

Table of Contents

add_conditions_dialog_ui	2
amend_offense_dialog_ui	11
case_information	15
case_types	18
criminal_dialogs	20
custom_widgets	27
main_window_ui	30
minor_misdemeanor_dialog_ui	37
minor_misdemeanor_dialogs	50
party_types	63
template_types	65

```

1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file 'resources/ui/AddConditionsDialog.ui'
4  #
5  # Created by: PyQt5 UI code generator 5.15.4
6  #
7  # WARNING: Any manual changes made to this file will be lost when pyuic5 is
8  # run again. Do not edit this file unless you know what you are doing.
9
10
11 from PyQt5 importQtCore, QtGui, QtWidgets
12
13
14 class Ui\_AddConditionsDialog(object):
15     def setupUi(self, AddConditionsDialog):
16         AddConditionsDialog.setObjectName("AddConditionsDialog")
17         AddConditionsDialog.setWindowModality(QtCore.Qt.NonModal)
18         AddConditionsDialog.setEnabled(True)
19         AddConditionsDialog.resize(1000, 842)
20         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
21 QtWidgets.QSizePolicy.Preferred)
22         sizePolicy.setHorizontalStretch(0)
23         sizePolicy.setVerticalStretch(0)
24         sizePolicy.setHeightForWidth(AddConditionsDialog.sizePolicy().hasHeightForWidth())
25         AddConditionsDialog.setSizePolicy(sizePolicy)
26         AddConditionsDialog.setMinimumSize(QtCore.QSize(1000, 800))
27         AddConditionsDialog.setMaximumSize(QtCore.QSize(2500, 3500))
28         palette = QtGui.QPalette()
29         brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
30         brush.setStyle(QtCore.Qt.SolidPattern)
31         palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
32         brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
33         brush.setStyle(QtCore.Qt.SolidPattern)
34         palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
35         brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
36         brush.setStyle(QtCore.Qt.SolidPattern)
37         palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
38         brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
39         brush.setStyle(QtCore.Qt.SolidPattern)
40         palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)
41         brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
42         brush.setStyle(QtCore.Qt.SolidPattern)
43         palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)
44         brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
45         brush.setStyle(QtCore.Qt.SolidPattern)
46         palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)

```

```

47     brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
48     brush.setStyle(QtCore.Qt.SolidPattern)
49     palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)
50     brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
51     brush.setStyle(QtCore.Qt.SolidPattern)
52     palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Base, brush)
53     brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
54     brush.setStyle(QtCore.Qt.SolidPattern)
55     palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
56     AddConditionsDialog.setPalette(palette)
57     font = QtGui.QFont()
58     font.setFamily("Palatino Linotype")
59     font.setPointSize(11)
60     font.setBold(True)
61     font.setItalic(False)
62     font.setWeight(75)
63     AddConditionsDialog.setFont(font)
64     AddConditionsDialog.setToolTip("")
65     AddConditionsDialog.setStyleSheet("background-color: rgb(25, 49, 91);\n"
66 "font: 75 11pt \"Palatino Linotype\";\n"
67 "font-weight: bold;")
68     AddConditionsDialog.setModal(True)
69     self.frame_2 = QtWidgets.QFrame(AddConditionsDialog)
70     self.frame_2.setGeometry(QtCore.QRect(10, 10, 981, 111))
71     font = QtGui.QFont()
72     font.setFamily("Palatino Linotype")
73     font.setPointSize(11)
74     font.setBold(True)
75     font.setItalic(False)
76     font.setWeight(75)
77     self.frame_2.setFont(font)
78     self.frame_2.setStyleSheet("background-color: rgb(240, 240, 240);")
79     self.frame_2 setFrameShape(QtWidgets.QFrame.Box)
80     self.frame_2 setFrameShadow(QtWidgets.QFrame.Plain)
81     self.frame_2.setLineWidth(2)
82     self.frame_2.setObjectName("frame_2")
83     self.layoutWidget_2 = QtWidgets.QWidget(self.frame_2)
84     self.layoutWidget_2.setGeometry(QtCore.QRect(10, 10, 961, 91))
85     self.layoutWidget_2.setObjectName("layoutWidget_2")
86     self.charges_gridLayout = QtWidgets.QGridLayout(self.layoutWidget_2)
87     self.charges_gridLayout.setContentsMargins(0, 0, 0, 0)
88     self.charges_gridLayout.setObjectName("charges_gridLayout")
89     self.offense_label_1 = QtWidgets.QLabel(self.layoutWidget_2)
90     font = QtGui.QFont()
91     font.setFamily("Palatino Linotype")
92     font.setPointSize(11)

```

```
93     font.setBold(True)
94     font.setItalic(False)
95     font.setWeight(75)
96     self.offense_label_1.setFont(font)
97     self.offense_label_1.setWordWrap(True)
98     self.offense_label_1.setObjectName("offense_label_1")
99     self.charges_gridLayout.addWidget(self.offense_label_1, 0, 0, 1, 1)
100    self.label_14 = QtWidgets.QLabel(self.layoutWidget_2)
101    font = QtGui.QFont()
102    font.setFamily("Palatino Linotype")
103    font.setPointSize(11)
104    font.setBold(True)
105    font.setItalic(False)
106    font.setWeight(75)
107    self.label_14.setFont(font)
108    self.label_14.setObjectName("label_14")
109    self.charges_gridLayout.addWidget(self.label_14, 2, 0, 1, 1)
110    self.label_19 = QtWidgets.QLabel(self.layoutWidget_2)
111    font = QtGui.QFont()
112    font.setFamily("Palatino Linotype")
113    font.setPointSize(11)
114    font.setBold(True)
115    font.setItalic(False)
116    font.setWeight(75)
117    self.label_19.setFont(font)
118    self.label_19.setObjectName("label_19")
119    self.charges_gridLayout.addWidget(self.label_19, 1, 0, 1, 1)
120    self.continue_Button = QtWidgets.QPushButton(AddConditionsDialog)
121    self.continue_Button.setGeometry(QtCore.QRect(810, 800, 181, 31))
122    self.continue_Button.setStyleSheet("background-color: rgb(255, 255, 0);")
123    self.continue_Button.setAutoDefault(False)
124    self.continue_Button.setObjectName("continue_Button")
125    self.license_suspension_frame = QtWidgets.QFrame(AddConditionsDialog)
126    self.license_suspension_frame.setEnabled(False)
127    self.license_suspension_frame.setGeometry(QtCore.QRect(10, 130, 981, 121))
128    self.license_suspension_frame.setStyleSheet("background-color: rgb(240, 240, 240);")
129    self.license_suspension_frame setFrameShape(QtWidgets.QFrame.Box)
130    self.license_suspension_frame setFrameShadow(QtWidgets.QFrame.Plain)
131    self.license_suspension_frame.setLineWidth(2)
132    self.license_suspension_frame.setObjectName("license_suspension_frame")
133    self.label_24 = QtWidgets.QLabel(self.license_suspension_frame)
134    self.label_24.setGeometry(QtCore.QRect(10, 10, 961, 21))
135    font = QtGui.QFont()
136    font.setFamily("Palatino Linotype")
137    font.setPointSize(11)
138    font.setBold(True)
```

```
139 font.setItalic(False)
140 font.setUnderline(True)
141 font.setWeight(75)
142 self.label_24.setFont(font)
143 self.label_24.setAlignment(QtCore.Qt.AlignCenter)
144 self.label_24.setObjectName("label_24")
145 self.layoutWidget = QtWidgets.QWidget(self.license_suspension_frame)
146 self.layoutWidget.setGeometry(QtCore.QRect(10, 40, 951, 71))
147 self.layoutWidget.setObjectName("layoutWidget")
148 self.gridLayout_3 = QtWidgets.QGridLayout(self.layoutWidget)
149 self.gridLayout_3.setContentsMargins(0, 0, 0, 0)
150 self.gridLayout_3.setObjectName("gridLayout_3")
151 self.label_9 = QtWidgets.QLabel(self.layoutWidget)
152 self.label_9.setObjectName("label_9")
153 self.gridLayout_3.addWidget(self.label_9, 0, 0, 1, 1)
154 self.license_type_box = QtWidgets.QComboBox(self.layoutWidget)
155 self.license_type_box.setStyleSheet("background-color: rgb(255, 255, 255);")
156 self.license_type_box.setObjectName("license_type_box")
157 self.license_type_box.addItem("")
158 self.license_type_box.addItem("")
159 self.license_type_box.addItem("")
160 self.gridLayout_3.addWidget(self.license_type_box, 0, 1, 1, 1)
161 self.label_18 = QtWidgets.QLabel(self.layoutWidget)
162 self.label_18.setObjectName("label_18")
163 self.gridLayout_3.addWidget(self.label_18, 0, 2, 1, 1)
164 self.license_suspension_date_box = QtWidgets.QDateEdit(self.layoutWidget)
165 self.license_suspension_date_box.setStyleSheet("background-color: rgb(255, 255,
166 255);")
167 self.license_suspension_date_box.setCalendarPopup(True)
168 self.license_suspension_date_box.setObjectName("license_suspension_date_box")
169 self.gridLayout_3.addWidget(self.license_suspension_date_box, 0, 3, 1, 1)
170 self.label_21 = QtWidgets.QLabel(self.layoutWidget)
171 self.label_21.setObjectName("label_21")
172 self.gridLayout_3.addWidget(self.label_21, 1, 0, 1, 1)
173 self.term_of_suspension_box = QtWidgets.QComboBox(self.layoutWidget)
174 self.term_of_suspension_box.setStyleSheet("background-color: rgb(255, 255, 255);")
175 self.term_of_suspension_box.setObjectName("term_of_suspension_box")
176 self.term_of_suspension_box.addItem("")
177 self.term_of_suspension_box.addItem("")
178 self.term_of_suspension_box.addItem("")
179 self.term_of_suspension_box.addItem("")
180 self.gridLayout_3.addWidget(self.term_of_suspension_box, 1, 1, 1, 1)
181 self.remedial_driving_class_checkBox = QtWidgets.QCheckBox(self.layoutWidget)
182
183 self.remedial_driving_class_checkBox.setObjectName("remedial_driving_class_checkBox")
184 self.gridLayout_3.addWidget(self.remedial_driving_class_checkBox, 1, 2, 1, 2)
```

```

185     self.community_service_frame = QtWidgets.QFrame(AddConditionsDialog)
186     self.community_service_frame.setEnabled(False)
187     self.community_service_frame.setGeometry(QtCore.QRect(10, 260, 981, 91))
188     self.community_service_frame.setStyleSheet("background-color: rgb(240, 240, 240);")
189     self.community_service_frame setFrameShape(QtWidgets.QFrame.Box)
190     self.community_service_frame setFrameShadow(QtWidgets.QFrame.Plain)
191     self.community_service_frame.setLineWidth(2)
192     self.community_service_frame.setObjectName("community_service_frame")
193     self.label_26 = QtWidgets.QLabel(self.community_service_frame)
194     self.label_26.setGeometry(QtCore.QRect(10, 10, 961, 21))
195     font = QtGui.QFont()
196     font.setFamily("Palatino Linotype")
197     font.setPointSize(11)
198     font.setBold(True)
199     font.setItalic(False)
200     font.setUnderline(True)
201     font.setWeight(75)
202     self.label_26.setFont(font)
203     self.label_26.setAlignment(QtCore.Qt.AlignCenter)
204     self.label_26.setObjectName("label_26")
205     self.layoutWidget1 = QtWidgets.QWidget(self.community_service_frame)
206     self.layoutWidget1.setGeometry(QtCore.QRect(13, 39, 951, 31))
207     self.layoutWidget1.setObjectName("layoutWidget1")
208     self.gridLayout = QtWidgets.QGridLayout(self.layoutWidget1)
209     self.gridLayout.setContentsMargins(0, 0, 0, 0)
210     self.gridLayout.setObjectName("gridLayout")
211     self.label_15 = QtWidgets.QLabel(self.layoutWidget1)
212     self.label_15.setObjectName("label_15")
213     self.gridLayout.addWidget(self.label_15, 0, 0, 1, 1)
214     self.community_service_hours_ordered_box =
215     QtWidgets.QSpinBox(self.layoutWidget1)
216
217     self.community_service_hours_ordered_box.setObjectName("community_service_hours_order
218     d_box")
219     self.gridLayout.addWidget(self.community_service_hours_ordered_box, 0, 1, 1, 1)
220     self.label = QtWidgets.QLabel(self.layoutWidget1)
221     self.label.setObjectName("label")
222     self.gridLayout.addWidget(self.label, 0, 2, 1, 1)
223     self.community_service_days_to_complete_box =
224     QtWidgets.QComboBox(self.layoutWidget1)
225
226     self.community_service_days_to_complete_box.setObjectName("community_service_days_to_c
227     omplete_box")
228     self.community_service_days_to_complete_box.addItem("")
229     self.community_service_days_to_complete_box.addItem("")
230     self.community_service_days_to_complete_box.addItem("")

```

```
231     self.community_service_days_to_complete_box.addItem("")
232     self.community_service_days_to_complete_box.addItem("")
233     self.community_service_days_to_complete_box.addItem("")
234     self.community_service_days_to_complete_box.addItem("")
235     self.gridLayout.addWidget(self.community_service_days_to_complete_box, 0, 3, 1, 1)
236     self.label_22 = QtWidgets.QLabel(self.layoutWidget1)
237     self.label_22.setObjectName("label_22")
238     self.gridLayout.addWidget(self.label_22, 0, 4, 1, 1)
239     self.community_service_date_to_complete_box =
240     QtWidgets.QDateEdit(self.layoutWidget1)
241     self.community_service_date_to_complete_box.setStyleSheet("background-color:
242     rgb(255, 255, 255);")
243     self.community_service_date_to_complete_box.setCalendarPopup(True)
244
245     self.community_service_date_to_complete_box.setObjectName("community_service_date_to_c
246     omplete_box")
247     self.gridLayout.addWidget(self.community_service_date_to_complete_box, 0, 5, 1, 1)
248     self.other_conditions_frame = QtWidgets.QFrame(AddConditionsDialog)
249     self.other_conditions_frame.setEnabled(False)
250     self.other_conditions_frame.setGeometry(QtCore.QRect(10, 360, 981, 121))
251     self.other_conditions_frame.setStyleSheet("background-color: rgb(240, 240, 240);")
252     self.other_conditions_frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
253     self.other_conditions_frame.setFrameShadow(QtWidgets.QFrame.Raised)
254     self.other_conditions_frame.setObjectName("other_conditions_frame")
255     self.label_28 = QtWidgets.QLabel(self.other_conditions_frame)
256     self.label_28.setGeometry(QtCore.QRect(10, 10, 961, 21))
257     font = QtGui.QFont()
258     font.setFamily("Palatino Linotype")
259     font.setPointSize(11)
260     font.setBold(True)
261     font.setItalic(False)
262     font.setUnderline(True)
263     font.setWeight(75)
264     self.label_28.setFont(font)
265     self.label_28.setAlignment(QtCore.Qt.AlignCenter)
266     self.label_28.setObjectName("label_28")
267     self.other_conditions_plainTextEdit =
268     QtWidgets.QPlainTextEdit(self.other_conditions_frame)
269     self.other_conditions_plainTextEdit.setGeometry(QtCore.QRect(10, 40, 961, 71))
270     self.other_conditions_plainTextEdit.setStyleSheet("background-color: rgb(255, 255,
271     255);")
272     self.other_conditions_plainTextEdit.setObjectName("other_conditions_plainTextEdit")
273     self.community_control_frame = QtWidgets.QFrame(AddConditionsDialog)
274     self.community_control_frame.setEnabled(False)
275     self.community_control_frame.setGeometry(QtCore.QRect(10, 490, 981, 121))
276     self.community_control_frame.setStyleSheet("background-color: rgb(240, 240, 240);")
```



```

277     self.community_control_frame.setFrameShape(QtWidgets.QFrame.Box)
278     self.community_control_frame.setFrameShadow(QtWidgets.QFrame.Plain)
279     self.community_control_frame.setLineWidth(2)
280     self.community_control_frame.setObjectName("community_control_frame")
281     self.label_27 = QtWidgets.QLabel(self.community_control_frame)
282     self.label_27.setGeometry(QtCore.QRect(10, 10, 961, 21))
283     font = QtGui.QFont()
284     font.setFamily("Palatino Linotype")
285     font.setPointSize(11)
286     font.setBold(True)
287     font.setItalic(False)
288     font.setUnderline(True)
289     font.setWeight(75)
290     self.label_27.setFont(font)
291     self.label_27.setAlignment(QtCore.Qt.AlignCenter)
292     self.label_27.setObjectName("label_27")
293     self.layoutWidget2 = QtWidgets.QWidget(self.community_control_frame)
294     self.layoutWidget2.setGeometry(QtCore.QRect(11, 40, 501, 66))
295     self.layoutWidget2.setObjectName("layoutWidget2")
296     self.gridLayout_2 = QtWidgets.QGridLayout(self.layoutWidget2)
297     self.gridLayout_2.setContentsMargins(0, 0, 0, 0)
298     self.gridLayout_2.setObjectName("gridLayout_2")
299     self.label_23 = QtWidgets.QLabel(self.layoutWidget2)
300     self.label_23.setObjectName("label_23")
301     self.gridLayout_2.addWidget(self.label_23, 0, 0, 1, 1)
302     self.type_of_community_control_box = QtWidgets.QComboBox(self.layoutWidget2)
303     self.type_of_community_control_box.setStyleSheet("background-color: rgb(255, 255,
304 255);")
305
306     self.type_of_community_control_box.setObjectName("type_of_community_control_box")
307     self.type_of_community_control_box.addItem("")
308     self.type_of_community_control_box.addItem("")
309     self.type_of_community_control_box.addItem("")
310     self.gridLayout_2.addWidget(self.type_of_community_control_box, 0, 1, 1, 1)
311     self.label_29 = QtWidgets.QLabel(self.layoutWidget2)
312     self.label_29.setObjectName("label_29")
313     self.gridLayout_2.addWidget(self.label_29, 1, 0, 1, 1)
314     self.term_of_community_control_box = QtWidgets.QComboBox(self.layoutWidget2)
315     self.term_of_community_control_box.setStyleSheet("background-color: rgb(255, 255,
316 255);")
317
318     self.term_of_community_control_box.setObjectName("term_of_community_control_box")
319     self.term_of_community_control_box.addItem("")
320     self.term_of_community_control_box.addItem("")
321     self.term_of_community_control_box.addItem("")
322     self.term_of_community_control_box.addItem("")

