

```

1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.geometry.*?>
5: <?import javafx.scene.control.*?>
6: <?import java.lang.*?>
7: <?import javafx.scene.layout.*?>
8: <?import javafx.scene.layout.BorderPane?>
9:
10: <Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="net.penguincoders.NER.NERUIController">
11:     <children>
12:         <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0" prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
13:             <font>
14:                 <Font size="20.0" />
15:             </font></TextArea>
16:         <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text="Enter Text For Keyword Extraction" textFill="#176fc3">
17:             <font>
18:                 <Font size="24.0" />
19:             </font>
20:         </Label>
21:         <Button fx:id="extractKeyword" layoutX="255.0" layoutY="454.0" mnemonicParsing="false" onAction="#goToOutputScreen" prefHeight="48.0" prefWidth="232.0" text="Go" textFill="#0c74e4">
22:             <font>
23:                 <Font size="24.0" />
24:             </font>
25:         </Button>
26:         <Button layoutX="651.0" layoutY="538.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="37.0" prefWidth="89.0" text="Back" textFill="#1d53b2">
27:             <font>
28:                 <Font size="20.0" />
29:             </font>
30:         </Button>
31:     </children>
32: </Pane>

```

```

1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.scene.control.*?>
5: <?import java.lang.*?>
6: <?import javafx.scene.layout.*?>
7: <?import javafx.scene.layout.BorderPane?>
8:
9: <BorderPane xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx
x:controller="net.penguincoders.NER.NEROutputController">
10:     <right>
11:         <Pane prefHeight="600.0" prefWidth="800.0" BorderPane.alignment="CENTER">
12:             <children>
13:                 <Label layoutX="253.0" layoutY="60.0" prefHeight="61.0" prefWidth="294.0
" text="Extracted Keywords" textAlignment="CENTER" textFill="#187bb8">
14:                     <font>
15:                         <Font size="28.0" />
16:                     </font>
17:                 </Label>
18:                 <Label layoutX="61.0" layoutY="177.0" prefHeight="51.0" prefWidth="87.0"
text="Person" textFill="#0da4d9">
19:                     <font>
20:                         <Font size="24.0" />
21:                     </font>
22:                 </Label>
23:                 <Label layoutX="308.0" layoutY="177.0" prefHeight="51.0" prefWidth="161.
0" text="Organization" textFill="#0da4d9">
24:                     <font>
25:                         <Font size="24.0" />
26:                     </font>
27:                 </Label>
28:                 <Label layoutX="615.0" layoutY="177.0" prefHeight="51.0" prefWidth="111.
0" text="Location" textFill="#0da4d9">
29:                     <font>
30:                         <Font size="24.0" />
31:                     </font>
32:                 </Label>
33:                 <Separator layoutX="230.0" layoutY="228.0" orientation="VERTICAL" prefHe
ight="240.0" prefWidth="0.0" />
34:                 <Separator layoutX="547.0" layoutY="228.0" orientation="VERTICAL" prefHe
ight="240.0" prefWidth="0.0" />
35:                 <Label fx:id="personLabel" layoutX="48.0" layoutY="239.0" prefHeight="31
9.0" prefWidth="165.0" wrapText="true" />
36:                 <Label fx:id="organizationLabel" layoutX="287.0" layoutY="245.0" prefHei
ght="319.0" prefWidth="250.0" wrapText="true" />
37:                 <Label fx:id="locationLabel" layoutX="596.0" layoutY="245.0" prefHeight=
"319.0" prefWidth="186.0" wrapText="true" />
38:                 <Button fx:id="backButton" layoutX="367.0" layoutY="545.0" mnemonicParsi
ng="false" onAction="#gotToPreviousScreen" prefHeight="35.0" prefWidth="100.0" text="Back">
39:                     <font>
40:                         <Font size="19.0" />
41:                     </font>
42:                 </Button>
43:             </children>
44:         </Pane>
45:     </right>
46: </BorderPane>

```

```

1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.geometry.*?>
5: <?import javafx.scene.control.*?>
6: <?import java.lang.*?>
7: <?import javafx.scene.layout.*?>
8: <?import javafx.scene.layout.BorderPane?>
9:
10: <Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-In
