```java
package view;

import java.awt.Color;

/**
 * Class AddCaseInterface to construct interface for user to add a new Case.
 * Interacts with NewCaseController to ensure data input is valid.
 * @author Daria Vekic (Student ID: 586661)
 *
 */
public class AddCaseInterface extends JFrame {

    private static final long serialVersionUID = 1L;
    //Instance fields for AddCaseInterface
    private JPanel contentPane;
    private JTextField fNameTxtField;
    private JTextField lNameTxtField;
    private JTextField dobTxtField;
    private JTextField addressFirstLineTxtField;
    private JTextField addressSecondLineTxtField;
    private JTextField postcodeTxtField;
    private JTextField phoneNumTxtField;
    private JTextField caseTitleTxtField;
    private JTextArea caseDescriptionTxtArea;
    private JComboBox caseTypeComboBox;
    private JComboBox enqTypeComboBox;
    private JComboBox solComboBox;
    private JButton confirmCaseButton;
    private CommonElements commonElements = new CommonElements(); //to access common elements
    private NewCaseController controller = new NewCaseController(); //for Controller interaction


    /**
     * Constructor method that calls createNewCaseForm() method.
     * @param employees the List of Employees to be searched.
     * @param clients the List of Clients to be added to.
     * @param cases the List of Cases to be added to.
     */
    public AddCaseInterface(ArrayList<Employee> employees, ArrayList<Client> clients, ArrayList<Case> cases) {
        createNewCaseForm(employees, clients, cases);
    } //end constructor method


    /**
     * Method to construct the GUI.
     * @param employees the List of Employees to be searched.
     * @param clients the List of Clients to be added to.
     * @param cases the List of Cases to be added to.
     */
    private void createNewCaseForm(ArrayList<Employee> employees, ArrayList<Client> clients, ArrayList<Case> cases) {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 1100, 710);
        contentPane = new JPanel();
        contentPane.setBackground(Color.DARK_GRAY);
```

```
 79         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
 80         setContentPane(contentPane);
 81         contentPane.setLayout(null);
 82 
 83         //add the navigation menu to the GUI
 84         commonElements.constructNavMenu(contentPane, employees, clients, cases);
 85 
 86         JLabel caseFormLabel = new JLabel("Case Details");
 87         caseFormLabel.setFont(new Font("Verdana", Font.BOLD, 16));
 88         caseFormLabel.setForeground(Color.WHITE);
 89         caseFormLabel.setBounds(223, 130, 126, 22);
 90         contentPane.add(caseFormLabel);
 91 
 92         JLabel clientFormLabel = new JLabel("Client Details");
 93         clientFormLabel.setForeground(Color.WHITE);
 94         clientFormLabel.setFont(new Font("Verdana", Font.BOLD, 16));
 95         clientFormLabel.setBounds(502, 130, 126, 22);
 96         contentPane.add(clientFormLabel);
 97 
 98         JLabel solicitorFormLabel = new JLabel("Solicitor Details");
 99         solicitorFormLabel.setForeground(Color.WHITE);
100         solicitorFormLabel.setFont(new Font("Verdana", Font.BOLD, 16));
101         solicitorFormLabel.setBounds(835, 130, 155, 22);
102         contentPane.add(solicitorFormLabel);
103 
104         caseTypeComboBox = new JComboBox();
105         caseTypeComboBox.setForeground(Color.BLACK);
106         caseTypeComboBox.setFont(new Font("Verdana", Font.ITALIC, 10));
107         caseTypeComboBox.setModel(new DefaultComboBoxModel(CaseType.values()));
108         caseTypeComboBox.setBounds(299, 163, 140, 33);
109         caseTypeComboBox.addActionListener(new ActionListener()
110         {
111             public void actionPerformed(ActionEvent event) {
112                 caseTypeComboBoxActionPerformed(event);
113             } //end method actionPerformed
114         } //end method ActionListener
115         ); //end call to caseTypeComboBoxActionPerformed ActionListener
116         contentPane.add(caseTypeComboBox);
117 
118         JLabel caseTypeLabel = new JLabel("Case Type:");
119         caseTypeLabel.setFont(new Font("Verdana", Font.BOLD, 12));
120         caseTypeLabel.setForeground(Color.WHITE);
121         caseTypeLabel.setBounds(223, 174, 73, 14);
122         contentPane.add(caseTypeLabel);
123 
124         JLabel fNameLabel = new JLabel("Client First Name:");