```



```

323     self.gridLayout_2.addWidget(self.term_of_community_control_box, 1, 1, 1, 1)
324
325     self.retranslateUi(AddConditionsDialog)
326     self.continue_Button.pressed.connect(AddConditionsDialog.add_conditions)
327     self.continue_Button.released.connect(AddConditionsDialog.close_window)
328
329     self.community_service_days_to_complete_box.currentIndexChanged['int'].connect(AddCondi
330     tionsDialog.set_service_date)
331     QtCore.QMetaObject.connectSlotsByName(AddConditionsDialog)
332
333     def retranslateUi(self, AddConditionsDialog):
334         _translate = QtCore.QCoreApplication.translate
335         AddConditionsDialog.setWindowTitle(_translate("AddConditionsDialog", "Additional
336         Conditions"))
337         self.offense_label_1.setText(_translate("AddConditionsDialog", "Offense:"))
338         self.label_14.setText(_translate("AddConditionsDialog", "Finding:"))
339         self.label_19.setText(_translate("AddConditionsDialog", "Statute:"))
340         self.continue_Button.setText(_translate("AddConditionsDialog", "Continue"))
341         self.label_24.setText(_translate("AddConditionsDialog", "License Suspension"))
342         self.label_9.setText(_translate("AddConditionsDialog", "License type:"))
343         self.license_type_box.setItemText(0, _translate("AddConditionsDialog", "driving"))
344         self.license_type_box.setItemText(1, _translate("AddConditionsDialog", "hunting"))
345         self.license_type_box.setItemText(2, _translate("AddConditionsDialog", "concealed
346         carry"))
347         self.label_18.setText(_translate("AddConditionsDialog", "License suspended from:"))
348         self.label_21.setText(_translate("AddConditionsDialog", "Term of suspension:"))
349         self.term_of_suspension_box.setItemText(0, _translate("AddConditionsDialog", "6
350         months"))
351         self.term_of_suspension_box.setItemText(1, _translate("AddConditionsDialog", "12
352         months"))
353         self.term_of_suspension_box.setItemText(2, _translate("AddConditionsDialog", "18
354         months"))
355         self.term_of_suspension_box.setItemText(3, _translate("AddConditionsDialog", "24
356         months"))
357         self.remedial_driving_class_checkBox.setText(_translate("AddConditionsDialog",
358         "Remedial driving class required for OL return"))
359         self.label_26.setText(_translate("AddConditionsDialog", "Community Service"))
360         self.label_15.setText(_translate("AddConditionsDialog", "Hours:"))
361         self.label.setText(_translate("AddConditionsDialog", "Days to Complete:"))
362         self.community_service_days_to_complete_box.setItemText(0,
363         _translate("AddConditionsDialog", "0"))
364         self.community_service_days_to_complete_box.setItemText(1,
365         _translate("AddConditionsDialog", "30"))
366         self.community_service_days_to_complete_box.setItemText(2,
367         _translate("AddConditionsDialog", "60"))
368         self.community_service_days_to_complete_box.setItemText(3,

```

```
369 _translate("AddConditionsDialog", "90"))
370     self.community_service_days_to_complete_box.setItemText(4,
371 _translate("AddConditionsDialog", "120"))
372     self.community_service_days_to_complete_box.setItemText(5,
373 _translate("AddConditionsDialog", "150"))
374     self.community_service_days_to_complete_box.setItemText(6,
375 _translate("AddConditionsDialog", "180"))
376     self.label_22.setText(_translate("AddConditionsDialog", "Date to Complete:"))
377     self.label_28.setText(_translate("AddConditionsDialog", "Other Conditions"))
378     self.label_27.setText(_translate("AddConditionsDialog", "Community Control"))
379     self.label_23.setText(_translate("AddConditionsDialog", "Community Control Type:"))
380     self.type_of_community_control_box.setItemText(0, _translate("AddConditionsDialog",
381 "basic"))
382     self.type_of_community_control_box.setItemText(1, _translate("AddConditionsDialog",
383 "intensive"))
384     self.type_of_community_control_box.setItemText(2, _translate("AddConditionsDialog",
385 "monitored"))
386     self.label_29.setText(_translate("AddConditionsDialog", "Term of Community
387 Control:"))
388     self.term_of_community_control_box.setItemText(0, _translate("AddConditionsDialog",
389 "6 months"))
390     self.term_of_community_control_box.setItemText(1, _translate("AddConditionsDialog",
391 "12 months"))
392     self.term_of_community_control_box.setItemText(2, _translate("AddConditionsDialog",
393 "18 months"))
394     self.term_of_community_control_box.setItemText(3, _translate("AddConditionsDialog",
395 "24 months"))
```

```

1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file 'resources/ui/AmendOffenseDialog.ui'
4  #
5  # Created by: PyQt5 UI code generator 5.15.4
6  #
7  # WARNING: Any manual changes made to this file will be lost when pyuic5 is
8  # run again.  Do not edit this file unless you know what you are doing.
9
10
11 from PyQt5 import QtCore, QtGui, QtWidgets
12
13
14 class Ui_AmendOffenseDialog(object):
15     def setupUi(self, AmendOffenseDialog):
16         AmendOffenseDialog.setObjectName("AmendOffenseDialog")
17         AmendOffenseDialog.resize(1000, 300)
18         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred,
19         QtWidgets.QSizePolicy.Preferred)
20         sizePolicy.setHorizontalStretch(0)
21         sizePolicy.setVerticalStretch(0)
22         sizePolicy.setHeightForWidth(AmendOffenseDialog.sizePolicy().hasHeightForWidth())
23         AmendOffenseDialog.setSizePolicy(sizePolicy)
24         AmendOffenseDialog.setMinimumSize(QtCore.QSize(1000, 300))
25         font = QtGui.QFont()
26         font.setFamily("Palatino Linotype")
27         font.setPointSize(11)
28         font.setBold(False)
29         font.setItalic(False)
30         font.setWeight(9)
31         AmendOffenseDialog.setFont(font)
32         AmendOffenseDialog.setStyleSheet("background-color: rgb(25, 49, 91);\n"
33         "font: 75 11pt \"Palatino Linotype\";")
34         self.case_information_frame = QtWidgets.QFrame(AmendOffenseDialog)
35         self.case_information_frame.setGeometry(QtCore.QRect(9, 9, 981, 131))
36         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred,
37         QtWidgets.QSizePolicy.Preferred)
38         sizePolicy.setHorizontalStretch(0)
39         sizePolicy.setVerticalStretch(0)
40
41         sizePolicy.setHeightForWidth(self.case_information_frame.sizePolicy().hasHeightForWidth())
42         self.case_information_frame.setSizePolicy(sizePolicy)
43         self.case_information_frame.setMaximumSize(QtCore.QSize(1000, 800))
44         self.case_information_frame.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
45         self.case_information_frame.setStyleSheet("background-color: rgb(240, 240, 240);\n"
46         "font-weight: bold;")

```

```

47     self.case_information_frame.setFrameShape(QtWidgets.QFrame.Box)
48     self.case_information_frame.setFrameShadow(QtWidgets.QFrame.Raised)
49     self.case_information_frame.setLineWidth(2)
50     self.case_information_frame.setObjectName("case_information_frame")
51     self.defendant_name_label = QtWidgets.QLabel(self.case_information_frame)
52     self.defendant_name_label.setGeometry(QtCore.QRect(18, 53, 831, 27))
53     font = QtGui.QFont()
54     font.setFamily("Palatino Linotype")
55     font.setPointSize(11)
56     font.setBold(True)
57     font.setItalic(False)
58     font.setWeight(75)
59     self.defendant_name_label.setFont(font)
60     self.defendant_name_label.setObjectName("defendant_name_label")
61     self.case_number_label = QtWidgets.QLabel(self.case_information_frame)
62     self.case_number_label.setGeometry(QtCore.QRect(18, 13, 831, 27))
63     font = QtGui.QFont()
64     font.setFamily("Palatino Linotype")
65     font.setPointSize(11)
66     font.setBold(True)
67     font.setItalic(False)
68     font.setWeight(75)
69     self.case_number_label.setFont(font)
70     self.case_number_label.setObjectName("case_number_label")
71     self.defendant_attorney_name_label = QtWidgets.QLabel(self.case_information_frame)
72     self.defendant_attorney_name_label.setGeometry(QtCore.QRect(18, 93, 821, 27))
73     font = QtGui.QFont()
74     font.setFamily("Palatino Linotype")
75     font.setPointSize(11)
76     font.setBold(True)
77     font.setItalic(False)
78     font.setWeight(75)
79     self.defendant_attorney_name_label.setFont(font)
80     self.defendant_attorney_name_label.setObjectName("defendant_attorney_name_label")
81     self.cancel_Button = QtWidgets.QPushButton(self.case_information_frame)
82     self.cancel_Button.setGeometry(QtCore.QRect(832, 10, 141, 33))
83     self.cancel_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
84     self.cancel_Button.setObjectName("cancel_Button")
85     self.frame_2 = QtWidgets.QFrame(AmendOffenseDialog)
86     self.frame_2.setGeometry(QtCore.QRect(10, 150, 981, 141))
87     self.frame_2.setStyleSheet("background-color: rgb(240, 240, 240);\n"
88 "font: 75 11pt \"Palatino Linotype\";\n"
89 "font-weight: bold;")
90     self.frame_2.setFrameShape(QtWidgets.QFrame.Box)
91     self.frame_2.setFrameShadow(QtWidgets.QFrame.Raised)
92     self.frame_2.setLineWidth(2)

```

```
93     self.frame_2.setMidLineWidth(0)
94     self.frame_2.setObjectName("frame_2")
95     self.layoutWidget = QtWidgets.QWidget(self.frame_2)
96     self.layoutWidget.setGeometry(QtCore.QRect(830, 14, 141, 121))
97     self.layoutWidget.setObjectName("layoutWidget")
98     self.gridLayout = QtWidgets.QGridLayout(self.layoutWidget)
99     self.gridLayout.setContentsMargins(0, 0, 0, 0)
100    self.gridLayout.setObjectName("gridLayout")
101    self.clear_fields_Button = QtWidgets.QPushButton(self.layoutWidget)
102    self.clear_fields_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
103    self.clear_fields_Button.setObjectName("clear_fields_Button")
104    self.gridLayout.addWidget(self.clear_fields_Button, 0, 0, 1, 1)
105    self.continue_Button = QtWidgets.QPushButton(self.layoutWidget)
106    self.continue_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
107    self.continue_Button.setObjectName("continue_Button")
108    self.gridLayout.addWidget(self.continue_Button, 2, 0, 1, 1)
109    spacerItem = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum,
110    QtWidgets.QSizePolicy.Expanding)
111    self.gridLayout.addItem(spacerItem, 1, 0, 1, 1)
112    self.widget = QtWidgets.QWidget(self.frame_2)
113    self.widget.setGeometry(QtCore.QRect(13, 10, 791, 121))
114    self.widget.setObjectName("widget")
115    self.gridLayout_2 = QtWidgets.QGridLayout(self.widget)
116    self.gridLayout_2.setContentsMargins(0, 0, 0, 0)
117    self.gridLayout_2.setObjectName("gridLayout_2")
118    self.label = QtWidgets.QLabel(self.widget)
119    self.label.setObjectName("label")
120    self.gridLayout_2.addWidget(self.label, 0, 0, 1, 1)
121    self.motion_decision_box = QtWidgets.QComboBox(self.widget)
122    self.motion_decision_box.setStyleSheet("background-color: rgb(255, 255, 255);")
123    self.motion_decision_box.setEditable(False)
124    self.motion_decision_box.setObjectName("motion_decision_box")
125    self.motion_decision_box.addItem("")
126    self.motion_decision_box.addItem("")
127    self.gridLayout_2.addWidget(self.motion_decision_box, 0, 1, 1, 1)
128    self.label_2 = QtWidgets.QLabel(self.widget)
129    self.label_2.setObjectName("label_2")
130    self.gridLayout_2.addWidget(self.label_2, 1, 0, 1, 1)
131    self.original_charge_box = QtWidgets.QComboBox(self.widget)
132    self.original_charge_box.setStyleSheet("background-color: rgb(255, 255, 255);")
133    self.original_charge_box.setEditable(True)
134    self.original_charge_box.setObjectName("original_charge_box")
135    self.gridLayout_2.addWidget(self.original_charge_box, 1, 1, 1, 1)
136    self.label_3 = QtWidgets.QLabel(self.widget)
137    self.label_3.setObjectName("label_3")
138    self.gridLayout_2.addWidget(self.label_3, 2, 0, 1, 1)
```

```

139 self.amended_charge_box = QtWidgets.QComboBox(self.widget)
140 self.amended_charge_box.setStyleSheet("background-color: rgb(255, 255, 255);")
141 self.amended_charge_box.setEditable(True)
142 self.amended_charge_box.setObjectName("amended_charge_box")
143 self.gridLayout_2.addWidget(self.amended_charge_box, 2, 1, 1, 1)
144
145 self.retranslateUi(AmendOffenseDialog)
146 self.continue_Button.released.connect(AmendOffenseDialog.close_window)
147 self.cancel_Button.clicked.connect(AmendOffenseDialog.close_window)
148 self.clear_fields_Button.clicked.connect(self.original_charge_box.clearEditText)
149 self.clear_fields_Button.clicked.connect(self.amended_charge_box.clearEditText)
150 self.continue_Button.pressed.connect(AmendOffenseDialog.amend_offense)
151 QtCore.QMetaObject.connectSlotsByName(AmendOffenseDialog)
152 AmendOffenseDialog.setTabOrder(self.motion_decision_box, self.original_charge_box)
153 AmendOffenseDialog.setTabOrder(self.original_charge_box, self.amended_charge_box)
154 AmendOffenseDialog.setTabOrder(self.amended_charge_box, self.continue_Button)
155 AmendOffenseDialog.setTabOrder(self.continue_Button, self.clear_fields_Button)
156 AmendOffenseDialog.setTabOrder(self.clear_fields_Button, self.cancel_Button)
157
158 def retranslateUi(self, AmendOffenseDialog):
159     _translate = QtCore.QCoreApplication.translate
160     AmendOffenseDialog.setWindowTitle(_translate("AmendOffenseDialog", "Amend
161 Charge"))
162     self.defendant_name_label.setText(_translate("AmendOffenseDialog", "TextLabel"))
163     self.case_number_label.setText(_translate("AmendOffenseDialog", "TextLabel"))
164     self.defendant_attorney_name_label.setText(_translate("AmendOffenseDialog",
165 "TextLabel"))
166     self.cancel_Button.setText(_translate("AmendOffenseDialog", "Cancel"))
167     self.clear_fields_Button.setText(_translate("AmendOffenseDialog", "Clear Fields"))
168     self.continue_Button.setText(_translate("AmendOffenseDialog", "Continue"))
169     self.label.setText(_translate("AmendOffenseDialog", "Motion to amend is:"))
170     self.motion_decision_box.setItemText(0, _translate("AmendOffenseDialog", "Granted"))
171     self.motion_decision_box.setItemText(1, _translate("AmendOffenseDialog", "Denied"))
172     self.label_2.setText(_translate("AmendOffenseDialog", "Original charge:"))
173     self.label_3.setText(_translate("AmendOffenseDialog", "Amended charge:"))

```



```

1  """Module containing all data structures for everything at the moment."""
2  from models.party_types import JudicialOfficer, Defendant
3
4
5  class CaseInformation(object):
6      """This object stores all the information for a case both at inception and
7      as it is populated through the application."""
8
9      def __init__(self, judicial_officer):
10         self.judicial_officer = judicial_officer
11         self.case_number = None
12         self.defendant = Defendant()
13         self.fra_in_file = None
14         self.fra_in_court = None
15         self.plea_trial_date = None
16         self.charges_list = []
17         self.community_control_terms = None
18         self.ability_to_pay_time = None
19         self.balance_due_date = None
20         self.amend_offense_details = None
21         self.community_service_terms = None
22         self.hours_of_service = None
23         self.days_to_complete_service = None
24         self.due_date_for_service = None
25         self.license_suspension_details = None
26         self.other_conditions_details = None
27         self.court_costs_ordered = None
28         self.court_costs = 0
29         self.total_fines = 0
30
31     def add_charge_to_list(self, charge):
32         self.charges_list.append(charge)
33
34     def get_case_information(self):
35         """Returns a dictionary with all of case information required
36         to populate an entry."""
37         self.formatted_case_information = {
38             "case_number": self.case_number,
39             "defendant_last_name": self.defendant.last_name,
40             "defendant_first_name": self.defendant.first_name,
41             "plea_trial_date": self.plea_trial_date,
42             "amend_offense_details": self.amend_offense_details,
43             "charges_list": self.charges_list,
44             "ability_to_pay_time": self.ability_to_pay_time,
45             "balance_due_date": self.balance_due_date,
46             "community_control_terms": self.community_control_terms,