finitiy" prefHeight="600.0" prefWidth="800.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="h
ttp://javafx.com/fxml/1" fx:controller="net.penguincoders.Sentiment.SentimentUIController">
11:     <children>
12:         <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0"
prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
13:             <font>
14:                 <Font size="20.0" />
15:             </font></TextArea>
16:         <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text=
"Enter Text For Sentiment Analysis" textFill="#176fc3">
17:             <font>
18:                 <Font size="24.0" />
19:             </font>
20:         </Label>
21:         <Button fx:id="calculateScore" layoutX="50.0" layoutY="483.0" mnemonicParsing=
"false" onAction="#printLabelOutput" prefHeight="48.0" prefWidth="232.0" text="Calculate Sc
ore" textFill="#2419bc">
22:             <font>
23:                 <Font size="24.0" />
24:             </font>
25:         </Button>
26:         <Label fx:id="scoreLabel" layoutX="345.0" layoutY="483.0" prefHeight="48.0" pr
efWidth="403.0">
27:             <font>
28:                 <Font size="20.0" />
29:             </font></Label>
30:         <Button layoutX="632.0" layoutY="551.0" mnemonicParsing="false" onAction="#got
ToPreviousScreen" prefHeight="25.0" prefWidth="109.0" text="Back" textFill="#1d44c3">
31:             <font>
32:                 <Font size="19.0" />
33:             </font>
34:         </Button>
35:     </children>
36: </Pane>

```

```

1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.scene.control.*?>
5: <?import java.lang.*?>
6: <?import javafx.scene.layout.*?>
7: <?import javafx.scene.layout.AnchorPane?>
8:
9:
10: <Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="net.penguincoders.NLPUIController">
11:     <children>
12:         <Label layoutX="161.0" layoutY="83.0" prefHeight="60.0" prefWidth="478.0" text="Natural Language Processing" textFill="#1273c3">
13:             <font>
14:                 <Font size="32.0" />
15:             </font>
16:         </Label>
17:         <Label layoutX="54.0" layoutY="274.0" prefHeight="53.0" prefWidth="271.0" text="Sentiment Analysis" textFill="#0d82bc">
18:             <font>
19:                 <Font size="25.0" />
20:             </font>
21:         </Label>
22:         <Label layoutX="382.0" layoutY="274.0" prefHeight="53.0" prefWidth="370.0" text="Named Entity Recognition" textFill="#0d82bc">
23:             <font>
24:                 <Font size="25.0" />
25:             </font>
26:         </Label>
27:         <Button fx:id="sentimentAnalysisButton" layoutX="117.0" layoutY="372.0" mnemonicParsing="false" onAction="#sentimentAnalysis" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
28:             <font>
29:                 <Font size="20.0" />
30:             </font>
31:         </Button>
32:         <Button fx:id="NERButton" layoutX="523.0" layoutY="372.0" mnemonicParsing="false" onAction="#NER" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
33:             <font>
34:                 <Font size="20.0" />
35:             </font>
36:         </Button>
37:     </children>
38: </Pane>

```

```
1: package net.penguincoders;
2:
3: import java.util.ArrayList;
4: import java.util.List;
5:
6: import twitter4j.Query;
7: import twitter4j.QueryResult;
8: import twitter4j.Status;
9: import twitter4j.Twitter;
10: import twitter4j.TwitterException;
11: import twitter4j.TwitterFactory;
12:
13: public class TweetManager
14: {
15:     public static ArrayList<String> getTweets(String topic)
16:     {
17:         int count = 0;
18:         Twitter twitter = new TwitterFactory().getInstance();
19:         ArrayList<String> tweetList = new ArrayList<String>();
20:         try
21:         {
22:             Query query = new Query(topic);
23:             QueryResult result;
24:             do
25:             {
26:                 result = twitter.search(query);
27:                 List<Status> tweets = result.getTweets();
28:                 count+=1;
29:                 for (Status tweet : tweets)
30:                 {
31:                     tweetList.add(tweet.getText());
32:                 }
33:             }
34:             while (count != 1);
35:         }
36:         catch (TwitterException te)
37:         {
38:             te.printStackTrace();