125         fNameLabel.setForeground(Color.WHITE);
126         fNameLabel.setFont(new Font("Verdana", Font.BOLD, 12));
127         fNameLabel.setBounds(471, 174, 126, 14);
128         contentPane.add(fNameLabel);
129 
130         fNameTxtField = new JTextField();
131         fNameTxtField.addMouseListener(new MouseAdapter() {
132             @Override
```

```
133             public void mouseClicked(MouseEvent e) {
134                 fNameTxtField.setText("");
135             } //end method mouseClicked
136         } //end method MouseAdapter
137         ); //end call to MouseListener
138         fNameTxtField.setForeground(Color.GRAY);
139         fNameTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
140         fNameTxtField.setText("Enter first name");
141         fNameTxtField.setToolTipText("Name must start with capital letter and cannot contain numbers");
142         fNameTxtField.setBounds(593, 166, 140, 33);
143         contentPane.add(fNameTxtField);
144         fNameTxtField.setColumns(10);
145 
146         JLabel lNameLabel = new JLabel("Client Last Name:");
147         lNameLabel.setForeground(Color.WHITE);
148         lNameLabel.setFont(new Font("Verdana", Font.BOLD, 12));
149         lNameLabel.setBounds(471, 218, 126, 14);
150         contentPane.add(lNameLabel);
151 
152         lNameTxtField = new JTextField();
153         lNameTxtField.setText("Enter last name");
154         lNameTxtField.setToolTipText("Name must start with capital letter and cannot contain numbers");
155         lNameTxtField.setForeground(Color.GRAY);
156         lNameTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
157         lNameTxtField.setColumns(10);
158         lNameTxtField.setBounds(593, 210, 140, 33);
159         lNameTxtField.addMouseListener(new MouseAdapter() {
160             @Override
161             public void mouseClicked(MouseEvent e) {
162                 lNameTxtField.setText("");
163             } //end method mouseClicked
164         } //end method MouseAdapter
165         ); //end call to MouseListener
166         contentPane.add(lNameTxtField);
167 
168         JLabel dobLabel = new JLabel("Client DOB:");
169         dobLabel.setForeground(Color.WHITE);
170         dobLabel.setFont(new Font("Verdana", Font.BOLD, 12));
171         dobLabel.setBounds(510, 275, 87, 14);
172         contentPane.add(dobLabel);
173 
174         dobTxtField = new JTextField();
175         dobTxtField.setText("DD/MM/YYYY");
176         dobTxtField.setToolTipText("DOB must be entered in format DD/MM/YYYY");
177         dobTxtField.setForeground(Color.GRAY);
178         dobTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
179         dobTxtField.setColumns(10);
180         dobTxtField.setBounds(593, 267, 140, 33);
181         dobTxtField.addMouseListener(new MouseAdapter() {
182             @Override
183             public void mouseClicked(MouseEvent e) {
184                 dobTxtField.setText("");
185             } //end method mouseClicked
186         } //end method MouseAdapter
```

```
187         ); //end call to MouseListener
188         contentPane.add(dobTxtField);
189 
190         JLabel addressLabel = new JLabel("Client address:");
191         addressLabel.setForeground(Color.WHITE);
192         addressLabel.setFont(new Font("Verdana", Font.BOLD, 12));
193         addressLabel.setBounds(487, 335, 110, 14);
194         contentPane.add(addressLabel);
195 
196         addressFirstLineTxtField = new JTextField();
197         addressFirstLineTxtField.setText("Address line 1");
198         addressFirstLineTxtField.setToolTipText("Enter Client's first line of address");
199         addressFirstLineTxtField.setForeground(Color.GRAY);
200         addressFirstLineTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
201         addressFirstLineTxtField.setColumns(10);
202         addressFirstLineTxtField.setBounds(593, 327, 140, 33);
203         addressFirstLineTxtField.addMouseListener(new MouseAdapter() {
204             @Override
205             public void mouseClicked(MouseEvent e) {
206                 addressFirstLineTxtField.setText("");
207             } //end method mouseClicked
208         } //end method MouseAdapter
209         ); //end call to MouseListener
210         contentPane.add(addressFirstLineTxtField);
211 
212         addressSecondLineTxtField = new JTextField();