```

```

47         "fra_in_file": self.fra_in_file,
48         "fra_in_court": self.fra_in_court,
49         "community_service_terms": self.community_service_terms,
50         "hours_of_service": self.hours_of_service,
51         "days_to_complete_service": self.days_to_complete_service,
52         "due_date_for_service": self.due_date_for_service,
53         "judicial_officer": self.judicial_officer.first_name
54         + " "
55         + self.judicial_officer.last_name,
56         "judicial_officer_type": self.judicial_officer.officer_type,
57         "license_suspension_details": self.license_suspension_details,
58         "other_conditions_details": self.other_conditions_details,
59         "court_costs_ordered": self.court_costs_ordered,
60     }
61     return self.formatted_case_information
62
63
64 class CriminalCharge(object):
65     def __init__(self):
66         self.offense = None
67         self.statute = None
68         self.degree = None
69         self.plea = None
70         self.type = None
71         self.finding = None
72         self.fines_amount = None
73         self.fines_suspended = None
74         self.court_costs = None
75         self.jail_days = None
76         self.jail_days_suspended = None
77
78
79 class AmendOffenseDetails(object):
80     """TODO: This should be refactored to a pure function most likely."""
81     def __init__(self):
82         self.original_charge = None
83         self.amended_charge = None
84         self.motion_disposition = "granted"
85
86
87 class CommunityControlTerms(object):
88     def __init__(self):
89         self.community_control_required = False
90         self.term_of_community_control = 0
91         self.type_of_community_control = "basic"
92         self.not_refuse = False

```

```
93         self.not_consume = False
94
95
96 class CommunityServiceTerms(object):
97     def __init__(self):
98         self.community_service_ordered = False
99         self.hours_of_service = 0
100         self.days_to_complete_service = 0
101         self.due_date_for_service = None
102
103
104 class LicenseSuspensionTerms(object):
105     def __init__(self):
106         self.license_suspension_ordered = False
107         self.license_type = None
108         self.license_suspended_date = None
109         self.license_suspension_term = None
110         self.remedial_driving_class_required = False
111
112
113 class OtherConditionsTerms(object):
114     def __init__(self):
115         self.other_conditions_ordered = False
116         self.other_conditions_terms = None
```

```

1  """Module containing all the data structures for cases."""
2
3  from PartyTypes import Party
4
5  class Case(object):
6      """The base object from which all other objects are connected. Any matter
7      before the court will have at a minimum some type of case."""
8
9      def __init__(self, case_number: str) -> None:
10         self.case_number = case_number
11         self.case_type_code = case_number[2:5]
12         self.case_parties = []
13
14         def add_party_to_case(self, party_type: str) -> None:
15             """The party_type will generally be either "Plaintiff" or "Defendant."""
16             self.party_type = party_type
17             self.case_parties.append(self.party_type)
18
19
20  class TrafficCase(Case):
21      """
22      A case type for traffic cases. The format for the case number will be
23      either YYTRCXXXX or YYTRDXXXX.
24
25      YY = last two digits of the year the case was filed.
26      TRC = is for OVI offense cases.
27      TRD = is for all other traffic cases (non-OVI).
28      XXXX = the case number assigned to the case.
29
30      """
31
32      def __init__(self, case_number):
33          """
34          The initialization of a TrafficCase takes the case number and strips
35          out the case numbers to create a case_type_code (i.e. TRC or TRD).
36          """
37          super().__init__(case_number)
38          self.case_charges: list = []
39
40          def add_charge_to_case(self, traffic_charge: TrafficCharge) -> None:
41              self.traffic_charge: TrafficCharge = traffic_charge
42              self.case_charges.append(self.traffic_charge)
43
44
45  class CriminalCase(Case):
46      """

```

```
47     A case type for criminal cases. The format for the case number will be
48     either YYCRBXXXX
49
50     YY = last two digits of the year the case was filed.
51     CRB = is for all criminal cases.
52     XXXX = the case number assigned to the case.
53
54     """
55
56     def __init__(self, case_number):
57         """
58         The initialization of a CriminalCase takes the case number and strips
59         out the case numbers to create a case_type_code CRB.
60         """
61         super().__init__(case_number)
62
63
64     test = CriminalCase("21CRB0123")
65     print(test.case_type_code)
66     test.add_party_to_case(Party("Plaintiff"))
67     print(test.case_parties)
68     print(test.case_parties[0].party_type)
```

```

1  """The module that contains all controller classes that are common to criminal
2  cases (criminal includes traffic). """
3  import os
4  import pathlib
5  from docxtpl import DocxTemplate
6  from loguru import logger
7
8  from PyQt5.QtWidgets import QDialog
9  from PyQt5 import QtCore
10 from PyQt5.QtCore import QDate
11 from PyQt5.QtWidgets import QLabel, QPushButton, QMessageBox, QComboBox, QLineEdit
12 from PyQt5.QtSql import QSqlDatabase, QSqlQuery
13
14 from models.case_information import (
15     CaseInformation,
16     CriminalCharge,
17     AmendOffenseDetails,
18     LicenseSuspensionTerms,
19     CommunityControlTerms,
20     CommunityServiceTerms,
21     OtherConditionsTerms,
22 )
23 from views.add_conditions_dialog_ui import Ui_AddConditionsDialog
24 from views.amend_offense_dialog_ui import Ui_AmendOffenseDialog
25 from resources.db.DatabaseCreation import create_offense_list, create_statute_list
26
27
28 PATH = str(pathlib.Path().absolute())
29 TEMPLATE_PATH = PATH + "\\resources\\Templates\\"
30 SAVE_PATH = PATH + "\\resources\\Saved\\"
31 DB_PATH = PATH + "\\resources\\db\\"
32 CHARGES_DATABASE = DB_PATH + "\\charges.sqlite"
33
34
35 class BaseCriminalDialog(QDialog):
36     """This class is a base class to provide methods that are used by some or
37     all class controllers that are used in the application. This class is never
38     instantiated as its own dialog, but the init contains the setup for all
39     inherited class controllers."""
40
41     def __init__(self, parent=None):
42         super().__init__(parent)
43         self.setupUi(self)
44         self.doc = None
45         self.docname = None
46         self.pay_date_dict = {

```



```

47         "forthwith": 0,
48         "within 30 days": 30,
49         "within 60 days": 60,
50         "within 90 days": 90,
51     }
52
53     def close_window(self):
54         """Function connected to a button to close the window. Can be connected
55         to any button press/click/release to close a window."""
56         self.close()
57
58     @logger.catch
59     def add_charge_process(self, bool):
60         """The order of functions that are called when the add_charge_Button is
61         clicked(). The order is important to make sure the information is
62         updated before the charge is added and the data cleared from the fields.
63
64         The bool is passed as an argument through clicked() but not used."""
65         self.criminal_charge = CriminalCharge()
66         self.add_charge()
67         self.clear_charge_fields()
68
69     @logger.catch
70     def create_entry_process(self):
71         """The order of functions that are called when the create_entry_Button is pressed()
72         on the MinorMisdemeanorDialog. The order is important to make sure the information is
73         updated before the entry is created."""
74         self.update_case_information()
75         self.create_entry()
76         self.close_event()
77
78     @logger.catch
79     def create_entry(self):
80         """The standard function used to create an entry when a create entry
81         button is press/click/released."""
82         self.doc = DocxTemplate(self.template.template_path)
83         self.doc.render(self.case_information.get_case_information())
84         self.set_document_name()
85         self.doc.save(SAVE_PATH + self.docname)
86         os.startfile(SAVE_PATH + self.docname)
87
88     def set_document_name(self):
89         """Sets document name based on the case number and name of the template
90         must include '.docx' to make it a Word document."""
91         self.docname = (
92             self.case_information.case_number + "_" + self.template.template_name + ".docx"

```

```

93         )
94
95     def set_case_information_banner(self):
96         """Sets the banner on a view of the interface. It modifies label
97         widgets on the view to text that was entered."""
98         self.defendant_name_label.setText(
99             "State of Ohio v. {defendant_first_name} {defendant_last_name}".format(
100                 defendant_first_name = self.case_information.defendant.first_name,
101                 defendant_last_name = self.case_information.defendant.last_name
102             )
103         )
104         self.case_number_label.setText(self.case_information.case_number)
105         if self.case_information.defendant_attorney_name is not None:
106             self.defendant_attorney_name_label.setText(
107                 "Attorney: " + self.case_information.defendant_attorney_name
108             )
109         else:
110             self.defendant_attorney_name_label.setText("Attorney: None")
111
112     def update_case_information(self):
113         """Placeholder for future use to refactor out of other dialogs and reduce
114         code reuse."""
115         pass
116
117
118 class AddConditionsDialog(BaseCriminalDialog, Ui_AddConditionsDialog):
119     """The AddConditionsDialog is created when the addConditionsButton is clicked on
120     the MinorMisdemeanorDialog. The conditions that are available to enter information
121     for are based on the checkboxes that are checked on the MMD screen."""
122
123     @logger.catch
124     def __init__(self, minor_misdemeanor_dialog, parent=None):
125         super().__init__(parent)
126         self.case_information = minor_misdemeanor_dialog.case_information
127         self.modify_view()
128         self.community_service = (
129             minor_misdemeanor_dialog.community_service_checkBox.isChecked()
130         )
131         self.license_suspension = (
132             minor_misdemeanor_dialog.license_suspension_checkBox.isChecked()
133         )
134         self.community_control = (
135             minor_misdemeanor_dialog.community_control_checkBox.isChecked()
136         )
137         self.other_conditions = (
138             minor_misdemeanor_dialog.other_conditions_checkBox.isChecked()

```

```

139         )
140         self.other_conditions = (
141             minor_misdemeanor_dialog.other_conditions_checkBox.isChecked()
142         )
143         self.enable_condition_frames()
144
145     @logger.catch
146     def modify_view(self):
147         """Modifies the view of AddConditionsDialog that is created by the UI
148         file.
149         Gets the total number of charges from the charges in charges_list then
150         loops through the charges_list and adds parts of each charge to the
151         view. CLEAN UP?"""
152         index_of_charge_to_add = 0
153         column = self.charges_gridLayout.columnCount() + 1
154         total_charges_to_add = len(self.case_information.charges_list)
155         while index_of_charge_to_add < total_charges_to_add:
156             charge = vars(self.case_information.charges_list[index_of_charge_to_add])
157             if charge is not None:
158                 self.charges_gridLayout.addWidget(
159                     QLabel(charge.get("offense")), 0, column
160                 )
161                 self.charges_gridLayout.addWidget(
162                     QLabel(charge.get("statute")), 1, column
163                 )
164                 self.charges_gridLayout.addWidget(
165                     QLabel(charge.get("finding")), 2, column
166                 )
167                 column += 1
168                 index_of_charge_to_add += 1
169
170     @logger.catch
171     def enable_condition_frames(self):
172         """Enables the frames on the AddConditionsDialog dialog if the condition is checked
173         on the MinorMisdemeanorDialog screen. Also creates an instance of the object for
174         each condition."""
175         if self.other_conditions is True:
176             self.other_conditions_frame.setEnabled(True)
177             self.other_conditions_details = OtherConditionsTerms()
178         if self.license_suspension is True:
179             self.license_suspension_frame.setEnabled(True)
180             self.license_suspension_details = LicenseSuspensionTerms()
181             self.license_suspension_date_box.setDate(QtCore.QDate.currentDate())
182         if self.community_service is True:
183             self.community_service_frame.setEnabled(True)
184             self.community_service_terms = CommunityServiceTerms()

```

```

185         self.community_service_date_to_complete_box.setDate(
186             QtCore.QDate.currentDate()
187         )
188     if self.community_control is True:
189         self.community_control_frame.setEnabled(True)
190         self.community_control_terms = CommunityControlTerms()
191
192     @logger.catch
193     def add_conditions(self):
194         """The method is connected to the pressed() signal of continue_Button on the
195         Add Conditions screen."""
196         if self.community_service is True:
197             self.add_community_service_terms()
198         if self.community_control is True:
199             self.add_community_control_terms()
200         if self.license_suspension is True:
201             self.add_license_suspension_details()
202         if self.other_conditions is True:
203             self.add_other_condition_details()
204
205     @logger.catch
206     def add_community_control_terms(self):
207         """The method adds the data entered to the CommunityControlTerms object
208         that is created when the dialog is initialized. Then the data is transferred
209         to case_information."""
210         self.community_control_terms.type_of_community_control = (
211             self.type_of_community_control_box.currentText()
212         )
213         self.community_control_terms.term_of_community_control = (
214             self.term_of_community_control_box.currentText()
215         )
216         self.case_information.community_control_terms = self.community_control_terms
217
218     @logger.catch
219     def add_community_service_terms(self):
220         """The method adds the data entered to the CommunityServiceTerms object
221         that is created when the dialog is initialized. Then the data is transferred
222         to case_information."""
223         self.community_service_terms.hours_of_service = (
224             self.community_service_hours_ordered_box.value()
225         )
226         self.community_service_terms.days_to_complete_service = (
227             self.community_service_days_to_complete_box.currentText()
228         )
229         self.community_service_terms.due_date_for_service = (
230             self.community_service_date_to_complete_box.date().toString("MMMM dd, yyyy")

```

```

231         )
232         self.case_information.community_service_terms = self.community_service_terms
233
234     @logger.catch
235     def add_license_suspension_details(self):
236         """The method adds the data entered to the LicenseSuspensionTerms object
237         that is created when the dialog is initialized. Then the data is transferred
238         to case_information."""
239         self.license_suspension_details.license_type = (
240             self.license_type_box.currentText()
241         )
242         self.license_suspension_details.license_suspended_date = (
243             self.license_suspension_date_box.date().toString("MMMM dd, yyyy")
244         )
245         self.license_suspension_details.license_suspension_term = (
246             self.term_of_suspension_box.currentText()
247         )
248         if self.remedial_driving_class_checkBox.isChecked():
249             self.license_suspension_details.remedial_driving_class_required = True
250         else:
251             self.license_suspension_details.remedial_driving_class_required = False
252         self.case_information.license_suspension_details = (
253             self.license_suspension_details
254         )
255
256     @logger.catch
257     def add_other_condition_details(self):
258         """The method allows for adding other conditions based on free form text
259         entry."""
260         self.other_conditions_details.other_conditions_terms = (
261             self.other_conditions_plainTextEdit.toPlainText()
262         )
263         self.case_information.other_conditions_details = (
264             self.other_conditions_details
265         )
266
267     @logger.catch
268     def set_service_date(self, days_to_complete):
269         """Sets the community_service_date_to_complete_box based on the number
270         of days chosen in the community_service_date_to_complete_box."""
271         days_to_complete = int(
272             self.community_service_days_to_complete_box.currentText()
273         )
274         self.community_service_date_to_complete_box.setDate(
275             QDate.currentDate().addDays(days_to_complete)
276

```

```

277         )
278
279
280 class AmendOffenseDialog(BaseCriminalDialog, Ui_AmendOffenseDialog):
281     """The AddOffenseDialog is created when the amendOffenseButton is clicked on
282     the MinorMisdemeanorDialog screen. The case information is passed in
283     order to populate the case information banner.
284
285     The set_case_information_banner is an inherited method from BaseCriminalDialog."""
286     @logger.catch
287     def __init__(self, case_information, parent=None):
288         super().__init__(parent)
289         self.case_information = case_information
290         self.amend_offense_details = AmendOffenseDetails()
291         self.set_case_information_banner()
292         self.modify_view()
293
294     @logger.catch
295     def modify_view(self):
296         """The modify view method updates the view that is created on init.
297         Place items in this method that can't be added directly in QtDesigner
298         so that they don't need to be changed in the view file each time pyuic5
299         is run."""
300         offense_list = create_offense_list()
301         self.original_charge_box.addItem(offense_list)
302         self.amended_charge_box.addItem(offense_list)
303
304     @logger.catch
305     def amend_offense(self):
306         """Adds the data entered for the amended offense to the AmendOffenseDetails
307         object then points the case_information object to the AmendOffenseDetails
308         object."""
309         self.amend_offense_details.original_charge = (
310             self.original_charge_box.currentText()
311         )
312         self.amend_offense_details.amended_charge = (
313             self.amended_charge_box.currentText()
314         )
315         self.amend_offense_details.motion_disposition = (
316             self.motion_decision_box.currentText()
317         )
318         self.case_information.amend_offense_details = self.amend_offense_details

```



```

1  from PyQt5 import QtCore
2  from PyQt5.QtCore import QDate
3  from PyQt5.QtWidgets import QLabel, QPushButton, QMessageBox, QComboBox, QLineEdit
4
5  class PleaComboBox(QComboBox):
6      def __init__(self, parent=None):
7          super(QComboBox, self).__init__(parent)
8          self.set_up_widget()
9
10     def set_up_widget(self):
11         self.setMinimumSize(QtCore.QSize(200, 0))
12         self.setMaximumSize(QtCore.QSize(200, 50))
13         self.setStyleSheet("background-color: rgb(255, 255, 255);")
14         self.setEditable(True)
15         self.setFocusPolicy(QtCore.Qt.NoFocus)
16         self.setObjectName("plea_choice_box")
17         self.addItem("")
18         self.addItem("Guilty")
19         self.addItem("No Contest")
20         self.addItem("Not Guilty")
21         self.addItem("Dismissed")
22
23
24     class FindingComboBox(QComboBox):
25         def __init__(self, parent=None):
26             super(QComboBox, self).__init__(parent)
27             self.set_up_widget()
28
29         def set_up_widget(self):
30             self.setMinimumSize(QtCore.QSize(200, 0))
31             self.setMaximumSize(QtCore.QSize(200, 50))
32             self.setStyleSheet("background-color: rgb(255, 255, 255);")
33             self.setEditable(True)
34             self.setFocusPolicy(QtCore.Qt.NoFocus)
35             self.setObjectName("finding_choice_box")
36             self.addItem("")
37             self.addItem("Guilty")
38             self.addItem("Not Guilty")
39             self.addItem("Dismissed")
40             self.addItem("Guilty - Allied Offense")
41             self.addItem("Not Guilty - Allied Offense")
42
43
44     class FineLineEdit(QLineEdit):
45         def __init__(self, offense, parent=None):
46             super(QLineEdit, self).__init__(parent)