39:             System.out.println("Failed to search tweets: " + te.getMessage());
40:         }
41:         return tweetList;
42:     }
43: }
```

```
1: package net.penguincoders;
2:
3: import java.util.ArrayList;
4:
5: public class WhatToThink
6: {
7:     public static void main(String[] args)
8:     {
9:         String topic = "IPL 2019";
10:        int count = 0;
11:        ArrayList<String> tweets = TweetManager.getTweets(topic);
12:        System.out.println("tweets retrieved Done!!!");
13:        NLP.initSentiment();
14:        System.out.println(" init Done!!!");
15:        for(String tweet : tweets)
16:        {
17:            System.out.println(tweet + " : " + NLP.findSentiment(tweet))
;
18:            count++;
19:        }
20:        System.out.println("Done!!! Total = "+count);
21:    }
22: }
```

```
1: package net.penguincoders;
2:
3: import javafx.event.ActionEvent;
4: import javafx.fxml.FXML;
5: import javafx.fxml.FXMLLoader;
6: import javafx.scene.Node;
7: import javafx.scene.Parent;
8: import javafx.scene.Scene;
9: import javafx.scene.control.Button;
10: import javafx.stage.Stage;
11:
12: public class NLPUIController
13: {
14:     @FXML
15:     public Button sentimentAnalysisButton;
16:
17:     @FXML
18:     public Button NERButton;
19:
20:     @FXML
21:     public void sentimentAnalysis(ActionEvent event)throws Exception
22:     {
23:         ((Node)event.getSource()).getScene().getWindow().hide();
24:         FXMLLoader loader = new FXMLLoader(getClass().getResource("Sentiment
/SentimentUI.fxml"));
25:         Parent root = (Parent)loader.load();
26:         Stage output = new Stage();
27:         output.setTitle("Sentiment Analysis");
28:         output.setScene(new Scene(root));
29:         output.show();
30:     }
31:
32:     public void NER(ActionEvent event)throws Exception
33:     {
34:         ((Node)event.getSource()).getScene().getWindow().hide();
35:         FXMLLoader loader = new FXMLLoader(getClass().getResource("NER/NERUI
.fxml"));
36:         Parent root = (Parent)loader.load();
37:         Stage output = new Stage();
38:         output.setTitle("Named Entity Recognition");
39:         output.setScene(new Scene(root));
40:         output.show();
41:     }
42: }
```

```
1: package net.penguincoders;
2:
3: import java.io.FileInputStream;
4:
5: import javafx.application.Application;
6: import javafx.fxml.FXMLLoader;
7: import javafx.scene.Scene;
8: import javafx.scene.layout.Pane;
9: import javafx.stage.Stage;
10:
11: public class NLPUI extends Application
12: {
13:     @Override
14:     public void start(Stage stage) throws Exception
15:     {
16:         //Create the FXMLLoader
17:         FXMLLoader loader = new FXMLLoader();
18:         //SentimentUIController controller = new SentimentUIControll
er();
19:         //loader.setController(controller);
20:         //Path to FXML File
21:         String fxmlPath = "src/net/penguincoders/NLPUI.fxml";
22:         FileInputStream fxmlStream = new FileInputStream(fxmlPath);
23:         //Parent root = FXMLLoader.load(getClass().getResource("src/
net/penguincoders/Sentiment/SentimentUI.fxml"));
24:         //Create the pane and load all details
25:         Pane root = (Pane) loader.load(fxmlStream);
26:         //Create the scene
27:         Scene scene = new Scene(root);
28:         //Set the scene to the stage
29:         stage.setScene(scene);
30:         stage.setTitle("Natural Language Processing");
31:         stage.show();
32:     }
33:
34:     public static void main(String[] args)
35:     {
36:         launch(args);
37:         //System.out.println("Done executing.");
38:     }
39: }
```



```
1: package net.penguincoders;
2:
3: import java.util.Scanner;
4:
5: public class SimpleNLP
6: {
7:     public static void main(String args[])
8:     {
9:         Scanner ob = new Scanner(System.in);
10:        System.out.println("Enter text for analysis: ");
11:        String s = ob.nextLine();
12:        NLP.initSentiment();
13:        System.out.println("Sentiment: "+NLP.findSentiment(s));
14:        NLP.initNER();
15:        NLP.namedEntityRecognition(s);
16:        ob.close();
17:    }
18: }
```

```

1: package net.penguincoders.NER;
2:
3: import java.io.IOException;
4: import java.net.URL;
5: import java.util.ResourceBundle;
6:
7: import edu.stanford.nlp.pipeline.CoreDocument;