213         addressSecondLineTxtField.setText("Address line 2");
214         addressSecondLineTxtField.setToolTipText("Enter Client's second line of address");
215         addressSecondLineTxtField.setForeground(Color.GRAY);
216         addressSecondLineTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
217         addressSecondLineTxtField.setColumns(10);
218         addressSecondLineTxtField.setBounds(593, 363, 140, 33);
219         addressSecondLineTxtField.addMouseListener(new MouseAdapter() {
220             @Override
221             public void mouseClicked(MouseEvent e) {
222                 addressSecondLineTxtField.setText("");
223             } //end method mouseClicked
224         } //end method MouseAdapter
225         ); //end call to MouseListener
226         contentPane.add(addressSecondLineTxtField);
227 
228         postcodeTxtField = new JTextField();
229         postcodeTxtField.setText("Enter postcode");
230         postcodeTxtField.setToolTipText("UK postcodes only. Case insensitive and no space required");
231         postcodeTxtField.setForeground(Color.GRAY);
232         postcodeTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
233         postcodeTxtField.setColumns(10);
234         postcodeTxtField.setBounds(593, 407, 140, 33);
235         postcodeTxtField.addMouseListener(new MouseAdapter() {
236             @Override
237             public void mouseClicked(MouseEvent e) {
238                 postcodeTxtField.setText("");
239             } //end method mouseClicked
240         } //end method MouseAdapter
```

```
241         ); //end call to MouseListener
242         contentPane.add(postcodeTxtField);
243 
244         JLabel postcodeLabel = new JLabel("Client postcode:");
245         postcodeLabel.setForeground(Color.WHITE);
246         postcodeLabel.setFont(new Font("Verdana", Font.BOLD, 12));
247         postcodeLabel.setBounds(481, 415, 116, 14);
248         contentPane.add(postcodeLabel);
249 
250         JLabel phoneNumLabel = new JLabel("Client phone no.:");
251         phoneNumLabel.setForeground(Color.WHITE);
252         phoneNumLabel.setFont(new Font("Verdana", Font.BOLD, 12));
253         phoneNumLabel.setBounds(481, 478, 116, 14);
254         contentPane.add(phoneNumLabel);
255 
256         phoneNumTxtField = new JTextField();
257         phoneNumTxtField.setText("Enter phone no.");
258         phoneNumTxtField.setToolTipText("Must start with 0");
259         phoneNumTxtField.setForeground(Color.GRAY);
260         phoneNumTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
261         phoneNumTxtField.setColumns(10);
262         phoneNumTxtField.setBounds(593, 470, 140, 33);
263         phoneNumTxtField.addMouseListener(new MouseAdapter() {
264             @Override
265             public void mouseClicked(MouseEvent e) {
266                 phoneNumTxtField.setText("");
267             } //end method mouseClicked
268         } //end method MouseAdapter
269         ); //end call to MouseListener
270         contentPane.add(phoneNumTxtField);
271 
272         JLabel enqTypeLabel = new JLabel("Enquiry Type:");
273         enqTypeLabel.setForeground(Color.WHITE);
274         enqTypeLabel.setFont(new Font("Verdana", Font.BOLD, 12));
275         enqTypeLabel.setBounds(497, 532, 110, 14);
276         contentPane.add(enqTypeLabel);
277 
278         enqTypeComboBox = new JComboBox();
279         enqTypeComboBox.setModel(new DefaultComboBoxModel(CaseType.values()));
280         enqTypeComboBox.setToolTipText("Select a type from the list");
281         enqTypeComboBox.setForeground(Color.BLACK);
282         enqTypeComboBox.setFont(new Font("Verdana", Font.ITALIC, 10));
283         enqTypeComboBox.setBounds(593, 524, 140, 33);
284         enqTypeComboBox.addActionListener(new ActionListener()
285         {
286             public void actionPerformed(ActionEvent event) {
287                 enqTypeComboBoxActionPerformed(event);
288             } //end method actionPerformed
289         } //end method ActionListener
290         ); //end call to enqTypeComboBoxActionPerformed ActionListener
291         contentPane.add(enqTypeComboBox);
292 
293         JLabel solLabel = new JLabel("Solicitor:");
294         solLabel.setForeground(Color.WHITE);
```

```
295         solLabel.setFont(new Font("Verdana", Font.BOLD, 12));
296         solLabel.setBounds(809, 174, 73, 14);