```

```

47         self.set_up_widget(offense)
48
49     def set_up_widget(self, offense):
50         self.setMinimumSize(QtCore.QSize(200, 0))
51         self.setMaximumSize(QtCore.QSize(200, 50))
52         self.setStyleSheet("background-color: rgb(255, 255, 255);")
53         self.setObjectName("fines_amount")
54         if offense == "Seatbelt - Driver":
55             self.setText("30")
56         elif offense == "Seatbelt - Passenger":
57             self.setText("20")
58         elif offense == "Failure to Stop for School Bus":
59             self.setText("500")
60         else:
61             self.setText("")
62
63
64     class FineSuspendedLineEdit(QLineEdit):
65         def __init__(self, parent=None):
66             super(QLineEdit, self).__init__(parent)
67             self.set_up_widget()
68
69         def set_up_widget(self):
70             self.setMinimumSize(QtCore.QSize(200, 0))
71             self.setMaximumSize(QtCore.QSize(200, 50))
72             self.setStyleSheet("background-color: rgb(255, 255, 255);")
73             self.setObjectName("fines_suspended")
74
75
76     class DeleteButton(QPushButton):
77         def __init__(self, parent=None):
78             super(QPushButton, self).__init__(parent)
79             self.set_up_widget()
80
81         def set_up_widget(self):
82             self.setStyleSheet("background-color: rgb(160, 160, 160);")
83             self.setText("Delete")
84             self.setObjectName("delete_Button")
85             self.setFocusPolicy(QtCore.Qt.NoFocus)
86
87
88     class AmendButton(QPushButton):
89         def __init__(self, parent=None):
90             super(QPushButton, self).__init__(parent)
91             self.set_up_widget()
92

```

```
93     def set_up_widget(self):
94         self.setStyleSheet("background-color: rgb(160, 160, 160);")
95         self.setText("Amend")
96         self.setObjectName("amend_Button")
97         self.setFocusPolicy(QtCore.Qt.NoFocus)
```

```

1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file 'resources/ui/MainWindow.ui'
4  #
5  # Created by: PyQt5 UI code generator 5.15.4
6  #
7  # WARNING: Any manual changes made to this file will be lost when pyuic5 is
8  # run again.  Do not edit this file unless you know what you are doing.
9
10
11 from PyQt5 import QtCore, QtGui, QtWidgets
12
13
14 class Ui_MainWindow(object):
15     def setupUi(self, MainWindow):
16         MainWindow.setObjectName("MainWindow")
17         MainWindow.resize(1000, 800)
18         MainWindow.setMinimumSize(QtCore.QSize(1000, 800))
19         font = QtGui.QFont()
20         font.setFamily("Palatino Linotype")
21         font.setPointSize(11)
22         font.setBold(True)
23         font.setItalic(False)
24         font.setWeight(75)
25         MainWindow.setFont(font)
26         MainWindow.setStyleSheet("font: 75 11pt \"Palatino Linotype\";\n"
27 "color: rgb(255, 0, 0);\n"
28 "selection-color: rgb(255, 0, 0);\n"
29 "font-weight: bold;")
30         self.centralwidget = QtWidgets.QWidget(MainWindow)
31         self.centralwidget.setObjectName("centralwidget")
32         self.layoutWidget = QtWidgets.QWidget(self.centralwidget)
33         self.layoutWidget.setGeometry(QtCore.QRect(20, 290, 641, 68))
34         self.layoutWidget.setObjectName("layoutWidget")
35         self.gridLayout_3 = QtWidgets.QGridLayout(self.layoutWidget)
36         self.gridLayout_3.setContentsMargins(0, 0, 0, 0)
37         self.gridLayout_3.setObjectName("gridLayout_3")
38         self.FinalJudgmentEntryButton_6 = QtWidgets.QPushButton(self.layoutWidget)
39         self.FinalJudgmentEntryButton_6.setMinimumSize(QtCore.QSize(120, 30))
40         self.FinalJudgmentEntryButton_6.setObjectName("FinalJudgmentEntryButton_6")
41         self.gridLayout_3.addWidget(self.FinalJudgmentEntryButton_6, 0, 0, 1, 1)
42         self.FinalJudgmentEntryButton_7 = QtWidgets.QPushButton(self.layoutWidget)
43         self.FinalJudgmentEntryButton_7.setMinimumSize(QtCore.QSize(120, 30))
44         self.FinalJudgmentEntryButton_7.setObjectName("FinalJudgmentEntryButton_7")
45         self.gridLayout_3.addWidget(self.FinalJudgmentEntryButton_7, 0, 1, 1, 1)
46         self.MinorMisdemeanorTrafficButton_2 = QtWidgets.QPushButton(self.layoutWidget)

```

```
47         self.MinorMisdemeanorTrafficButton_2.setMinimumSize(QtCore.QSize(120, 30))
48
49     self.MinorMisdemeanorTrafficButton_2.setObjectName("MinorMisdemeanorTrafficButton_2")
50     self.gridLayout_3.addWidget(self.MinorMisdemeanorTrafficButton_2, 0, 2, 1, 1)
51     self.FinalJudgmentEntryButton_8 = QtWidgets.QPushButton(self.layoutWidget)
52     self.FinalJudgmentEntryButton_8.setMinimumSize(QtCore.QSize(120, 30))
53     self.FinalJudgmentEntryButton_8.setObjectName("FinalJudgmentEntryButton_8")
54     self.gridLayout_3.addWidget(self.FinalJudgmentEntryButton_8, 1, 0, 1, 1)
55     self.FinalJudgmentEntryButton_9 = QtWidgets.QPushButton(self.layoutWidget)
56     self.FinalJudgmentEntryButton_9.setMinimumSize(QtCore.QSize(120, 30))
57     self.FinalJudgmentEntryButton_9.setObjectName("FinalJudgmentEntryButton_9")
58     self.gridLayout_3.addWidget(self.FinalJudgmentEntryButton_9, 1, 1, 1, 1)
59     self.FinalJudgmentEntryButton_10 = QtWidgets.QPushButton(self.layoutWidget)
60     self.FinalJudgmentEntryButton_10.setMinimumSize(QtCore.QSize(120, 30))
61     self.FinalJudgmentEntryButton_10.setObjectName("FinalJudgmentEntryButton_10")
62     self.gridLayout_3.addWidget(self.FinalJudgmentEntryButton_10, 1, 2, 1, 1)
63     self.label_2 = QtWidgets.QLabel(self.centralwidget)
64     self.label_2.setGeometry(QtCore.QRect(20, 100, 151, 31))
65     font = QtGui.QFont()
66     font.setFamily("Palatino Linotype")
67     font.setPointSize(11)
68     font.setBold(True)
69     font.setItalic(False)
70     font.setUnderline(True)
71     font.setWeight(75)
72     self.label_2.setFont(font)
73     self.label_2.setObjectName("label_2")
74     self.label_3 = QtWidgets.QLabel(self.centralwidget)
75     self.label_3.setGeometry(QtCore.QRect(20, 250, 181, 31))
76     font = QtGui.QFont()
77     font.setFamily("Palatino Linotype")
78     font.setPointSize(11)
79     font.setBold(True)
80     font.setItalic(False)
81     font.setUnderline(True)
82     font.setWeight(75)
83     self.label_3.setFont(font)
84     self.label_3.setObjectName("label_3")
85     self.label_4 = QtWidgets.QLabel(self.centralwidget)
86     self.label_4.setGeometry(QtCore.QRect(20, 400, 171, 31))
87     font = QtGui.QFont()
88     font.setFamily("Palatino Linotype")
89     font.setPointSize(11)
90     font.setBold(True)
91     font.setItalic(False)
92     font.setUnderline(True)
```

```

93     font.setWeight(75)
94     self.label_4.setFont(font)
95     self.label_4.setObjectName("label_4")
96     self.layoutWidget_2 = QtWidgets.QWidget(self.centralwidget)
97     self.layoutWidget_2.setGeometry(QtCore.QRect(20, 440, 651, 68))
98     self.layoutWidget_2.setObjectName("layoutWidget_2")
99     self.gridLayout_4 = QtWidgets.QGridLayout(self.layoutWidget_2)
100    self.gridLayout_4.setContentsMargins(0, 0, 0, 0)
101    self.gridLayout_4.setObjectName("gridLayout_4")
102    self.FinalJudgmentEntryButton_11 = QtWidgets.QPushButton(self.layoutWidget_2)
103    self.FinalJudgmentEntryButton_11.setMinimumSize(QtCore.QSize(120, 30))
104    self.FinalJudgmentEntryButton_11.setObjectName("FinalJudgmentEntryButton_11")
105    self.gridLayout_4.addWidget(self.FinalJudgmentEntryButton_11, 0, 0, 1, 1)
106    self.FinalJudgmentEntryButton_12 = QtWidgets.QPushButton(self.layoutWidget_2)
107    self.FinalJudgmentEntryButton_12.setMinimumSize(QtCore.QSize(120, 30))
108    self.FinalJudgmentEntryButton_12.setObjectName("FinalJudgmentEntryButton_12")
109    self.gridLayout_4.addWidget(self.FinalJudgmentEntryButton_12, 0, 1, 1, 1)
110    self.MinorMisdemeanorTrafficButton_3 =
111    QtWidgets.QPushButton(self.layoutWidget_2)
112    self.MinorMisdemeanorTrafficButton_3.setMinimumSize(QtCore.QSize(120, 30))
113
114    self.MinorMisdemeanorTrafficButton_3.setObjectName("MinorMisdemeanorTrafficButton_3")
115    self.gridLayout_4.addWidget(self.MinorMisdemeanorTrafficButton_3, 0, 2, 1, 1)
116    self.FinalJudgmentEntryButton_13 = QtWidgets.QPushButton(self.layoutWidget_2)
117    self.FinalJudgmentEntryButton_13.setMinimumSize(QtCore.QSize(120, 30))
118    self.FinalJudgmentEntryButton_13.setObjectName("FinalJudgmentEntryButton_13")
119    self.gridLayout_4.addWidget(self.FinalJudgmentEntryButton_13, 1, 0, 1, 1)
120    self.FinalJudgmentEntryButton_14 = QtWidgets.QPushButton(self.layoutWidget_2)
121    self.FinalJudgmentEntryButton_14.setMinimumSize(QtCore.QSize(120, 30))
122    self.FinalJudgmentEntryButton_14.setObjectName("FinalJudgmentEntryButton_14")
123    self.gridLayout_4.addWidget(self.FinalJudgmentEntryButton_14, 1, 1, 1, 1)
124    self.FinalJudgmentEntryButton_15 = QtWidgets.QPushButton(self.layoutWidget_2)
125    self.FinalJudgmentEntryButton_15.setMinimumSize(QtCore.QSize(120, 30))
126    self.FinalJudgmentEntryButton_15.setObjectName("FinalJudgmentEntryButton_15")
127    self.gridLayout_4.addWidget(self.FinalJudgmentEntryButton_15, 1, 2, 1, 1)
128    self.layoutWidget1 = QtWidgets.QWidget(self.centralwidget)
129    self.layoutWidget1.setGeometry(QtCore.QRect(20, 140, 961, 68))
130    self.layoutWidget1.setObjectName("layoutWidget1")
131    self.gridLayout_2 = QtWidgets.QGridLayout(self.layoutWidget1)
132    self.gridLayout_2.setContentsMargins(0, 0, 0, 0)
133    self.gridLayout_2.setObjectName("gridLayout_2")
134    self.GreenSheetButton = QtWidgets.QPushButton(self.layoutWidget1)
135    self.GreenSheetButton.setMinimumSize(QtCore.QSize(120, 30))
136    self.GreenSheetButton.setObjectName("GreenSheetButton")
137    self.gridLayout_2.addWidget(self.GreenSheetButton, 0, 1, 1, 1)
138    self.FinalJudgmentEntryButton_5 = QtWidgets.QPushButton(self.layoutWidget1)

```



```
139 self.FinalJudgmentEntryButton_5.setMinimumSize(QtCore.QSize(120, 30))
140 self.FinalJudgmentEntryButton_5.setObjectName("FinalJudgmentEntryButton_5")
141 self.gridLayout_2.addWidget(self.FinalJudgmentEntryButton_5, 0, 2, 1, 1)
142 self.MinorMisdemeanorTrafficButton = QtWidgets.QPushButton(self.layoutWidget1)
143 self.MinorMisdemeanorTrafficButton.setMinimumSize(QtCore.QSize(120, 30))
144
145 self.MinorMisdemeanorTrafficButton.setObjectName("MinorMisdemeanorTrafficButton")
146 self.gridLayout_2.addWidget(self.MinorMisdemeanorTrafficButton, 0, 0, 1, 1)
147 self.FinalJudgmentEntryButton_2 = QtWidgets.QPushButton(self.layoutWidget1)
148 self.FinalJudgmentEntryButton_2.setMinimumSize(QtCore.QSize(120, 30))
149 self.FinalJudgmentEntryButton_2.setObjectName("FinalJudgmentEntryButton_2")
150 self.gridLayout_2.addWidget(self.FinalJudgmentEntryButton_2, 1, 0, 1, 1)
151 self.FinalJudgmentEntryButton_3 = QtWidgets.QPushButton(self.layoutWidget1)
152 self.FinalJudgmentEntryButton_3.setMinimumSize(QtCore.QSize(120, 30))
153 self.FinalJudgmentEntryButton_3.setObjectName("FinalJudgmentEntryButton_3")
154 self.gridLayout_2.addWidget(self.FinalJudgmentEntryButton_3, 1, 1, 1, 1)
155 self.FinalJudgmentEntryButton_4 = QtWidgets.QPushButton(self.layoutWidget1)
156 self.FinalJudgmentEntryButton_4.setMinimumSize(QtCore.QSize(120, 30))
157 self.FinalJudgmentEntryButton_4.setObjectName("FinalJudgmentEntryButton_4")
158 self.gridLayout_2.addWidget(self.FinalJudgmentEntryButton_4, 1, 2, 1, 1)
159 self.layoutWidget2 = QtWidgets.QWidget(self.centralwidget)
160 self.layoutWidget2.setGeometry(QtCore.QRect(22, 20, 817, 31))
161 self.layoutWidget2.setObjectName("layoutWidget2")
162 self.gridLayout = QtWidgets.QGridLayout(self.layoutWidget2)
163 self.gridLayout.setContentsMargins(0, 0, 0, 0)
164 self.gridLayout.setObjectName("gridLayout")
165 self.label = QtWidgets.QLabel(self.layoutWidget2)
166 font = QtGui.QFont()
167 font.setFamily("Palatino Linotype")
168 font.setPointSize(11)
169 font.setBold(True)
170 font.setItalic(False)
171 font.setWeight(75)
172 self.label.setFont(font)
173 self.label.setObjectName("label")
174 self.gridLayout.addWidget(self.label, 0, 0, 1, 1)
175 self.hemmeter_radioButton = QtWidgets.QRadioButton(self.layoutWidget2)
176 self.hemmeter_radioButton.setObjectName("hemmeter_radioButton")
177 self.buttonGroup = QtWidgets.QButtonGroup(MainWindow)
178 self.buttonGroup.setObjectName("buttonGroup")
179 self.buttonGroup.addButton(self.hemmeter_radioButton)
180 self.gridLayout.addWidget(self.hemmeter_radioButton, 0, 1, 1, 1)
181 self.rohrer_radioButton = QtWidgets.QRadioButton(self.layoutWidget2)
182 self.rohrer_radioButton.setObjectName("rohrer_radioButton")
183 self.buttonGroup.addButton(self.rohrer_radioButton)
184 self.gridLayout.addWidget(self.rohrer_radioButton, 0, 2, 1, 1)
```

```
185 self.pelanda_radioButton = QtWidgets.QRadioButton(self.layoutWidget2)
186 self.pelanda_radioButton.setObjectName("pelanda_radioButton")
187 self.buttonGroup.addButton(self.pelanda_radioButton)
188 self.gridLayout.addWidget(self.pelanda_radioButton, 0, 3, 1, 1)
189 self.bunner_radioButton = QtWidgets.QRadioButton(self.layoutWidget2)
190 self.bunner_radioButton.setObjectName("bunner_radioButton")
191 self.buttonGroup.addButton(self.bunner_radioButton)
192 self.gridLayout.addWidget(self.bunner_radioButton, 0, 4, 1, 1)
193 MainWindow.setCentralWidget(self.centralwidget)
194 self.menubar = QtWidgets.QMenuBar(MainWindow)
195 self.menubar.setGeometry(QtCore.QRect(0, 0, 1000, 31))
196 self.menubar.setObjectName("menubar")
197 self.menuFile = QtWidgets.QMenu(self.menubar)
198 self.menuFile.setObjectName("menuFile")
199 self.menuSettings = QtWidgets.QMenu(self.menubar)
200 self.menuSettings.setObjectName("menuSettings")
201 self.menuView = QtWidgets.QMenu(self.menubar)
202 self.menuView.setObjectName("menuView")
203 MainWindow.setMenuBar(self.menubar)
204 self.statusbar = QtWidgets.QStatusBar(MainWindow)
205 self.statusbar.setObjectName("statusbar")
206 MainWindow.setStatusBar(self.statusbar)
207 self.actionOpen = QtWidgets.QAction(MainWindow)
208 self.actionOpen.setObjectName("actionOpen")
209 self.actionOptions = QtWidgets.QAction(MainWindow)
210 self.actionOptions.setObjectName("actionOptions")
211 self.actionExit = QtWidgets.QAction(MainWindow)
212 self.actionExit.setObjectName("actionExit")
213 self.menu_file_exit = QtWidgets.QAction(MainWindow)
214 self.menu_file_exit.setObjectName("menu_file_exit")
215 self.actionGeneral_Motion_Entry = QtWidgets.QAction(MainWindow)
216 self.actionGeneral_Motion_Entry.setObjectName("actionGeneral_Motion_Entry")
217 self.actionFail_to_Appear_Minor_Traffic_Offense = QtWidgets.QAction(MainWindow)
218
219 self.actionFail_to_Appear_Minor_Traffic_Offense.setObjectName("actionFail_to_Appear_Mino
220 r_Traffic_Offense")
221 self.actionOmnibus_Motion_Form = QtWidgets.QAction(MainWindow)
222 self.actionOmnibus_Motion_Form.setObjectName("actionOmnibus_Motion_Form")
223 self.actionJury_Instructions = QtWidgets.QAction(MainWindow)
224 self.actionJury_Instructions.setObjectName("actionJury_Instructions")
225 self.actioncontrollers = QtWidgets.QAction(MainWindow)
226 self.actioncontrollers.setObjectName("actioncontrollers")
227 self.actionTemplates = QtWidgets.QAction(MainWindow)
228 self.actionTemplates.setObjectName("actionTemplates")
229 self.menuFile.addAction(self.actionOpen)
230 self.menuFile.addAction(self.actionOptions)
```

```

231     self.menuFile.addAction(self.actionExit)
232     self.menuFile.addSeparator()
233     self.menuFile.addAction(self.menu_file_exit)
234     self.menuSettings.addAction(self.actioncontrollers)
235     self.menubar.addAction(self.menuFile.menuAction())
236     self.menubar.addAction(self.menuSettings.menuAction())
237     self.menubar.addAction(self.menuView.menuAction())
238
239     self.retranslateUi(MainWindow)
240     QtCore.QMetaObject.connectSlotsByName(MainWindow)
241
242     def retranslateUi(self, MainWindow):
243         _translate = QtCore.QCoreApplication.translate
244         MainWindow.setWindowTitle(_translate("MainWindow", "MuniEntry - ver 0.2.0 -
245 alpha"))
246         self.FinalJudgmentEntryButton_6.setText(_translate("MainWindow", "Entry"))
247         self.FinalJudgmentEntryButton_7.setText(_translate("MainWindow", "Entry"))
248         self.MinorMisdemeanorTrafficButton_2.setText(_translate("MainWindow", "Entry"))
249         self.FinalJudgmentEntryButton_8.setText(_translate("MainWindow", "Entry"))
250         self.FinalJudgmentEntryButton_9.setText(_translate("MainWindow", "Entry"))
251         self.FinalJudgmentEntryButton_10.setText(_translate("MainWindow", "Entry"))
252         self.label_2.setText(_translate("MainWindow", "Traffic Entries"))
253         self.label_3.setText(_translate("MainWindow", "Criminal Entries"))
254         self.label_4.setText(_translate("MainWindow", "Civil Entries"))
255         self.FinalJudgmentEntryButton_11.setText(_translate("MainWindow", "Entry"))
256         self.FinalJudgmentEntryButton_12.setText(_translate("MainWindow", "Entry"))
257         self.MinorMisdemeanorTrafficButton_3.setText(_translate("MainWindow", "Entry"))
258         self.FinalJudgmentEntryButton_13.setText(_translate("MainWindow", "Entry"))
259         self.FinalJudgmentEntryButton_14.setText(_translate("MainWindow", "Entry"))
260         self.FinalJudgmentEntryButton_15.setText(_translate("MainWindow", "Entry"))
261         self.GreenSheetButton.setText(_translate("MainWindow", "Entry"))
262         self.FinalJudgmentEntryButton_5.setText(_translate("MainWindow", "Entry"))
263         self.MinorMisdemeanorTrafficButton.setText(_translate("MainWindow", "Minor
264 Misdemeanor"))
265         self.FinalJudgmentEntryButton_2.setText(_translate("MainWindow", "Entry"))
266         self.FinalJudgmentEntryButton_3.setText(_translate("MainWindow", "Entry"))
267         self.FinalJudgmentEntryButton_4.setText(_translate("MainWindow", "Entry"))
268         self.label.setText(_translate("MainWindow", "Judicial Officer:"))
269         self.hemmeter_radioButton.setText(_translate("MainWindow", "Judge Hemmeter"))
270         self.rohrer_radioButton.setText(_translate("MainWindow", "Judge Rohrer"))
271         self.pelanda_radioButton.setText(_translate("MainWindow", "Magistrate Pelanda"))
272         self.bunner_radioButton.setText(_translate("MainWindow", "Magistrate Bunner"))
273         self.menuFile.setTitle(_translate("MainWindow", "File"))
274         self.menuSettings.setTitle(_translate("MainWindow", "Settings"))
275         self.menuView.setTitle(_translate("MainWindow", "View"))
276         self.actionOpen.setText(_translate("MainWindow", "Open"))