8: import edu.stanford.nlp.pipeline.CoreEntityMention;
9: import javafx.event.ActionEvent;
10: import javafx.fxml.FXML;
11: import javafx.fxml.FXMLLoader;
12: import javafx.fxml.Initializable;
13: import javafx.scene.Node;
14: import javafx.scene.Parent;
15: import javafx.scene.Scene;
16: import javafx.scene.control.Button;
17: import javafx.scene.control.TextArea;
18: import javafx.scene.layout.BorderPane;
19: import javafx.stage.Stage;
20: import net.penguincoders.NLP;
21:
22: public class NERUIController implements Initializable
23: {
24:     static
25:     {
26:         NLP.initNER();
27:     }
28:
29:     @FXML
30:     public TextArea inputText;
31:
32:     @FXML
33:     public Button extractKeyword;
34:
35:     @FXML
36:     public BorderPane outputPane;
37:
38:     @FXML
39:     public void goToOutputScreen(ActionEvent event) throws IOException
40:     {
41:         ((Node)event.getSource()).getScene().getWindow().hide();
42:         FXMLLoader loader = new FXMLLoader(getClass().getResource("NEROutput
43: .fxml"));
44:         Parent root = (Parent)loader.load();
45:         NEROutputController neroc = loader.getController();
46:         //neroc.setPersonLabel("Hello Person 1");
47:         CoreDocument doc = NLP.namedEntityRecognition(inputText.getText());
48:         String person="",organization="",location="";
49:         for (CoreEntityMention em : doc.entityMentions())
50:         {
51:             if(em.entityType().equalsIgnoreCase("PERSON"))
52:                 person += em.text()+"\n";
53:             if(em.entityType().equalsIgnoreCase("ORGANIZATION"))
54:                 organization+=em.text()+"\n";
55:             if(em.entityType().equalsIgnoreCase("LOCATION") || em.entity
56: Type().equalsIgnoreCase("COUNTRY") || em.entityType().equalsIgnoreCase("CITY"))
57:                 location+=em.text()+"\n";
58:         }
59:         neroc.setPersonLabel(person);
60:         neroc.setLocationLabel(location);
61:         neroc.setOrganizationLabel(organization);
62:         Stage output = new Stage();
63:         output.setTitle("Named Entity Recognition");
64:         output.setScene(new Scene(root));
65:         output.show();
66:     }
67:
68:     @FXML
69:     public void gotToPreviousScreen(ActionEvent event)throws Exception

```

```
67:         {
68:             ((Node)event.getSource()).getScene().getWindow().hide();
69:             FXMLLoader loader = new FXMLLoader(getClass().getResource("../NLPUI.
fxml"));
70:             Parent root = (Parent)loader.load();
71:             Stage output = new Stage();
72:             output.setTitle("Natural Language Processing");
73:             output.setScene(new Scene(root));
74:             output.show();
75:         }
76:         @Override
77:         public void initialize(URL arg0, ResourceBundle arg1) {
78:             // TODO Auto-generated method stub
79:
80:         }
81:
82: }
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.geometry.*?>
5: <?import javafx.scene.control.*?>
6: <?import java.lang.*?>
7: <?import javafx.scene.layout.*?>
8: <?import javafx.scene.layout.BorderPane?>
9:
10: <Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="net.penguincoders.NER.NERUIController">
11:     <children>
12:         <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0" prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
13:             <font>
14:                 <Font size="20.0" />
15:             </font></TextArea>
16:         <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text="Enter Text For Keyword Extraction" textFill="#176fc3">
17:             <font>
18:                 <Font size="24.0" />
19:             </font>
20:         </Label>
21:         <Button fx:id="extractKeyword" layoutX="255.0" layoutY="454.0" mnemonicParsing="false" onAction="#goToOutputScreen" prefHeight="48.0" prefWidth="232.0" text="Go" textFill="#0c74e4">
22:             <font>
23:                 <Font size="24.0" />
24:             </font>
25:         </Button>
26:         <Button layoutX="651.0" layoutY="538.0" mnemonicParsing="false" onAction="#goToPreviousScreen" prefHeight="37.0" prefWidth="89.0" text="Back" textFill="#1d53b2">
27:             <font>
28:                 <Font size="20.0" />
29:             </font>
30:         </Button>
31:     </children>
32: </Pane>
```

```
1: package net.penguincoders.NER;
2:
3: import java.net.URL;
