37     * Compose the labeled list of categories.
38     */
39     protected Box composeList() {
40         Box hbox = Box.createHorizontalBox();
41         Box vbox = Box.createVerticalBox();
42
43         JLabel label = new JLabel("Current Categories:");
44         label.setAlignmentX(Component.LEFT_ALIGNMENT);
45         vbox.add(label);
46
47         /*
48         * Build a list with some dummy blank elements. This will be refined
49         * later to add model data.
50         */
51         String[] dummyData = {
52             "",
53             "",
54             "",
55             ""
56         };
57
58         /*
59         *
60         *
61         *
62         */
63         JList list = new JList(dummyData);
64         list.setAlignmentX(Component.LEFT_ALIGNMENT);
65         list.setBorder(BorderFactory.createLineBorder(Color.black));
66         vbox.add(list);
67
68         hbox.add(Box.createHorizontalStrut(25));
69         hbox.add(vbox);
70         hbox.add(Box.createHorizontalStrut(25));
71         return hbox;
72     }
73
74     /**
75     * Compose the first row of buttons, consisting of Add, Change, and Delete.
76     */
77     protected Box composeButtonRow1() {
78
79         Box hbox = Box.createHorizontalBox();
80
81         /*
82         * Construct the three buttons.
83         */
84         JButton addButton = new JButton("Add");
85         JButton changeButton = new JButton("Change");
86         JButton deleteButton = new JButton("Delete");
87
88         /*
89         * Attach the appropriate action listeners to each button.
90         */
91         addButton.addActionListener(
92             new CategoriesAddButtonListener((Schedule) model, this));
93
94         changeButton.addActionListener(
95             new CategoriesChangeButtonListener((Schedule) model, this));
96
97         deleteButton.addActionListener(
98             new CategoriesDeleteButtonListener((Schedule) model, this));
99
100        /*
101        * Add them to the hbox and return it.
102        */
103        hbox.add(Box.createHorizontalStrut(15));
104        hbox.add(addButton);
105        hbox.add(Box.createHorizontalStrut(20));
106        hbox.add(changeButton);
107        hbox.add(Box.createHorizontalStrut(20));
108        hbox.add(deleteButton);
109        hbox.add(Box.createHorizontalStrut(15));
110        return hbox;
111    }

```

```
112     }
113
114     /**
115     * Compose the second row of buttons, consisting of Save As, Load, and
116     * Done.
117     */
118     protected Box composeButtonRow2() {
119         Box hbox = Box.createHorizontalBox();
120
121         /*
122         * Construct the buttons.
123         */
124         JButton saveAsButton = new JButton("Save As ...");
125         JButton loadButton = new JButton("Load ...");
126         JButton doneButton = new JButton("Done");
127
128         /*
129         * Attach the appropriate action listeners to each button.
130         */
131         saveAsButton.addActionListener(
132             new CategoriesSaveAsButtonListener((Schedule) model, this));
133
134         loadButton.addActionListener(
135             new CategoriesLoadButtonListener((Schedule) model, this));
136
137         doneButton.addActionListener(
138             new ActionListener() {
139                 public void actionPerformed(ActionEvent e) {
140                     hide();
141                 }
142             }
143         );
144
145         /*
146         * Add them to the hbox and return it.
147         */
148         hbox.add(Box.createHorizontalStrut(15));
149         hbox.add(saveAsButton);
150         hbox.add(Box.createHorizontalStrut(20));
151         hbox.add(loadButton);
152         hbox.add(Box.createHorizontalStrut(20));
153         hbox.add(doneButton);
154         hbox.add(Box.createHorizontalStrut(15));
155         return hbox;
156     }
157 }
158
159 }
```