

```
1 package application;
2
3
4 import java.net.URL;
5 import java.sql.Date;
6 import java.util.ResourceBundle;
7
8 import javafx.event.ActionEvent;
9 import javafx.fxml.FXML;
10 import javafx.scene.control.Button;
11 import javafx.scene.control.ChoiceBox;
12 import javafx.scene.control.DatePicker;
13 import javafx.scene.control.ListView;
14 import javafx.scene.control.RadioButton;
15 import javafx.scene.control.SelectionMode;
16 import javafx.scene.control.SelectionModel;
17 import javafx.scene.control.TextField;
18 import javafx.scene.control.ToggleGroup;
19 import javafx.scene.layout.AnchorPane;
20
21 public class AddTransactionController {
22     @FXML
23     private TextField transactionTitle;
24
25     @FXML
26     private TextField transactionDes;
27
28     @FXML
29     private TextField transactionAmount;
30
31     @FXML
32     private ChoiceBox<String> incomeCB;
33
34     @FXML
35     private Button incomeSubmit;
36
37     @FXML
38     private ChoiceBox<String> expenseCB;
39
40     @FXML
```

```
41     private Button expenseSubmit;
42
43     @FXML
44     private ChoiceBox<String> SharedCB;
45
46     @FXML
47     private Button sharedSubmit;
48
49     @FXML
50     private ListView<String> sharedList;
51
52     @FXML
53     private DatePicker transactionDate;
54     @FXML
55     private AnchorPane incomePane;
56
57     @FXML
58     private AnchorPane expensePane;
59
60     @FXML
61     private AnchorPane sharedPane;
62
63     @FXML
64     private RadioButton incomeButton;
65
66     @FXML
67     private RadioButton expenseButton;
68
69     @FXML
70     private RadioButton sharedButton;
71
72
73     /**
74     * initialize - sets up AddTransaction.fxml
75     */
76     @FXML
77     private void initialize() {
78         ToggleGroup group = new ToggleGroup();
79         incomeButton.setToggleGroup(group);
80         expenseButton.setToggleGroup(group);
```

```
81         sharedButton.setToggleGroup(group);
82
83         incomePane.setVisible(true);
84         expensePane.setVisible(false);
85         sharedPane.setVisible(false);
86
87         this.incomeCB.getItems().addAll(Income.categoryOptions);
88         this.expenseCB.getItems().addAll(Expense.categoryOptions);
89         this.SharedCB.getItems().addAll(Expense.categoryOptions);
90         this.sharedList.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);
91         this.sharedList.getItems().addAll(SharedExpense.getAvailableUsers());
92     }
93
94     /**
95      * incomePane – displays incomePane and hides the rest
96      * @param event
97      */
98     public void incomePane(ActionEvent event) {
99         incomePane.setVisible(true);
100        expensePane.setVisible(false);
101        sharedPane.setVisible(false);
102    }
103    /**
104     * expensePane – displays expensePane and hides the rest
105     * @param event
106     */
107    public void expensePane(ActionEvent event) {
108        incomePane.setVisible(false);
109        expensePane.setVisible(true);
110        sharedPane.setVisible(false);
111    }
112    /**
113     * sharePane – displays sharePane and hides the rest
114     * @param event
115     */
116    public void sharePane(ActionEvent event) {
117        incomePane.setVisible(false);
118        expensePane.setVisible(false);
119        sharedPane.setVisible(true);
120    }
```

```
121
122 /**
123  * SubmitIncome - adds all the data entered into the database correctly.
124  * @param event
125  */
126 public void SubmitIncome(ActionEvent event) {
127     System.out.println("*****");
128     Income income = new Income();
129     income.setTitle(this.transactionTitle.getText());
130     income.setDate(Date.valueOf(this.transactionDate.getValue()));
131     income.setAmount(Double.valueOf(transactionAmount.getText()));
132     income.setDescription(this.transactionDes.getText());
133     income.setIncomeCategory(this.incomeCB.getValue());
134     income.setUserName(UserHolder.getInstance().getName());
135     System.out.println("*****");
136     income.save();
137 }
138
139 /**
140  * SubmitExpense - adds all the data entered into the database correctly.
141  * @param event
142  */
143 public void SubmitExpense(ActionEvent event) {
144     System.out.println("*****E*****");
145     Expense expense = new Expense();
146     expense.setTitle(this.transactionTitle.getText());
147     expense.setDate(Date.valueOf(this.transactionDate.getValue()));
148     expense.setAmount(Double.valueOf(transactionAmount.getText()));
149     expense.setDescription(this.transactionDes.getText());
150     expense.setExpenseCategory(this.incomeCB.getValue());
151     expense.setUserName(UserHolder.getInstance().getName());
152     expense.save();
153 }
154
155 /**
156  * SubmitSharedExpense - adds all the data entered into the database correctly.
157  * @param event
158  */
159 public void SubmitSharedExpense(ActionEvent event) {
160     System.out.println("*****SE*****");
```

```
161     SharedExpense expense = new SharedExpense();
162     expense.setTitle(this.transactionTitle.getText());
163     expense.setDate(Date.valueOf(this.transactionDate.getValue()));
164     expense.setAmount(Double.valueOf(transactionAmount.getText()));
165     expense.setDescription(this.transactionDes.getText());
166     expense.setExpenseCatagory(this.incomeCB.getValue());
167     expense.setUserName(UserHolder.getInstance().getName());
168     expense.setSharedWith(this.sharedList.getSelectionModel().getSelectedItem());
169     expense.save();
170 }
171 }
172
173
```