4: import java.util.ResourceBundle;
5:
6: import javafx.event.ActionEvent;
7: import javafx.fxml.FXML;
8: import javafx.fxml.FXMLLoader;
9: import javafx.fxml.Initializable;
10: import javafx.scene.Node;
11: import javafx.scene.Parent;
12: import javafx.scene.Scene;
13: import javafx.scene.control.Button;
14: import javafx.scene.control.Label;
15: import javafx.scene.layout.BorderPane;
16: import javafx.stage.Stage;
17:
18: public class NEROutputController implements Initializable
19: {
20:
21:     @FXML
22:     Label personLabel;
23:
24:     @FXML
25:     Label locationLabel;
26:
27:     @FXML
28:     Label organizationLabel;
29:
30:     @FXML
31:     Button backButton;
32:
33:     public void setPersonLabel(String input)
34:     {
35:         personLabel.setText(input);
36:     }
37:
38:     public void setLocationLabel(String input)
39:     {
40:         locationLabel.setText(input);
41:     }
42:
43:     public void setOrganizationLabel(String input)
44:     {
45:         organizationLabel.setText(input);
46:     }
47:
48:     @FXML
49:     public void gotToPreviousScreen(ActionEvent event)throws Exception
50:     {
51:         ((Node)event.getSource()).getScene().getWindow().hide();
52:         FXMLLoader loader = new FXMLLoader(getClass().getResource("NERUI.fxml
53: l"));
54:         Parent root = (Parent)loader.load();
55:         Stage output = new Stage();
56:         output.setTitle("Named Entity Recognition");
57:         output.setScene(new Scene(root));
58:         output.show();
59:     }
60:     @Override
61:     public void initialize(URL arg0, ResourceBundle arg1) {
62:         // TODO Auto-generated method stub
63:     }
64:
65: }
```

```
1: package net.penguincoders.NER;
2:
3: import java.io.FileInputStream;
4:
5: import javafx.application.Application;
6: import javafx.fxml.FXMLLoader;
7: import javafx.scene.Scene;
8: import javafx.scene.layout.Pane;
9: import javafx.stage.Stage;
10:
11: public class NERUI extends Application
12: {
13:
14:     @Override
15:     public void start(Stage stage) throws Exception
16:     {
17:         //Create the FXMLLoader
18:         FXMLLoader loader = new FXMLLoader();
19:         //NERUIController controller = new NERUIController();
20:         //loader.setController(controller);
21:         //Path to FXML File
22:         String fxmlPath = "src/net/penguincoders/NER/NERUI.fxml";
23:         FileInputStream fxmlStream = new FileInputStream(fxmlPath);
24:         //Create the pane and load all details
25:         Pane root = (Pane) loader.load(fxmlStream);
26:         //Create the scene
27:         Scene scene = new Scene(root);
28:         //Set the scene to the stage
29:         stage.setScene(scene);
30:         stage.setTitle("Named Entity Recognition");
31:         stage.show();
32:     }
33:
34:     public static void main(String[] args)
35:     {
36:         launch(args);
37:         //System.out.println("Done executing.");
38:     }
39:
40: }
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.scene.control.*?>
5: <?import java.lang.*?>
6: <?import javafx.scene.layout.*?>
7: <?import javafx.scene.layout.BorderPane?>
8:
9: <BorderPane xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx
x:controller="net.penguincoders.NER.NEROutputController">
10:     <right>
11:         <Pane prefHeight="600.0" prefWidth="800.0" BorderPane.alignment="CENTER">
12:             <children>
13:                 <Label layoutX="253.0" layoutY="60.0" prefHeight="61.0" prefWidth="294.0
" text="Extracted Keywords" textAlignment="CENTER" textFill="#187bb8">
14:                     <font>
15:                         <Font size="28.0" />
16:                     </font>
17:                 </Label>
18:                 <Label layoutX="61.0" layoutY="177.0" prefHeight="51.0" prefWidth="87.0"
text="Person" textFill="#0da4d9">
19:                     <font>
20:                         <Font size="24.0" />
21:                     </font>
22:                 </Label>
23:                 <Label layoutX="308.0" layoutY="177.0" prefHeight="51.0" prefWidth="161.
0" text="Organization" textFill="#0da4d9">
24:                     <font>
25:                         <Font size="24.0" />
26:                     </font>
27:                 </Label>
28:                 <Label layoutX="615.0" layoutY="177.0" prefHeight="51.0" prefWidth="111.