```

```
277         self.actionOptions.setText(_translate("MainWindow", "Options"))
278         self.actionExit.setText(_translate("MainWindow", "Templates"))
279         self.menu_file_exit.setText(_translate("MainWindow", "Exit"))
280         self.actionGeneral_Motion_Entry.setText(_translate("MainWindow", "General Motion
281 Entry"))
282         self.actionFail_to_Appear_Minor_Traffic_Offense.setText(_translate("MainWindow",
283 "Fail to Appear - Minor Traffic Offense"))
284         self.actionOmnibus_Motion_Form.setText(_translate("MainWindow", "Omnibus
285 Motion Form"))
286         self.actionJury_Instructions.setText(_translate("MainWindow", "Jury Instructions"))
287         self.actioncontrollers.setText(_translate("MainWindow", "controllers"))
288         self.actionTemplates.setText(_translate("MainWindow", "Templates"))
```

```

1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file
4  'resources/ui/MinorMisdemeanorDialog.ui'
5  #
6  # Created by: PyQt5 UI code generator 5.15.4
7  #
8  # WARNING: Any manual changes made to this file will be lost when pyuic5 is
9  # run again. Do not edit this file unless you know what you are doing.
10
11
12  from PyQt5 import QtCore, QtGui, QtWidgets
13
14
15  class Ui\_MinorMisdemeanorDialog(object):
16      def setupUi(self, MinorMisdemeanorDialog):
17          MinorMisdemeanorDialog.setObjectName("MinorMisdemeanorDialog")
18          MinorMisdemeanorDialog.setWindowModality(QtCore.Qt.NonModal)
19          MinorMisdemeanorDialog.resize(1000, 775)
20          sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
21  QtWidgets.QSizePolicy.Preferred)
22          sizePolicy.setHorizontalStretch(0)
23          sizePolicy.setVerticalStretch(0)
24
25          sizePolicy.setHeightForWidth(MinorMisdemeanorDialog.sizePolicy().hasHeightForWidth())
26          MinorMisdemeanorDialog.setSizePolicy(sizePolicy)
27          MinorMisdemeanorDialog.setMinimumSize(QtCore.QSize(1000, 775))
28          MinorMisdemeanorDialog.setMaximumSize(QtCore.QSize(2500, 3500))
29          palette = QtGui.QPalette()
30          brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
31          brush.setStyle(QtCore.Qt.SolidPattern)
32          palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
33          brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
34          brush.setStyle(QtCore.Qt.SolidPattern)
35          palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
36          brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
37          brush.setStyle(QtCore.Qt.SolidPattern)
38          palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
39          brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
40          brush.setStyle(QtCore.Qt.SolidPattern)
41          palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)
42          brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
43          brush.setStyle(QtCore.Qt.SolidPattern)
44          palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)
45          brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
46          brush.setStyle(QtCore.Qt.SolidPattern)

```

```

47 palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)
48 brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
49 brush.setStyle(QtCore.Qt.SolidPattern)
50 palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)
51 brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
52 brush.setStyle(QtCore.Qt.SolidPattern)
53 palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Base, brush)
54 brush = QtGui.QBrush(QtGui.QColor(25, 49, 91))
55 brush.setStyle(QtCore.Qt.SolidPattern)
56 palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
57 MinorMisdemeanorDialog.setPalette(palette)
58 font = QtGui.QFont()
59 font.setFamily("Palatino Linotype")
60 font.setPointSize(11)
61 font.setBold(True)
62 font.setItalic(False)
63 font.setWeight(75)
64 MinorMisdemeanorDialog.setFont(font)
65 MinorMisdemeanorDialog.setToolTip("")
66 MinorMisdemeanorDialog.setStyleSheet("background-color: rgb(25, 49, 91);\n"
67 "font: 75 11pt \"Palatino Linotype\";\n"
68 "font-weight: bold;")
69 MinorMisdemeanorDialog.setModal(True)
70 self.frame = QtWidgets.QFrame(MinorMisdemeanorDialog)
71 self.frame.setGeometry(QtCore.QRect(10, 110, 981, 101))
72 sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
73 QtWidgets.QSizePolicy.Maximum)
74 sizePolicy.setHorizontalStretch(0)
75 sizePolicy.setVerticalStretch(0)
76 sizePolicy.setHeightForWidth(self.frame.sizePolicy().hasHeightForWidth())
77 self.frame.setSizePolicy(sizePolicy)
78 self.frame.setAutoFillBackground(False)
79 self.frame.setStyleSheet("background-color: rgb(240, 240, 240);\n"
80 "\n"
81 "")
82 self.frame.setFrameShape(QtWidgets.QFrame.Box)
83 self.frame.setFrameShadow(QtWidgets.QFrame.Plain)
84 self.frame.setLineWidth(2)
85 self.frame.setObjectName("frame")
86 self.layoutWidget = QtWidgets.QWidget(self.frame)
87 self.layoutWidget.setGeometry(QtCore.QRect(833, 10, 143, 81))
88 self.layoutWidget.setObjectName("layoutWidget")
89 self.gridLayout = QtWidgets.QGridLayout(self.layoutWidget)
90 self.gridLayout.setContentsMargins(0, 0, 0, 0)
91 self.gridLayout.setObjectName("gridLayout")
92 self.clear_fields_charge_Button = QtWidgets.QPushButton(self.layoutWidget)

```



```
93 self.clear_fields_charge_Button.setFocusPolicy(QtCore.Qt.NoFocus)
94 self.clear_fields_charge_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
95 self.clear_fields_charge_Button.setObjectName("clear_fields_charge_Button")
96 self.gridLayout.addWidget(self.clear_fields_charge_Button, 0, 0, 1, 1)
97 self.add_charge_Button = QtWidgets.QPushButton(self.layoutWidget)
98 self.add_charge_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
99 self.add_charge_Button.setObjectName("add_charge_Button")
100 self.gridLayout.addWidget(self.add_charge_Button, 2, 0, 1, 1)
101 self.layoutWidget1 = QtWidgets.QWidget(self.frame)
102 self.layoutWidget1.setGeometry(QtCore.QRect(10, 9, 811, 81))
103 self.layoutWidget1.setObjectName("layoutWidget1")
104 self.gridLayout_3 = QtWidgets.QGridLayout(self.layoutWidget1)
105 self.gridLayout_3.setContentsMargins(0, 0, 0, 0)
106 self.gridLayout_3.setObjectName("gridLayout_3")
107 self.label_6 = QtWidgets.QLabel(self.layoutWidget1)
108 self.label_6.setObjectName("label_6")
109 self.gridLayout_3.addWidget(self.label_6, 0, 0, 1, 1)
110 self.statute_choice_box = QtWidgets.QComboBox(self.layoutWidget1)
111 self.statute_choice_box.setEnabled(True)
112 self.statute_choice_box.setMinimumSize(QtCore.QSize(300, 0))
113 self.statute_choice_box.setFocusPolicy(QtCore.Qt.WheelFocus)
114 self.statute_choice_box.setStyleSheet("background-color: rgb(255, 255, 255);")
115 self.statute_choice_box.setEditable(True)
116 self.statute_choice_box.setObjectName("statute_choice_box")
117 self.gridLayout_3.addWidget(self.statute_choice_box, 0, 1, 1, 1)
118 self.freeform_entry_checkBox = QtWidgets.QCheckBox(self.layoutWidget1)
119 font = QtGui.QFont()
120 font.setFamily("Palatino Linotype")
121 font.setPointSize(9)
122 font.setBold(False)
123 font.setItalic(False)
124 font.setWeight(9)
125 self.freeform_entry_checkBox.setFont(font)
126 self.freeform_entry_checkBox.setFocusPolicy(QtCore.Qt.NoFocus)
127 self.freeform_entry_checkBox.setStyleSheet("font: 75 9pt \"Palatino Linotype\";")
128 self.freeform_entry_checkBox.setObjectName("freeform_entry_checkBox")
129 self.gridLayout_3.addWidget(self.freeform_entry_checkBox, 0, 2, 1, 2)
130 self.label_15 = QtWidgets.QLabel(self.layoutWidget1)
131 self.label_15.setObjectName("label_15")
132 self.gridLayout_3.addWidget(self.label_15, 1, 0, 1, 1)
133 self.offense_choice_box = QtWidgets.QComboBox(self.layoutWidget1)
134 self.offense_choice_box.setMinimumSize(QtCore.QSize(300, 0))
135 self.offense_choice_box.setStyleSheet("background-color: rgb(255, 255, 255);")
136 self.offense_choice_box.setEditable(True)
137 self.offense_choice_box.setObjectName("offense_choice_box")
138 self.gridLayout_3.addWidget(self.offense_choice_box, 1, 1, 1, 1)
```



```
139 self.label_23 = QtWidgets.QLabel(self.layoutWidget1)
140 self.label_23.setObjectName("label_23")
141 self.gridLayout_3.addWidget(self.label_23, 1, 2, 1, 1)
142 self.degree_choice_box = QtWidgets.QComboBox(self.layoutWidget1)
143 self.degree_choice_box.setEnabled(True)
144 self.degree_choice_box.setFocusPolicy(QtCore.Qt.NoFocus)
145 self.degree_choice_box.setStyleSheet("background-color: rgb(255, 255, 255);")
146 self.degree_choice_box.setObjectName("degree_choice_box")
147 self.degree_choice_box.addItem("")
148 self.degree_choice_box.addItem("")
149 self.degree_choice_box.addItem("")
150 self.degree_choice_box.addItem("")
151 self.degree_choice_box.addItem("")
152 self.degree_choice_box.addItem("")
153 self.gridLayout_3.addWidget(self.degree_choice_box, 1, 3, 1, 1)
154 self.frame_2 = QtWidgets.QFrame(MinorMisdemeanorDialog)
155 self.frame_2.setGeometry(QtCore.QRect(10, 220, 981, 331))
156 font = QtGui.QFont()
157 font.setFamily("Palatino Linotype")
158 font.setPointSize(11)
159 font.setBold(True)
160 font.setItalic(False)
161 font.setWeight(75)
162 self.frame_2.setFont(font)
163 self.frame_2.setStyleSheet("background-color: rgb(240, 240, 240);")
164 self.frame_2 setFrameShape(QtWidgets.QFrame.Box)
165 self.frame_2 setFrameShadow(QtWidgets.QFrame.Plain)
166 self.frame_2.setLineWidth(2)
167 self.frame_2.setObjectName("frame_2")
168 self.layoutWidget_2 = QtWidgets.QWidget(self.frame_2)
169 self.layoutWidget_2.setGeometry(QtCore.QRect(10, 10, 961, 271))
170 self.layoutWidget_2.setObjectName("layoutWidget_2")
171 self.charges_gridLayout = QtWidgets.QGridLayout(self.layoutWidget_2)
172 self.charges_gridLayout.setContentsMargins(0, 0, 0, 0)
173 self.charges_gridLayout.setObjectName("charges_gridLayout")
174 self.label_14 = QtWidgets.QLabel(self.layoutWidget_2)
175 font = QtGui.QFont()
176 font.setFamily("Palatino Linotype")
177 font.setPointSize(11)
178 font.setBold(True)
179 font.setItalic(False)
180 font.setWeight(75)
181 self.label_14.setFont(font)
182 self.label_14.setObjectName("label_14")
183 self.charges_gridLayout.addWidget(self.label_14, 4, 0, 1, 1)
184 self.plea_label_1 = QtWidgets.QLabel(self.layoutWidget_2)
```

```
185 font = QtGui.QFont()
186 font.setFamily("Palatino Linotype")
187 font.setPointSize(11)
188 font.setBold(True)
189 font.setItalic(False)
190 font.setWeight(75)
191 self.plea_label_1.setFont(font)
192 self.plea_label_1.setObjectName("plea_label_1")
193 self.charges_gridLayout.addWidget(self.plea_label_1, 3, 0, 1, 1)
194 self.offense_label_1 = QtWidgets.QLabel(self.layoutWidget_2)
195 font = QtGui.QFont()
196 font.setFamily("Palatino Linotype")
197 font.setPointSize(11)
198 font.setBold(True)
199 font.setItalic(False)
200 font.setWeight(75)
201 self.offense_label_1.setFont(font)
202 self.offense_label_1.setWordWrap(True)
203 self.offense_label_1.setObjectName("offense_label_1")
204 self.charges_gridLayout.addWidget(self.offense_label_1, 0, 0, 1, 1)
205 self.label_19 = QtWidgets.QLabel(self.layoutWidget_2)
206 font = QtGui.QFont()
207 font.setFamily("Palatino Linotype")
208 font.setPointSize(11)
209 font.setBold(True)
210 font.setItalic(False)
211 font.setWeight(75)
212 self.label_19.setFont(font)
213 self.label_19.setObjectName("label_19")
214 self.charges_gridLayout.addWidget(self.label_19, 1, 0, 1, 1)
215 self.label_17 = QtWidgets.QLabel(self.layoutWidget_2)
216 font = QtGui.QFont()
217 font.setFamily("Palatino Linotype")
218 font.setPointSize(11)
219 font.setBold(True)
220 font.setItalic(False)
221 font.setWeight(75)
222 self.label_17.setFont(font)
223 self.label_17.setObjectName("label_17")
224 self.charges_gridLayout.addWidget(self.label_17, 5, 0, 1, 1)
225 self.label_4 = QtWidgets.QLabel(self.layoutWidget_2)
226 self.label_4.setText("")
227 self.label_4.setObjectName("label_4")
228 self.charges_gridLayout.addWidget(self.label_4, 7, 0, 1, 1)
229 self.label_20 = QtWidgets.QLabel(self.layoutWidget_2)
230 font = QtGui.QFont()
```

```
231 font.setFamily("Palatino Linotype")
232 font.setPointSize(11)
233 font.setBold(True)
234 font.setItalic(False)
235 font.setWeight(75)
236 self.label_20.setFont(font)
237 self.label_20.setObjectName("label_20")
238 self.charges_gridLayout.addWidget(self.label_20, 2, 0, 1, 1)
239 self.label_10 = QtWidgets.QLabel(self.layoutWidget_2)
240 font = QtGui.QFont()
241 font.setFamily("Palatino Linotype")
242 font.setPointSize(11)
243 font.setBold(True)
244 font.setItalic(False)
245 font.setWeight(75)
246 self.label_10.setFont(font)
247 self.label_10.setObjectName("label_10")
248 self.charges_gridLayout.addWidget(self.label_10, 6, 0, 1, 1)
249 self.label_5 = QtWidgets.QLabel(self.layoutWidget_2)
250 self.label_5.setText("")
251 self.label_5.setObjectName("label_5")
252 self.charges_gridLayout.addWidget(self.label_5, 8, 0, 1, 1)
253 self.guilty_all_Button = QtWidgets.QPushButton(self.frame_2)
254 self.guilty_all_Button.setGeometry(QtCore.QRect(10, 290, 141, 34))
255 self.guilty_all_Button.setFocusPolicy(QtCore.Qt.NoFocus)
256 self.guilty_all_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
257 self.guilty_all_Button.setObjectName("guilty_all_Button")
258 self.no_contest_all_Button = QtWidgets.QPushButton(self.frame_2)
259 self.no_contest_all_Button.setGeometry(QtCore.QRect(170, 290, 141, 34))
260 self.no_contest_all_Button.setFocusPolicy(QtCore.Qt.NoFocus)
261 self.no_contest_all_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
262 self.no_contest_all_Button.setObjectName("no_contest_all_Button")
263 self.costs_and_fines_Button = QtWidgets.QPushButton(self.frame_2)
264 self.costs_and_fines_Button.setGeometry(QtCore.QRect(330, 290, 141, 34))
265 self.costs_and_fines_Button.setFocusPolicy(QtCore.Qt.NoFocus)
266 self.costs_and_fines_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
267 self.costs_and_fines_Button.setFlat(False)
268 self.costs_and_fines_Button.setObjectName("costs_and_fines_Button")
269 self.frame_3 = QtWidgets.QFrame(MinorMisdemeanorDialog)
270 self.frame_3.setGeometry(QtCore.QRect(10, 10, 981, 91))
271 sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
272 QtWidgets.QSizePolicy.Fixed)
273 sizePolicy.setHorizontalStretch(0)
274 sizePolicy.setVerticalStretch(0)
275 sizePolicy.setHeightForWidth(self.frame_3.sizePolicy().hasHeightForWidth())
276 self.frame_3.setSizePolicy(sizePolicy)
```

```

277         self.frame_3.setStyleSheet("background-color: rgb(240, 240, 240);\n"
278 "border-color: rgb(160, 48, 35);")
279         self.frame_3.setFrameShape(QtWidgets.QFrame.Box)
280         self.frame_3.setFrameShadow(QtWidgets.QFrame.Plain)
281         self.frame_3.setLineWidth(2)
282         self.frame_3.setObjectName("frame_3")
283         self.clear_fields_case_Button = QtWidgets.QPushButton(self.frame_3)
284         self.clear_fields_case_Button.setGeometry(QtCore.QRect(835, 50, 141, 33))
285         self.clear_fields_case_Button.setFocusPolicy(QtCore.Qt.NoFocus)
286         self.clear_fields_case_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
287         self.clear_fields_case_Button.setObjectName("clear_fields_case_Button")
288         self.cancel_Button = QtWidgets.QPushButton(self.frame_3)
289         self.cancel_Button.setGeometry(QtCore.QRect(835, 10, 141, 33))
290         self.cancel_Button.setFocusPolicy(QtCore.Qt.NoFocus)
291         self.cancel_Button.setStyleSheet("background-color: rgb(160, 160, 160);")
292         self.cancel_Button.setObjectName("cancel_Button")
293         self.layoutWidget2 = QtWidgets.QWidget(self.frame_3)
294         self.layoutWidget2.setGeometry(QtCore.QRect(11, 10, 811, 71))
295         self.layoutWidget2.setObjectName("layoutWidget2")
296         self.gridLayout_2 = QtWidgets.QGridLayout(self.layoutWidget2)
297         self.gridLayout_2.setContentsMargins(0, 0, 0, 0)
298         self.gridLayout_2.setObjectName("gridLayout_2")
299         self.label = QtWidgets.QLabel(self.layoutWidget2)
300         self.label.setObjectName("label")
301         self.gridLayout_2.addWidget(self.label, 0, 0, 1, 1)
302         self.defendant_first_name_lineEdit = QtWidgets.QLineEdit(self.layoutWidget2)
303         self.defendant_first_name_lineEdit.setMaximumSize(QtCore.QSize(500, 16777215))
304         self.defendant_first_name_lineEdit.setStyleSheet("background-color: rgb(255, 255,
305 255);")
306         self.defendant_first_name_lineEdit.setObjectName("defendant_first_name_lineEdit")
307         self.gridLayout_2.addWidget(self.defendant_first_name_lineEdit, 0, 1, 1, 1)
308         self.label_26 = QtWidgets.QLabel(self.layoutWidget2)
309         self.label_26.setObjectName("label_26")
310         self.gridLayout_2.addWidget(self.label_26, 0, 2, 1, 1)
311         self.plea_trial_date = QtWidgets.QDateEdit(self.layoutWidget2)
312         self.plea_trial_date.setFocusPolicy(QtCore.Qt.NoFocus)
313         self.plea_trial_date.setStyleSheet("background-color: rgb(255, 255, 255);")
314         self.plea_trial_date.setMinimumDate(QtCore.QDate(2021, 1, 1))
315         self.plea_trial_date.setCalendarPopup(True)
316         self.plea_trial_date.setDate(QtCore.QDate(2021, 1, 1))
317         self.plea_trial_date.setObjectName("plea_trial_date")
318         self.gridLayout_2.addWidget(self.plea_trial_date, 0, 3, 1, 1)
319         self.label_2 = QtWidgets.QLabel(self.layoutWidget2)
320         self.label_2.setObjectName("label_2")
321         self.gridLayout_2.addWidget(self.label_2, 1, 0, 1, 1)
322         self.defendant_last_name_lineEdit = QtWidgets.QLineEdit(self.layoutWidget2)