297         contentPane.add(solLabel);
298 
299         String[] empNames = controller.getNames(employees);
300         solComboBox = new JComboBox<Object>(empNames);
301         solComboBox.setToolTipText("Select the Solicitor that will handle this Case");
302         solComboBox.setForeground(Color.BLACK);
303         solComboBox.setFont(new Font("Verdana", Font.ITALIC, 10));
304         solComboBox.setBounds(881, 163, 140, 33);
305         contentPane.add(solComboBox);
306 
307         JLabel caseTitleLabel = new JLabel("Case Title:");
308         caseTitleLabel.setForeground(Color.WHITE);
309         caseTitleLabel.setFont(new Font("Verdana", Font.BOLD, 12));
310         caseTitleLabel.setBounds(223, 219, 73, 14);
311         caseTitleLabel.addMouseListener(new MouseAdapter() {
312             @Override
313             public void mouseClicked(MouseEvent e) {
314                 caseTitleLabel.setText("");
315             } //end method mouseClicked
316         } //end method MouseAdapter
317         ); //end call to MouseListener
318         contentPane.add(caseTitleLabel);
319 
320         caseTitleTxtField = new JTextField();
321         caseTitleTxtField.setText("Enter case title");
322         caseTitleTxtField.setToolTipText("Enter in format [Client surname] v [Opponent]");
323         caseTitleTxtField.setForeground(Color.GRAY);
324         caseTitleTxtField.setFont(new Font("Verdana", Font.PLAIN, 9));
325         caseTitleTxtField.setColumns(10);
326         caseTitleTxtField.setBounds(299, 207, 140, 33);
327         //Clear the field ready for user input
328         caseTitleTxtField.addMouseListener(new MouseAdapter() {
329             @Override
330             public void mouseClicked(MouseEvent e) {
331                 caseTitleTxtField.setText("");
332             } //end method mouseClicked
333         } //end method MouseAdapter
334         ); //end call to MouseListener
335         contentPane.add(caseTitleTxtField);
336 
337         JLabel caseDescriptionLabel = new JLabel("Description:");
338         caseDescriptionLabel.setForeground(Color.WHITE);
339         caseDescriptionLabel.setFont(new Font("Verdana", Font.BOLD, 12));
340         caseDescriptionLabel.setBounds(209, 255, 87, 14);
341         contentPane.add(caseDescriptionLabel);
342 
343         caseDescriptionTxtArea = new JTextArea();
344         caseDescriptionTxtArea.setLineWrap(true);
345         caseDescriptionTxtArea.setForeground(Color.GRAY);
346         caseDescriptionTxtArea.setFont(new Font("Verdana", Font.PLAIN, 10));
347         caseDescriptionTxtArea.setText("Enter description here");
348         caseDescriptionTxtArea.setToolTipText("Enter a brief description of the Case");
```

```
349         caseDescriptionTxtArea.setBounds(299, 251, 140, 145);
350         caseDescriptionTxtArea.addMouseListener(new MouseAdapter() {
351             @Override
352             public void mouseClicked(MouseEvent e) {
353                 caseDescriptionTxtArea.setText("");
354             } //end method mouseClicked
355         } //end method MouseAdapter
356         ); //end call to MouseListener
357         contentPane.add(caseDescriptionTxtArea);
358 
359         confirmCaseButton = new JButton("Confirm New Case");
360         confirmCaseButton.setBorder(null);
361         confirmCaseButton.setBackground(new Color(144, 238, 144));
362         confirmCaseButton.setAlignmentX(Component.CENTER_ALIGNMENT);
363         confirmCaseButton.setFont(new Font("Arial", Font.BOLD, 14));
364         confirmCaseButton.setBounds(840, 554, 181, 85);
365         confirmCaseButton.addActionListener(new ActionListener()
366         {
367             public void actionPerformed(ActionEvent event) {
368                 try {
369                     confirmCaseButtonActionPerformed(event, employees, clients, cases);
370                 } //end try
371                 catch (ParseException e) {
372                     e.printStackTrace();
373                 } //end catch
374             } //end method actionPerformed
375         } //end method ActionListener
376         ); //end call to confirmCaseButtonActionPerformed ActionListener
377         contentPane.add(confirmCaseButton);
378 
379         setLocationRelativeTo(null);
380     } //end method createNewCaseForm
381 
382 
383     /**
384      * Method to reset all fields back to their original state.