0" text="Location" textFill="#0da4d9">
29:                     <font>
30:                         <Font size="24.0" />
31:                     </font>
32:                 </Label>
33:                 <Separator layoutX="230.0" layoutY="228.0" orientation="VERTICAL" prefHe
ight="240.0" prefWidth="0.0" />
34:                 <Separator layoutX="547.0" layoutY="228.0" orientation="VERTICAL" prefHe
ight="240.0" prefWidth="0.0" />
35:                 <Label fx:id="personLabel" layoutX="48.0" layoutY="239.0" prefHeight="31
9.0" prefWidth="165.0" wrapText="true" />
36:                 <Label fx:id="organizationLabel" layoutX="287.0" layoutY="245.0" prefHei
ght="319.0" prefWidth="250.0" wrapText="true" />
37:                 <Label fx:id="locationLabel" layoutX="596.0" layoutY="245.0" prefHeight=
"319.0" prefWidth="186.0" wrapText="true" />
38:                 <Button fx:id="backButton" layoutX="367.0" layoutY="545.0" mnemonicParsi
ng="false" onAction="#gotToPreviousScreen" prefHeight="35.0" prefWidth="100.0" text="Back">
39:                     <font>
40:                         <Font size="19.0" />
41:                     </font>
42:                 </Button>
43:             </children>
44:         </Pane>
45:     </right>
46: </BorderPane>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.geometry.*?>
5: <?import javafx.scene.control.*?>
6: <?import java.lang.*?>
7: <?import javafx.scene.layout.*?>
8: <?import javafx.scene.layout.BorderPane?>
9:
10: <Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="net.penguincoders.Sentiment.SentimentUIController">
11:     <children>
12:         <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0" prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
13:             <font>
14:                 <Font size="20.0" />
15:             </font></TextArea>
16:         <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text="Enter Text For Sentiment Analysis" textFill="#176fc3">
17:             <font>
18:                 <Font size="24.0" />
19:             </font>
20:         </Label>
21:         <Button fx:id="calculateScore" layoutX="50.0" layoutY="483.0" mnemonicParsing="false" onAction="#printLabelOutput" prefHeight="48.0" prefWidth="232.0" text="Calculate Score" textFill="#2419bc">
22:             <font>
23:                 <Font size="24.0" />
24:             </font>
25:         </Button>
26:         <Label fx:id="scoreLabel" layoutX="345.0" layoutY="483.0" prefHeight="48.0" prefWidth="403.0">
27:             <font>
28:                 <Font size="20.0" />
29:             </font></Label>
30:         <Button layoutX="632.0" layoutY="551.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="25.0" prefWidth="109.0" text="Back" textFill="#1d44c3">
31:             <font>
32:                 <Font size="19.0" />
33:             </font>
34:         </Button>
35:     </children>
36: </Pane>
```



```
1: package net.penguincoders.Sentiment;
2:
3: import java.io.FileInputStream;
4:
5: import javafx.application.Application;
6: import javafx.fxml.FXMLLoader;
7: import javafx.scene.Parent;
8: import javafx.scene.Scene;
9: import javafx.scene.layout.Pane;
10: import javafx.stage.Stage;
11:
12: public class SentimentUI extends Application
13: {
14:     @Override
15:     public void start(Stage stage) throws Exception
16:     {
17:         //Create the FXMLLoader
18:         FXMLLoader loader = new FXMLLoader();
19:         //SentimentUIController controller = new SentimentUIController();
20:         //loader.setController(controller);
21:         //Path to FXML File
22:         String fxmlPath = "src/net/penguincoders/Sentiment/SentimentUI.fxml"
;
23:         FileInputStream fxmlStream = new FileInputStream(fxmlPath);
24:         //Parent root = FXMLLoader.load(getClass().getResource("src/net/peng
uincoders/Sentiment/SentimentUI.fxml"));
25:         //Create the pane and load all details
26:         Pane root = (Pane) loader.load(fxmlStream);
27:         //Create the scene
28:         Scene scene = new Scene(root);
29:         //Set the scene to the stage
30:         stage.setScene(scene);
31:         stage.setTitle("Sentiment Analysis");
32:         stage.show();
33:     }
34:
35:     public static void main(String[] args)
36:     {
37:         launch(args);
38:         //System.out.println("Done executing.");