```

```
323     self.defendant_last_name_lineEdit.setEnabled(True)
324     self.defendant_last_name_lineEdit.setStyleSheet("background-color: rgb(255, 255,
325 255);")
326     self.defendant_last_name_lineEdit.setObjectName("defendant_last_name_lineEdit")
327     self.gridLayout_2.addWidget(self.defendant_last_name_lineEdit, 1, 1, 1, 1)
328     self.label_3 = QtWidgets.QLabel(self.layoutWidget2)
329     self.label_3.setObjectName("label_3")
330     self.gridLayout_2.addWidget(self.label_3, 1, 2, 1, 1)
331     self.case_number_lineEdit = QtWidgets.QLineEdit(self.layoutWidget2)
332     self.case_number_lineEdit.setMaximumSize(QtCore.QSize(500, 16777215))
333     self.case_number_lineEdit.setStyleSheet("background-color: rgb(255, 255, 255);")
334     self.case_number_lineEdit.setObjectName("case_number_lineEdit")
335     self.gridLayout_2.addWidget(self.case_number_lineEdit, 1, 3, 1, 1)
336     self.frame_4 = QtWidgets.QFrame(MinorMisdemeanorDialog)
337     self.frame_4.setGeometry(QtCore.QRect(10, 560, 491, 121))
338     self.frame_4.setStyleSheet("background-color: rgb(240, 240, 240);")
339     self.frame_4.setFrameShape(QtWidgets.QFrame.Box)
340     self.frame_4.setFrameShadow(QtWidgets.QFrame.Plain)
341     self.frame_4.setLineWidth(2)
342     self.frame_4.setObjectName("frame_4")
343     self.layoutWidget3 = QtWidgets.QWidget(self.frame_4)
344     self.layoutWidget3.setGeometry(QtCore.QRect(10, 10, 471, 99))
345     self.layoutWidget3.setObjectName("layoutWidget3")
346     self.formLayout = QtWidgets.QFormLayout(self.layoutWidget3)
347     self.formLayout.setContentsMargins(0, 0, 0, 0)
348     self.formLayout.setObjectName("formLayout")
349     self.label_11 = QtWidgets.QLabel(self.layoutWidget3)
350     self.label_11.setObjectName("label_11")
351     self.formLayout.setWidget(0, QtWidgets.QFormLayout.LabelRole, self.label_11)
352     self.court_costs_box = QtWidgets.QComboBox(self.layoutWidget3)
353     self.court_costs_box.setStyleSheet("background-color: rgb(255, 255, 255);")
354     self.court_costs_box.setObjectName("court_costs_box")
355     self.court_costs_box.addItem("")
356     self.court_costs_box.addItem("")
357     self.formLayout.setWidget(0, QtWidgets.QFormLayout.FieldRole, self.court_costs_box)
358     self.label_9 = QtWidgets.QLabel(self.layoutWidget3)
359     self.label_9.setObjectName("label_9")
360     self.formLayout.setWidget(1, QtWidgets.QFormLayout.LabelRole, self.label_9)
361     self.ability_to_pay_box = QtWidgets.QComboBox(self.layoutWidget3)
362     self.ability_to_pay_box.setStyleSheet("background-color: rgb(255, 255, 255);")
363     self.ability_to_pay_box.setObjectName("ability_to_pay_box")
364     self.ability_to_pay_box.addItem("")
365     self.ability_to_pay_box.addItem("")
366     self.ability_to_pay_box.addItem("")
367     self.ability_to_pay_box.addItem("")
368     self.formLayout.setWidget(1, QtWidgets.QFormLayout.FieldRole,
```



```

369 self.ability_to_pay_box)
370     self.label_18 = QtWidgets.QLabel(self.layoutWidget3)
371     self.label_18.setObjectName("label_18")
372     self.formLayout.setWidget(2, QtWidgets.QFormLayout.LabelRole, self.label_18)
373     self.balance_due_date = QtWidgets.QDateEdit(self.layoutWidget3)
374     self.balance_due_date.setStyleSheet("background-color: rgb(255, 255, 255);")
375     self.balance_due_date.setObjectName("balance_due_date")
376     self.formLayout.setWidget(2, QtWidgets.QFormLayout.FieldRole,
377 self.balance_due_date)
378     self.create_entry_Button = QtWidgets.QPushButton(MinorMisdemeanorDialog)
379     self.create_entry_Button.setGeometry(QtCore.QRect(810, 740, 181, 31))
380     self.create_entry_Button.setStyleSheet("background-color: rgb(0, 255, 0);")
381     self.create_entry_Button.setAutoDefault(False)
382     self.create_entry_Button.setObjectName("create_entry_Button")
383     self.frame_5 = QtWidgets.QFrame(MinorMisdemeanorDialog)
384     self.frame_5.setGeometry(QtCore.QRect(10, 690, 491, 81))
385     self.frame_5.setStyleSheet("background-color: rgb(240, 240, 240);")
386     self.frame_5.setFrameShape(QtWidgets.QFrame.Box)
387     self.frame_5.setFrameShadow(QtWidgets.QFrame.Plain)
388     self.frame_5.setLineWidth(2)
389     self.frame_5.setObjectName("frame_5")
390     self.layoutWidget4 = QtWidgets.QWidget(self.frame_5)
391     self.layoutWidget4.setGeometry(QtCore.QRect(11, 10, 471, 66))
392     self.layoutWidget4.setObjectName("layoutWidget4")
393     self.formLayout_3 = QtWidgets.QFormLayout(self.layoutWidget4)
394     self.formLayout_3.setContentsMargins(0, 0, 0, 0)
395     self.formLayout_3.setObjectName("formLayout_3")
396     self.label_21 = QtWidgets.QLabel(self.layoutWidget4)
397     self.label_21.setObjectName("label_21")
398     self.formLayout_3.setWidget(0, QtWidgets.QFormLayout.LabelRole, self.label_21)
399     self.fra_in_file_box = QtWidgets.QComboBox(self.layoutWidget4)
400     self.fra_in_file_box.setStyleSheet("background-color: rgb(255, 255, 255);")
401     self.fra_in_file_box.setObjectName("fra_in_file_box")
402     self.fra_in_file_box.addItem("")
403     self.fra_in_file_box.addItem("")
404     self.fra_in_file_box.addItem("")
405     self.formLayout_3.setWidget(0, QtWidgets.QFormLayout.FieldRole,
406 self.fra_in_file_box)
407     self.label_22 = QtWidgets.QLabel(self.layoutWidget4)
408     self.label_22.setObjectName("label_22")
409     self.formLayout_3.setWidget(1, QtWidgets.QFormLayout.LabelRole, self.label_22)
410     self.fra_in_court_box = QtWidgets.QComboBox(self.layoutWidget4)
411     self.fra_in_court_box.setStyleSheet("background-color: rgb(255, 255, 255);")
412     self.fra_in_court_box.setObjectName("fra_in_court_box")
413     self.fra_in_court_box.addItem("")
414     self.fra_in_court_box.addItem("")

```

```

415         self.fra_in_court_box.addItem("")
416         self.formLayout_3.setWidget(1, QtWidgets.QFormLayout.FieldRole,
417 self.fra_in_court_box)
418         self.frame_6 = QtWidgets.QFrame(MinorMisdemeanorDialog)
419         self.frame_6.setGeometry(QtCore.QRect(510, 560, 481, 171))
420         self.frame_6.setStyleSheet("background-color: rgb(240, 240, 240);\n"
421 "font: 75 9pt \"Palatino Linotype\";\n"
422 "font-weight: bold;")
423         self.frame_6.setFrameShape(QtWidgets.QFrame.Box)
424         self.frame_6.setFrameShadow(QtWidgets.QFrame.Plain)
425         self.frame_6.setLineWidth(2)
426         self.frame_6.setObjectName("frame_6")
427         self.label_24 = QtWidgets.QLabel(self.frame_6)
428         self.label_24.setGeometry(QtCore.QRect(14, 10, 451, 16))
429         font = QtGui.QFont()
430         font.setFamily("Palatino Linotype")
431         font.setPointSize(9)
432         font.setBold(True)
433         font.setItalic(False)
434         font.setUnderline(True)
435         font.setWeight(75)
436         self.label_24.setFont(font)
437         self.label_24.setAlignment(QtCore.Qt.AlignCenter)
438         self.label_24.setObjectName("label_24")
439         self.layoutWidget5 = QtWidgets.QWidget(self.frame_6)
440         self.layoutWidget5.setGeometry(QtCore.QRect(10, 30, 461, 131))
441         self.layoutWidget5.setObjectName("layoutWidget5")
442         self.gridLayout_5 = QtWidgets.QGridLayout(self.layoutWidget5)
443         self.gridLayout_5.setContentsMargins(0, 0, 0, 0)
444         self.gridLayout_5.setObjectName("gridLayout_5")
445         self.license_suspension_checkBox = QtWidgets.QCheckBox(self.layoutWidget5)
446         self.license_suspension_checkBox.setObjectName("license_suspension_checkBox")
447         self.gridLayout_5.addWidget(self.license_suspension_checkBox, 0, 0, 1, 1)
448         self.checkBox_4 = QtWidgets.QCheckBox(self.layoutWidget5)
449         self.checkBox_4.setEnabled(False)
450         self.checkBox_4.setFocusPolicy(QtCore.Qt.NoFocus)
451         self.checkBox_4.setObjectName("checkBox_4")
452         self.gridLayout_5.addWidget(self.checkBox_4, 0, 1, 1, 1)
453         self.community_service_checkBox = QtWidgets.QCheckBox(self.layoutWidget5)
454         self.community_service_checkBox.setObjectName("community_service_checkBox")
455         self.gridLayout_5.addWidget(self.community_service_checkBox, 1, 0, 1, 1)
456         self.checkBox_5 = QtWidgets.QCheckBox(self.layoutWidget5)
457         self.checkBox_5.setEnabled(False)
458         self.checkBox_5.setFocusPolicy(QtCore.Qt.NoFocus)
459         self.checkBox_5.setObjectName("checkBox_5")
460         self.gridLayout_5.addWidget(self.checkBox_5, 1, 1, 1, 1)