385      * Used after a new Case is successfully confirmed.
386      */
387     private void resetAllFields() {
388         caseTitleTxtField.setText("Enter case title");
389         caseTitleTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
390         caseDescriptionTxtArea.setText("Enter description here");
391         caseDescriptionTxtArea.setBorder(new EmptyBorder(5, 5, 5, 5));
392         fNameTxtField.setText("Enter first name");
393         fNameTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
394         lNameTxtField.setText("Enter last name");
395         lNameTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
396         dobTxtField.setText("DD/MM/YYYY");
397         dobTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
398         addressFirstLineTxtField.setText("Address line 1");
399         addressFirstLineTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
400         addressSecondLineTxtField.setText("Address line 2");
401         addressSecondLineTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
402         postcodeTxtField.setText("Enter postcode");
```

```
403         postcodeTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
404         phoneNumTxtField.setText("Enter phone no.");
405         phoneNumTxtField.setBorder(new EmptyBorder(5, 5, 5, 5));
406     } //end method resetAllFields
407 
408 
409     /*------------------------ACTION LISTENERS------------------------*/
410 
411 
412     /**
413      * Method to set value of Enquiry Type combo box to match Case Type Combo Box.
414      * An Enquiry Type will always match a Case Type
415      * @param event the user selects a value in the Case Type Combo Box.
416      */
417     private void caseTypeComboBoxActionPerformed(ActionEvent event) {
418         enqTypeComboBox.setSelectedItem(caseTypeComboBox.getSelectedItem());
419     } //end method caseTypeComboBoxButtonActionPerformed
420 
421 
422     /**
423      * The routine to follow if the "Confirm Case" Button is pressed.
424      * @param event the "Confirm Case" button is pressed.
425      * @param employees the List of Employee objects to be searched.
426      * @param cases the List the new Case object will be added to.
427      * @param clients the List the Client object will be added to.
428      * @throws ParseException thrown if the DOB field cannot be parsed.
429      */
430     private void confirmCaseButtonActionPerformed(ActionEvent event, ArrayList<Employee> employees,
431             ArrayList<Client> clients, ArrayList<Case> cases) throws ParseException {
432         int emptyFields = controller.checkEmpty(caseTitleTxtField, caseDescriptionTxtArea, fNameTxtField, lNameTxtField,
433                 dobTxtField, addressFirstLineTxtField, addressSecondLineTxtField, postcodeTxtField,
434                 phoneNumTxtField);
435         //if there's no empty fields
436         if(emptyFields==0) {
437             boolean added = controller.addNewCase(caseTitleTxtField, caseTypeComboBox, caseDescriptionTxtArea,
438                     solComboBox, fNameTxtField, lNameTxtField,
439                     dobTxtField, addressFirstLineTxtField, addressSecondLineTxtField,
440                     postcodeTxtField, phoneNumTxtField, employees, clients, cases);
441             if(added) { //if added is true
442                 resetAllFields();
443             } else {
444                 commonElements.showError(controller.error11, "Invalid Fields");
445             }
446         }
447         else {//else field(s) empty
448             commonElements.showError(controller.error2, "Empty fields");
449         }//end if else
450     } //end method confirmCaseButtonActionPerformed
451 
452 
453     /**
454      * Method to set value of Case combo box to match Enquiry Type Combo Box.
455      * A Case Type will always match Enquiry Type.
456      * @param event the user selects a value in the Enquiry Type Combo Box.
```

```
457      */
458     private void enqTypeComboBoxActionPerformed(ActionEvent event) {
459         caseTypeComboBox.setSelectedItem(enqTypeComboBox.getSelectedItem());
460     } //end method enqTypeComboBoxActionPerformed
461 } //end class AddCaseInterface
```