```



```
461 self.community_control_checkBox = QtWidgets.QCheckBox(self.layoutWidget5)
462 self.community_control_checkBox.setEnabled(False)
463 self.community_control_checkBox.setObjectName("community_control_checkBox")
464 self.gridLayout_5.addWidget(self.community_control_checkBox, 2, 0, 1, 1)
465 self.checkBox_3 = QtWidgets.QCheckBox(self.layoutWidget5)
466 self.checkBox_3.setEnabled(False)
467 self.checkBox_3.setFocusPolicy(QtCore.Qt.NoFocus)
468 self.checkBox_3.setObjectName("checkBox_3")
469 self.gridLayout_5.addWidget(self.checkBox_3, 2, 1, 1, 1)
470 self.checkBox_7 = QtWidgets.QCheckBox(self.layoutWidget5)
471 self.checkBox_7.setEnabled(False)
472 self.checkBox_7.setFocusPolicy(QtCore.Qt.NoFocus)
473 self.checkBox_7.setObjectName("checkBox_7")
474 self.gridLayout_5.addWidget(self.checkBox_7, 3, 0, 1, 1)
475 self.other_conditions_checkBox = QtWidgets.QCheckBox(self.layoutWidget5)
476 self.other_conditions_checkBox.setEnabled(True)
477 self.other_conditions_checkBox.setFocusPolicy(QtCore.Qt.NoFocus)
478 self.other_conditions_checkBox.setObjectName("other_conditions_checkBox")
479 self.gridLayout_5.addWidget(self.other_conditions_checkBox, 3, 1, 1, 1)
480 self.add_conditions_Button = QtWidgets.QPushButton(MinorMisdemeanorDialog)
481 self.add_conditions_Button.setGeometry(QtCore.QRect(510, 740, 181, 31))
482 self.add_conditions_Button.setStyleSheet("background-color: rgb(157, 53, 36);")
483 self.add_conditions_Button.setAutoDefault(False)
484 self.add_conditions_Button.setObjectName("add_conditions_Button")
485
486 self.retranslateUi(MinorMisdemeanorDialog)
487 QtCore.QMetaObject.connectSlotsByName(MinorMisdemeanorDialog)
488 MinorMisdemeanorDialog.setTabOrder(self.defendant_first_name_lineEdit,
489 self.defendant_last_name_lineEdit)
490 MinorMisdemeanorDialog.setTabOrder(self.defendant_last_name_lineEdit,
491 self.case_number_lineEdit)
492 MinorMisdemeanorDialog.setTabOrder(self.case_number_lineEdit,
493 self.statute_choice_box)
494 MinorMisdemeanorDialog.setTabOrder(self.statute_choice_box,
495 self.offense_choice_box)
496 MinorMisdemeanorDialog.setTabOrder(self.offense_choice_box,
497 self.add_charge_Button)
498 MinorMisdemeanorDialog.setTabOrder(self.add_charge_Button,
499 self.ability_to_pay_box)
500 MinorMisdemeanorDialog.setTabOrder(self.ability_to_pay_box, self.balance_due_date)
501 MinorMisdemeanorDialog.setTabOrder(self.balance_due_date, self.fra_in_file_box)
502 MinorMisdemeanorDialog.setTabOrder(self.fra_in_file_box, self.fra_in_court_box)
503 MinorMisdemeanorDialog.setTabOrder(self.fra_in_court_box,
504 self.license_suspension_checkBox)
505 MinorMisdemeanorDialog.setTabOrder(self.license_suspension_checkBox,
506 self.community_service_checkBox)
```

```

507         MinorMisdemeanorDialog.setTabOrder(self.community_service_checkBox,
508 self.community_control_checkBox)
509         MinorMisdemeanorDialog.setTabOrder(self.community_control_checkBox,
510 self.add_conditions_Button)
511         MinorMisdemeanorDialog.setTabOrder(self.add_conditions_Button,
512 self.create_entry_Button)
513
514     def retranslateUi(self, MinorMisdemeanorDialog):
515         _translate = QtCore.QCoreApplication.translate
516         MinorMisdemeanorDialog.setWindowTitle(_translate("MinorMisdemeanorDialog",
517 "Minor Misdemeanor Case Information"))
518         self.clear_fields_charge_Button.setText(_translate("MinorMisdemeanorDialog", "Clear
519 Fields"))
520         self.add_charge_Button.setText(_translate("MinorMisdemeanorDialog", "Add Charge"))
521         self.label_6.setText(_translate("MinorMisdemeanorDialog", "Statute:"))
522         self.freeform_entry_checkBox.setText(_translate("MinorMisdemeanorDialog", "Enable
523 freeform entry for offense/statute"))
524         self.label_15.setText(_translate("MinorMisdemeanorDialog", "Offense:"))
525         self.label_23.setText(_translate("MinorMisdemeanorDialog", "Degree:"))
526         self.degree_choice_box.setItemText(0, _translate("MinorMisdemeanorDialog", "Minor
527 Misdemeanor"))
528         self.degree_choice_box.setItemText(1, _translate("MinorMisdemeanorDialog", "M1"))
529         self.degree_choice_box.setItemText(2, _translate("MinorMisdemeanorDialog", "M2"))
530         self.degree_choice_box.setItemText(3, _translate("MinorMisdemeanorDialog", "M3"))
531         self.degree_choice_box.setItemText(4, _translate("MinorMisdemeanorDialog", "M4"))
532         self.degree_choice_box.setItemText(5, _translate("MinorMisdemeanorDialog",
533 "Unclassified Misdemeanor"))
534         self.label_14.setText(_translate("MinorMisdemeanorDialog", "Finding:"))
535         self.plea_label_1.setText(_translate("MinorMisdemeanorDialog", "Plea:"))
536         self.offense_label_1.setText(_translate("MinorMisdemeanorDialog", "Offense:"))
537         self.label_19.setText(_translate("MinorMisdemeanorDialog", "Statute:"))
538         self.label_17.setText(_translate("MinorMisdemeanorDialog", "Fines:"))
539         self.label_20.setText(_translate("MinorMisdemeanorDialog", "Degree:"))
540         self.label_10.setText(_translate("MinorMisdemeanorDialog", "Fines Suspended:"))
541         self.guilty_all_Button.setText(_translate("MinorMisdemeanorDialog", "Guilty All"))
542         self.no_contest_all_Button.setText(_translate("MinorMisdemeanorDialog", "No Contest
543 All"))
544         self.costs_and_fines_Button.setText(_translate("MinorMisdemeanorDialog",
545 "Costs/Fines"))
546         self.clear_fields_case_Button.setText(_translate("MinorMisdemeanorDialog", "Clear
547 Fields"))
548         self.cancel_Button.setText(_translate("MinorMisdemeanorDialog", "Cancel"))
549         self.label.setText(_translate("MinorMisdemeanorDialog", "Def. First Name:"))
550         self.label_26.setText(_translate("MinorMisdemeanorDialog", "Date:"))
551         self.label_2.setText(_translate("MinorMisdemeanorDialog", "Def. Last Name:"))
552         self.label_3.setText(_translate("MinorMisdemeanorDialog", "Case Number:"))

```

```
553         self.label_11.setText(_translate("MinorMisdemeanorDialog", "Court Costs:"))
554         self.court_costs_box.setItemText(0, _translate("MinorMisdemeanorDialog", "Yes"))
555         self.court_costs_box.setItemText(1, _translate("MinorMisdemeanorDialog", "No"))
556         self.label_9.setText(_translate("MinorMisdemeanorDialog", "Time to pay fines/costs:"))
557         self.ability_to_pay_box.setItemText(0, _translate("MinorMisdemeanorDialog",
558 "forthwith"))
559         self.ability_to_pay_box.setItemText(1, _translate("MinorMisdemeanorDialog", "within
560 30 days"))
561         self.ability_to_pay_box.setItemText(2, _translate("MinorMisdemeanorDialog", "within
562 60 days"))
563         self.ability_to_pay_box.setItemText(3, _translate("MinorMisdemeanorDialog", "within
564 90 days"))
565         self.label_18.setText(_translate("MinorMisdemeanorDialog", "Fines/costs due date:"))
566         self.create_entry_Button.setText(_translate("MinorMisdemeanorDialog", "Create
567 Entry"))
568         self.label_21.setText(_translate("MinorMisdemeanorDialog", "FRA shown in
569 complaint/file:"))
570         self.fra_in_file_box.setItemText(0, _translate("MinorMisdemeanorDialog", "N/A"))
571         self.fra_in_file_box.setItemText(1, _translate("MinorMisdemeanorDialog", "Yes"))
572         self.fra_in_file_box.setItemText(2, _translate("MinorMisdemeanorDialog", "No"))
573         self.label_22.setText(_translate("MinorMisdemeanorDialog", "FRA shown in court:"))
574         self.fra_in_court_box.setItemText(0, _translate("MinorMisdemeanorDialog", "N/A"))
575         self.fra_in_court_box.setItemText(1, _translate("MinorMisdemeanorDialog", "Yes"))
576         self.fra_in_court_box.setItemText(2, _translate("MinorMisdemeanorDialog", "No"))
577         self.label_24.setText(_translate("MinorMisdemeanorDialog", "Additional Conditions"))
578         self.license_suspension_checkBox.setText(_translate("MinorMisdemeanorDialog",
579 "License Suspension"))
580         self.checkBox_4.setText(_translate("MinorMisdemeanorDialog", "Commitment"))
581         self.community_service_checkBox.setText(_translate("MinorMisdemeanorDialog",
582 "Community Service"))
583         self.checkBox_5.setText(_translate("MinorMisdemeanorDialog", "Forfeiture"))
584         self.community_control_checkBox.setText(_translate("MinorMisdemeanorDialog",
585 "Community Control"))
586         self.checkBox_3.setText(_translate("MinorMisdemeanorDialog", "Fingerprinting"))
587         self.checkBox_7.setText(_translate("MinorMisdemeanorDialog", "Vehicle Conditions"))
588         self.other_conditions_checkBox.setText(_translate("MinorMisdemeanorDialog",
589 "Other"))
590         self.add_conditions_Button.setText(_translate("MinorMisdemeanorDialog", "Add
591 Conditions"))
```

```

1  """The controller module for the minor misdemeanor dialog - it is not limited
2  to minor misdemeanors, but does not contain functions to account for jail time.
3  Loads all charges - including non-minor-misdemeanors from a database."""
4  import pathlib
5  from datetime import date, timedelta
6  from loguru import logger
7
8  from PyQt5 import QtCore
9  from PyQt5.QtCore import QDate
10 from PyQt5.QtWidgets import QLabel, QPushButton, QMessageBox, QComboBox, QLineEdit
11 from PyQt5.QtSql import QSqlDatabase, QSqlQuery
12
13 from views.custom_widgets import (
14     PleaComboBox,
15     FindingComboBox,
16     FineLineEdit,
17     FineSuspendedLineEdit,
18     DeleteButton,
19     AmendButton,
20 )
21 from views.minor_misdemeanor_dialog_ui import Ui_MinorMisdemeanorDialog
22 from models.template_types import TEMPLATE_DICT
23 from models.case_information import (
24     CaseInformation,
25     CriminalCharge,
26     AmendOffenseDetails,
27     LicenseSuspensionTerms,
28     CommunityControlTerms,
29     CommunityServiceTerms,
30     OtherConditionsTerms,
31 )
32 from models.messages import TURNS_AT_INTERSECTIONS as TURNS_WARNING
33 from controllers.criminal_dialogs import (
34     BaseCriminalDialog,
35     AddConditionsDialog,
36     AmendOffenseDialog,
37 )
38 from resources.db.DatabaseCreation import create_offense_list, create_statute_list
39
40
41 PATH = str(pathlib.Path().absolute())
42 TEMPLATE_PATH = PATH + "\\resources\\templates\\"
43 SAVE_PATH = PATH + "\\resources\\saved\\"
44 DB_PATH = PATH + "\\resources\\db\\"
45 CHARGES_DATABASE = DB_PATH + "\\charges.sqlite"
46

```

```

47
48 @logger.catch
49 def create_database_connections():
50     """The databases for the application are created upon import of the module, which is done
51     on application startup. The connections to the databases are created, but the opening and
52     closing of the databases is handled by the appropriate class dialog."""
53     offense_database_connection = QSqlDatabase.addDatabase("QSQLITE", "offenses")
54     offense_database_connection.setDatabaseName(CHARGES_DATABASE)
55     statute_database_connection = QSqlDatabase.addDatabase("QSQLITE", "statutes")
56     statute_database_connection.setDatabaseName(CHARGES_DATABASE)
57     return offense_database_connection, statute_database_connection
58
59
60 @logger.catch
61 def open_databases():
62     """
63     https://www.tutorialspoint.com/pyqt/pyqt\_database\_handling.htm
64     https://doc.qt.io/qtforpython/overviews/sql-connecting.html
65     NOTE: If running create_psql_table.py to update database, must delete
66     the old charges.sqlite file to insure it is updated.
67     """
68     database_offenses.open()
69     database_statutes.open()
70
71
72 @logger.catch
73 def close_databases():
74     """Closes any databases that were opened at the start of the dialog."""
75     database_offenses.close()
76     database_offenses.removeDatabase(CHARGES_DATABASE)
77     database_statutes.close()
78     database_statutes.removeDatabase(CHARGES_DATABASE)
79
80
81 class MinorMisdemeanorDialog(BaseCriminalDialog, Ui_MinorMisdemeanorDialog):
82     """The dialog inherits from the BaseCriminalDialog (controller) and the
83     Ui_MinorMisdemeanorDialog (view).
84
85     This dialog is used when there will not be any jail time imposed. It does
86     not inherently limit cases to minor misdemeanors or unclassified
87     misdemeanors, however, it does not include fields to enter jail time."""
88
89     @logger.catch
90     def __init__(self, judicial_officer, parent=None):
91         open_databases()
92         super().__init__(parent)

```

```

93     self.case_information = CaseInformation(judicial_officer)
94     self.modify_view()
95     self.connect_signals_to_slots()
96     self.template = TEMPLATE_DICT.get(self.case_information.judicial_officer.last_name)
97     self.delete_button_list = []
98     self.amend_button_list = []
99
100    @logger.catch
101    def modify_view(self):
102        """The modify view method updates the view that is created on init from the
103        Ui_MinorMisdemeanorDialog. Place items in this method that can't be added
104        directly in QtDesigner so that they don't need to be changed in the view file
105        each time pyuic5 is run."""
106        statute_list = create_statute_list()
107        self.statute_choice_box.addItem(statute_list)
108        self.offense_choice_box.addItem(create_offense_list())
109        self.plea_trial_date.setDate(QDate.currentDate())
110        self.balance_due_date.setDate(QDate.currentDate())
111        self.statute_choice_box.setCurrentText("")
112        self.offense_choice_box.setCurrentText("")
113
114    @logger.catch
115    def connect_signals_to_slots(self):
116        """The method connects any signals to slots. Generally, connecting with
117        pressed is preferred to clicked because a clicked event sends a bool
118        argument to the function. However, clicked is used in some instances
119        because it is a press and release of a button. Using pressed sometimes
120        caused an event to be triggered twice."""
121        self.cancel_Button.pressed.connect(self.close_event)
122        self.clear_fields_case_Button.pressed.connect(self.clear_case_information_fields)
123        self.create_entry_Button.pressed.connect(self.create_entry_process)
124        self.add_conditions_Button.pressed.connect(self.start_add_conditions_dialog)
125        self.add_charge_Button.clicked.connect(self.add_charge_process)
126        self.clear_fields_charge_Button.pressed.connect(self.clear_charge_fields)
127        self.statute_choice_box.currentTextChanged.connect(self.set_offense)
128        self.offense_choice_box.currentTextChanged.connect(self.set_statute)
129        self.fra_in_file_box.currentTextChanged.connect(self.set_fra_in_file)
130        self.fra_in_court_box.currentTextChanged.connect(self.set_fra_in_court)
131        self.ability_to_pay_box.currentTextChanged.connect(self.set_pay_date)
132        self.guilty_all_Button.pressed.connect(self.guilty_all_plea_and_findings)
133        self.no_contest_all_Button.pressed.connect(self.no_contest_all_plea_and_findings)
134        self.costs_and_fines_Button.clicked.connect(self.show_costs_and_fines)
135
136    def clear_case_information_fields(self):
137        self.defendant_first_name_lineEdit.clear()
138        self.defendant_last_name_lineEdit.clear()

```



```

139         self.case_number_lineEdit.clear()
140         self.defendant_first_name_lineEdit.setFocus()
141
142     def clear_charge_fields(self):
143         """Clears the fields that are used for adding a charge. The
144         statute_choice_box and offense_choice_box use the clearEditText
145         method because those boxes are editable."""
146         self.statute_choice_box.clearEditText()
147         self.offense_choice_box.clearEditText()
148
149     @logger.catch
150     def start_amend_offense_dialog(self):
151         """Opens the amend offense dialog as a modal window. The
152         case_information is passed to the dialog class in order to populate
153         the case information banner."""
154         self.update_case_information()
155         AmendOffenseDialog(self.case_information).exec()
156
157     @logger.catch
158     def start_add_conditions_dialog(self):
159         """Opens the add conditions dialog as a modal window. It passes the
160         instance of the MinorMisdemeanorDialog class (self) as an argument
161         so that the AddConditionsDialog can access all data from the
162         MinorMisdemeanorDialog when working in the AddConditionsDialog."""
163         AddConditionsDialog(self).exec()
164
165     @logger.catch
166     def close_event(self):
167         """Place any cleanup items (i.e. close_databases) here that should be
168         called when the entry is created and the dialog closed."""
169         close_databases()
170         self.close_window()
171
172     @logger.catch
173     def add_charge(self):
174         """The add_charge_process, from which this is called, creates a criminal
175         charge object and adds the data in the view to the object. The criminal
176         charge (offense, statute, degree and type) is then added to the case
177         information model (by appending the charge object to the criminal
178         charges list).
179
180         The offense, statute and degree are added to the view by the method
181         add_charge_to_view, not this method. This method is triggered on
182         clicked() of the Add Charge button."""
183         self.criminal_charge.offense = self.offense_choice_box.currentText()
184         self.criminal_charge.statute = self.statute_choice_box.currentText()

```



```

185     self.criminal_charge.degree = self.degree_choice_box.currentText()
186     self.criminal_charge.type = self.set_offense_type()
187     self.case_information.add_charge_to_list(self.criminal_charge)
188     self.add_charge_to_view()
189     self.statute_choice_box.setFocus()
190
191     @logger.catch
192     def add_charge_to_view(self):
193         """Adds the charge that was added through add_charge method to the
194         view/GUI. The first row=0 because of python zero-based indexing. The
195         column is set at one more than the current number of columns because
196         it is the column to which the charge will be added.
197
198         :added_charge_index: - The added charge index is one less than the
199         total charges in charges_list because of zero-based indexing. Thus, if
200         there is one charge, the index of the charge to be added to the
201         charge_dict from the charges_list is 0.
202
203         The python builtin vars function returns the __dict__ attribute of
204         the object.
205
206         The self.criminal_charge.offense added as a parameter for FineLineEdit
207         is the current one added when "Add Charge" is pressed.
208
209         TODO: Refactor so that there isn't a need for a if branch to skip the
210         attribute for charge type."""
211         row = 0
212         column = self.charges_gridLayout.columnCount() + 1
213         added_charge_index = len(self.case_information.charges_list) - 1
214         charge = vars(self.case_information.charges_list[added_charge_index])
215         for value in charge.values():
216             if value is not None:
217                 if value in ["Moving Traffic", "Non-moving Traffic", "Criminal"]:
218                     break
219                 self.charges_gridLayout.addWidget(QLabel(value), row, column)
220                 row += 1
221         self.charges_gridLayout.addWidget(PleaComboBox(), row, column)
222         row += 1
223         self.charges_gridLayout.addWidget(FindingComboBox(), row, column)
224         row += 1
225         self.charges_gridLayout.addWidget(FineLineEdit(self.criminal_charge.offense), row,
226 column)
227         row += 1
228         self.charges_gridLayout.addWidget(FineSuspendedLineEdit(), row, column)
229         row += 1
230         self.add_delete_button_to_view(row, column)

```

```

231         row +=1
232         self.add_amend_button_to_view(row, column)
233
234     def add_delete_button_to_view(self, row, column):
235         delete_button = DeleteButton()
236         self.delete_button_list.append(delete_button)
237         delete_button.pressed.connect(self.delete_charge)
238         self.charges_gridLayout.addWidget(delete_button, row, column)
239
240     def add_amend_button_to_view(self, row, column):
241         amend_button = AmendButton()
242         self.amend_button_list.append(amend_button)
243         amend_button.pressed.connect(self.start_amend_offense_dialog)
244         self.charges_gridLayout.addWidget(amend_button, row, column)
245
246     @logger.catch
247     def delete_charge(self):
248         """Deletes the offense from the case_information.charges list. Then
249         decrements the total charges by one so that other functions using the
250         total charges for indexing are correct."""
251         index = self.delete_button_list.index(self.sender())
252         del self.case_information.charges_list[index]
253         del self.delete_button_list[index]
254         self.delete_charge_from_view()
255         self.statute_choice_box.setFocus()
256
257     @logger.catch
258     def delete_charge_from_view(self):
259         """Uses the delete_button that is indexed to the column to delete the
260         QLabels for the charge."""
261         index = self.charges_gridLayout.indexOf(self.sender())
262         column = self.charges_gridLayout.getItemPosition(index)[1]
263         for row in range(self.charges_gridLayout.rowCount()):
264             layout_item = self.charges_gridLayout.itemAtPosition(row, column)
265             if layout_item is not None:
266                 layout_item.widget().deleteLater()
267                 self.charges_gridLayout.removeItem(layout_item)
268
269     @logger.catch
270     def update_case_information(self):
271         """The method calls functions to update the case information model
272         with the data for the case that is in the fields on the view. This does
273         not update the model
274         with information in the charge fields (offense, statute, plea, etc.)
275         the charge information is transferred to the model upon press of the
276         add charge button.

```

```

277
278     Fields that are updated upon pressed() of createEntryButton = case
279     number, first name, last name, ability to pay time, balance due date,
280     date of plea/trial, operator license number, date of birth, FRA (proof
281     of insurance) in complaint, FRA in court. """
282     self.update_party_information()
283     self.update_costs_and_fines_information()
284     self.add_dispositions_and_fines()
285     self.check_add_conditions()
286     self.calculate_costs_and_fines()
287
288     @logger.catch
289     def update_party_information(self):
290         self.case_information.case_number = self.case_number_lineEdit.text()
291         self.case_information.defendant.first_name = self.defendant_first_name_lineEdit.text()
292         self.case_information.defendant.last_name = self.defendant_last_name_lineEdit.text()
293         self.case_information.plea_trial_date = self.plea_trial_date.date().toString("MMMM dd,
294     yyyy")
295
296     @logger.catch
297     def update_costs_and_fines_information(self):
298         self.case_information.court_costs_ordered = self.court_costs_box.currentText()
299         self.case_information.ability_to_pay_time = self.ability_to_pay_box.currentText()
300         self.case_information.balance_due_date =
301     self.balance_due_date.date().toString("MMMM dd, yyyy")
302
303     @logger.catch
304     def add_dispositions_and_fines(self):
305         """Row 3 - plea, 4 - finding, 5 - fine, 6 fine-suspended.
306         Columns start at 0 for labels and 2 for first entry then 4 etc.
307
308         Column count increases by 2 instead of one due to grid adding two
309         columns when a charge is added (odd numbered column is empty). """
310         column = 2
311         try:
312             for index in range(len(self.case_information.charges_list)):
313                 self.case_information.charges_list[index].plea =
314     self.charges_gridLayout.itemAtPosition(3,column).widget().currentText()
315                 self.case_information.charges_list[index].finding =
316     self.charges_gridLayout.itemAtPosition(4,column).widget().currentText()
317                 self.case_information.charges_list[index].fines_amount =
318     self.charges_gridLayout.itemAtPosition(5,column).widget().text()
319                 if self.charges_gridLayout.itemAtPosition(6,column).widget().text() == "":
320                     self.case_information.charges_list[index].fines_suspended = "0"
321                 else:
322                     self.case_information.charges_list[index].fines_suspended =

```

```

323 self.charges_gridLayout.itemAtPosition(6,column).widget().text()
324     index +=1
325     column +=2
326 except AttributeError:
327     print("Attribute error allowed to pass for lack of widget")
328
329 @logger.catch
330 def check_add_conditions(self):
331     """Checks to see what conditions in the Add Conditions box are checked and then
332     transfers the information from the conditions to case_information model if the
333     box is checked.
334
335     TODO:
336     in future refactor this to have it loop through the different conditions so code
337     doesn't need to be added each time a condition is added."""
338     if self.license_suspension_checkBox.isChecked():
339         self.case_information.license_suspension_details.license_suspension_ordered = (
340             True
341         )
342     if self.community_control_checkBox.isChecked():
343         self.case_information.community_control_terms.community_control_required = (
344             True
345         )
346     if self.community_service_checkBox.isChecked():
347         self.case_information.community_service_terms.community_service_ordered = (
348             True
349         )
350     if self.other_conditions_checkBox.isChecked():
351         self.case_information.other_conditions_details.other_conditions_ordered = (
352             True
353         )
354
355 @logger.catch
356 def calculate_costs_and_fines(self):
357     self.case_information.court_costs = 0
358     if self.court_costs_box.currentText() == "Yes":
359         for index, charge in enumerate(self.case_information.charges_list):
360             if self.case_information.court_costs == 124:
361                 break
362             else:
363                 if charge.type == "Moving Traffic":
364                     if self.case_information.court_costs < 124:
365                         self.case_information.court_costs = 124
366                 elif charge.type == "Criminal":
367                     if self.case_information.court_costs < 114:
368                         self.case_information.court_costs = 114

```

```

369         elif charge.type == "Non-moving Traffic":
370             if self.case_information.court_costs < 95:
371                 self.case_information.court_costs = 95
372     total_fines = 0
373     try:
374         for index, charge in enumerate(self.case_information.charges_list):
375             if charge.fines_amount == "":
376                 charge.fines_amount = 0
377                 total_fines = total_fines + int(charge.fines_amount)
378             self.case_information.total_fines = total_fines
379             total_fines_suspended = 0
380             for index, charge in enumerate(self.case_information.charges_list):
381                 if charge.fines_suspended == "":
382                     charge.fines_suspended = 0
383                     total_fines_suspended = total_fines_suspended + int(charge.fines_suspended)
384             self.case_information.total_fines_suspended = total_fines_suspended
385     except TypeError:
386         print("A type error was allowed to pass - this is because of deleted charge.")
387
388     def show_costs_and_fines(self, bool):
389         """The bool is the toggle from the clicked() of the button pressed. No
390         action is taken with respect to it."""
391         self.update_case_information()
392         message = QMessageBox()
393         message.setIcon(QMessageBox.Information)
394         message.setWindowTitle("Total Costs and Fines")
395         message.setInformativeText("Costs: $" + str(self.case_information.court_costs) +\
396             "\nFines: $" + str(self.case_information.total_fines) +\
397             "\nFines Suspended: $" + str(self.case_information.total_fines_suspended) +\
398             "\n\n*Does not include possible bond forfeiture or other costs \n that may be assessed
399 as a result of prior actions in case. ")
400         total_fines_and_costs = (self.case_information.court_costs +\
401             self.case_information.total_fines) - self.case_information.total_fines_suspended
402         message.setText("Total Costs and Fines Due By Due Date: $" +
403             str(total_fines_and_costs))
404         message.setStandardButtons(QMessageBox.Ok)
405         message.exec_()
406
407     def guilty_all_plea_and_findings(self):
408         """Sets the plea and findings boxes to guilty for all charges currently
409         in the charges_gridLayout."""
410         for column in range(self.charges_gridLayout.columnCount()):
411             try:
412                 if isinstance(self.charges_gridLayout.itemAtPosition(3, column).widget(),
413                     PleaComboBox):
414

```

```

415 self.charges_gridLayout.itemAtPosition(3,column).widget().setCurrentText("Guilty")
416
417 self.charges_gridLayout.itemAtPosition(4,column).widget().setCurrentText("Guilty")
418     column +=1
419     except AttributeError:
420         pass
421     self.set_cursor_to_FineLineEdit()
422
423 def no_contest_all_plea_and_findings(self):
424     """Sets the plea box to no contest and findings boxes to guilty for all
425     charges currently in the charges_gridLayout."""
426     for column in range(self.charges_gridLayout.columnCount()):
427         try:
428             if isinstance(self.charges_gridLayout.itemAtPosition(3, column).widget(),
429 PleaComboBox):
430
431 self.charges_gridLayout.itemAtPosition(3,column).widget().setCurrentText("No Contest")
432
433 self.charges_gridLayout.itemAtPosition(4,column).widget().setCurrentText("Guilty")
434         column +=1
435         except AttributeError:
436             pass
437         self.set_cursor_to_FineLineEdit()
438
439 def set_cursor_to_FineLineEdit(self):
440     for column in range(self.charges_gridLayout.columnCount()):
441         try:
442             if isinstance(self.charges_gridLayout.itemAtPosition(5, column).widget(),
443 FineLineEdit):
444                 self.charges_gridLayout.itemAtPosition(5, column).widget().setFocus()
445                 break
446             except AttributeError:
447                 pass
448
449 @logger.catch
450 def set_fra_in_file(self, current_text):
451     """Sets the FRA (proof of insurance) to true if the view indicates 'yes'
452     that the FRA was shown in the complaint of file."""
453     if current_text == "Yes":
454         self.case_information.fra_in_file = True
455         self.fra_in_court_box.setCurrentText("No")
456     elif current_text == "No":
457         self.case_information.fra_in_file = False
458     else:
459         self.case_information.fra_in_file = None
460

```

```

461 @logger.catch
462 def set_fra_in_court(self, current_text):
463     """Sets the FRA (proof of insurance) to true if the view indicates 'yes'
464     that the FRA was shown in court."""
465     if current_text == "Yes":
466         self.case_information.fra_in_court = True
467     elif current_text == "No":
468         self.case_information.fra_in_court = False
469     else:
470         self.case_information.fra_in_court = None
471
472 def set_offense_type(self):
473     key = self.statute_choice_box.currentText()
474     if self.freeform_entry_checkBox.isChecked():
475         return None
476     query = QSqlQuery(database_statutes)
477     query.prepare("SELECT * FROM charges WHERE " "statute LIKE '% ' || :key || '% '")
478     query.bindValue(":key", key)
479     query.exec()
480     while query.next():
481         statute = query.value(2)
482         offense_type = query.value(4)
483         if statute == key:
484             return offense_type
485
486 @logger.catch
487 def set_statute(self, key):
488     """This method queries based on the offense and then sets the statute
489     and degree based on the offense in the database.
490
491     :key: is the string that is passed by the function each time the field
492     is changed on the view. This is the offense."""
493     if self.freeform_entry_checkBox.isChecked():
494         return None
495     query = QSqlQuery(database_offenses)
496     query.prepare("SELECT * FROM charges WHERE " "offense LIKE '% ' || :key || '% '")
497     query.bindValue(":key", key)
498     query.exec()
499     while query.next():
500         offense = query.value(1)
501         statute = query.value(2)
502         degree = query.value(3)
503         if offense == key:
504             self.statute_choice_box.setCurrentText(statute)
505             self.degree_choice_box.setCurrentText(degree)
506             break

```



```

507
508 @logger.catch
509 def set_offense(self, key):
510     """This method queries based on the statute and then sets the offense
511     and degree based on the statute in the database.
512
513     :key: is the string that is passed by the function each time the field
514     is changed on the view. This is the statute."""
515     if self.freeform_entry_checkBox.isChecked():
516         return None
517     query = QSqlQuery(database_statutes)
518     query.prepare("SELECT * FROM charges WHERE " "statute LIKE '%" || :key || '%"")
519     query.bindValue(":key", key)
520     query.exec()
521     while query.next():
522         offense = query.value(1)
523         statute = query.value(2)
524         degree = query.value(3)
525         if statute == key:
526             self.offense_choice_box.setCurrentText(offense)
527             self.degree_choice_box.setCurrentText(degree)
528             break
529
530 @logger.catch
531 def set_pay_date(self, time_to_pay_text):
532     """Sets the balance of fines and costs to a future date (or today)
533     depending on the selection of ability_to_pay_box. The inner function
534     will move the actual date to the next tuesday per court procedure for
535     show cause hearings being on Tuesday. Would need to be modified if the
536     policy changed."""
537     days_to_add = self.pay_date_dict[time_to_pay_text]
538     future_date = date.today() + timedelta(days_to_add)
539     today = date.today()
540
541     def next_tuesday(future_date, weekday=1):
542         """This function returns the number of days to add to today to set
543         the payment due date out to the Tuesday after the number of days
544         set in the set_pay_date function. The default of 1 for weekday is
545         what sets it to a Tuesday. If it is 0 it would be Monday, 3 would
546         be Wednesday, etc."""
547         days_ahead = weekday - future_date.weekday()
548         if days_ahead <= 0: # Target day already happened this week
549             days_ahead += 7
550         return future_date + timedelta(days_ahead)
551
552     future_date = next_tuesday(future_date, 1)

```

```
553         total_days_to_add = (future_date - today).days
554         self.balance_due_date.setDate(QDate.currentDate().addDays(total_days_to_add))
555
556
557 if __name__ == "__main__":
558     print("MMD ran directly")
559 else:
560     print("MMD ran when imported")
561     database_offenses, database_statutes = create_database_connections()
```

```

1  """Module containing all the data structures for parties in cases."""
2
3  class Party(object):
4      """The base object from which all other party objects are inherited."""
5      def __init__(self, party_type, party_name: str):
6          self.party_type = party_type # Either Person or Entity
7
8
9  class PlaintiffEntity(Party):
10     """A type of party in a case. Generally the State of Ohio in a criminal
11     case."""
12     def __init__(self, party_type: 'Entity'):
13         super().__init__(party_type)
14         self.full_name = party_type.entity_name
15
16
17  class Person(object):
18     """A base class for all people of any party type."""
19     def __init__(self) -> None:
20         self.first_name = None
21         self.last_name = None
22         self.date_of_birth = None
23
24
25  class Defendant(Person):
26     """Class for defendants that inherits from Person class."""
27     def __init__(self):
28         super().__init__()
29
30
31  class JudicialOfficer(Person):
32     """
33     A subclass of a person that is for all different types of judicial
34     officers: Judges, Visiting Judges, Acting Judges, Magistrates.
35
36     In the future may want to add attorney registration number to this class.
37     """
38     def __init__(self, first_name: str, last_name: str, officer_type: str) -> None:
39         super().__init__()
40         self.first_name = first_name
41         self.last_name = last_name
42         self.officer_type = officer_type
43
44
45  class Entity(object):
46     """A base class for non-person parties

```

```
47         (i.e. State of Ohio or a business)."""
48
49     def __init__(self, entity_name: str):
50         self.entity_name = entity_name
51
52
53     """
54     test = JudicialOfficer("Marianne", "Hemmeter", "Judge")
55     print(test.officer_type)
56     test_2 = PlaintiffEntity(Entity("State of Ohio"))
57     print(test_2.party_type)
58     print(test_2.full_name)
59     """
```

```

1  """Module that contains all the template objects for use in the
2  application."""
3  import pathlib
4
5
6  PATH = str(pathlib.Path().absolute())
7  TEMPLATE_PATH = PATH + "\\resources\\templates\\"
8
9
10 class Template():
11     """Template objects contain all the relevant data for each type of template.
12     Changes to content of template should be made on the template directly."""
13     def __init__(self, template_name, template_path):
14         self.template_name = template_name
15         self.template_path = TEMPLATE_PATH + template_path
16
17
18 Magistrate_And_Judge_Final_Judgment_Template = Template(
19     "Traffic Judgment Entry",
20     "Magistrate_And_Judge_Final_Judgment_Template.docx",
21 )
22
23 Judge_Final_Judgment_Template = Template(
24     "Traffic Judgment Entry",
25     "Judge_Final_Judgment_Template.docx",
26 )
27
28
29 TEMPLATE_DICT = {
30     "Bunner": Magistrate_And_Judge_Final_Judgment_Template,
31     "Pelanda": Magistrate_And_Judge_Final_Judgment_Template,
32     "Rohrer": Magistrate_And_Judge_Final_Judgment_Template,
33     "Hemmeter": Magistrate_And_Judge_Final_Judgment_Template,
34 }

```