39:     }
40: }
```

```

1: package net.penguincoders.Sentiment;
2:
3: import java.net.URL;
4: import java.util.ResourceBundle;
5:
6: import javafx.event.ActionEvent;
7: import javafx.fxml.FXML;
8: import javafx.fxml.FXMLLoader;
9: import javafx.scene.Node;
10: import javafx.scene.Parent;
11: import javafx.scene.Scene;
12: import javafx.scene.control.Button;
13: import javafx.scene.control.Label;
14: import javafx.scene.control.TextArea;
15: import javafx.stage.Stage;
16: import net.penguincoders.NLP;
17: import javafx.fxml.Initializable;
18:
19: public class SentimentUIController implements Initializable
20: {
21:     static
22:     {
23:         NLP.initSentiment();
24:     }
25:     @FXML
26:     public TextArea inputText;
27:     @FXML
28:     public Button calculateScore;
29:     @FXML
30:     public Label scoreLabel;
31:     @FXML
32:     public URL location;
33:     @FXML
34:     public ResourceBundle resources;
35:
36:     // Add a public no-args constructor
37:     public SentimentUIController() {
38:     }
39:     @FXML
40:     public void printLabelOutput()
41:     {
42:         String input = inputText.getText();
43:         String score = NLP.findSentiment(input);
44:         scoreLabel.setText("Sentiment: "+score);
45:     }
46:
47:     @FXML
48:     public void gotToPreviousScreen(ActionEvent event)throws Exception
49:     {
50:         ((Node)event.getSource()).getScene().getWindow().hide();
51:         FXMLLoader loader = new FXMLLoader(getClass().getResource("../NLPUI.
fxml"));
52:         Parent root = (Parent)loader.load();
53:         Stage output = new Stage();
54:         output.setTitle("Natural Language Processing");
55:         output.setScene(new Scene(root));
56:         output.show();
57:     }
58:     @Override
59:     public void initialize(URL arg0, ResourceBundle arg1) {
60:         // TODO Auto-generated method stub
61:     }
62: }

```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <?import javafx.scene.text.*?>
4: <?import javafx.scene.control.*?>
5: <?import java.lang.*?>
6: <?import javafx.scene.layout.*?>
7: <?import javafx.scene.layout.AnchorPane?>
8:
9:
10: <Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="net.penguincoders.NLPUIController">
11:     <children>
12:         <Label layoutX="161.0" layoutY="83.0" prefHeight="60.0" prefWidth="478.0" text="Natural Language Processing" textFill="#1273c3">
13:             <font>
14:                 <Font size="32.0" />
15:             </font>
16:         </Label>
17:         <Label layoutX="54.0" layoutY="274.0" prefHeight="53.0" prefWidth="271.0" text="Sentiment Analysis" textFill="#0d82bc">
18:             <font>
19:                 <Font size="25.0" />
20:             </font>
21:         </Label>
22:         <Label layoutX="382.0" layoutY="274.0" prefHeight="53.0" prefWidth="370.0" text="Named Entity Recognition" textFill="#0d82bc">
23:             <font>
24:                 <Font size="25.0" />
25:             </font>
26:         </Label>
27:         <Button fx:id="sentimentAnalysisButton" layoutX="117.0" layoutY="372.0" mnemonicParsing="false" onAction="#sentimentAnalysis" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
28:             <font>
29:                 <Font size="20.0" />
30:             </font>
31:         </Button>
32:         <Button fx:id="NERButton" layoutX="523.0" layoutY="372.0" mnemonicParsing="false" onAction="#NER" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
33:             <font>
34:                 <Font size="20.0" />
35:             </font>
36:         </Button>
37:     </children>
38: </Pane>
```

```
1: package net.penguincoders;
2:
3: import edu.stanford.nlp.ling.CoreAnnotations;
4: import edu.stanford.nlp.neural.rnn.RNNCoreAnnotations;
5: import edu.stanford.nlp.pipeline.Annotation;
6: import edu.stanford.nlp.pipeline.CoreDocument;
7: import edu.stanford.nlp.pipeline.CoreEntityMention;
8: import edu.stanford.nlp.pipeline.StanfordCoreNLP;
9: import edu.stanford.nlp.sentiment.SentimentCoreAnnotations.SentimentAnnotatedTree;
10: import edu.stanford.nlp.trees.Tree;
11: import edu.stanford.nlp.util.CoreMap;
12:
13: public class NLP
14: {
15:     static StanfordCoreNLP pipeline;
16:
17:     public static void initSentiment()
18:     {
19:         pipeline = new StanfordCoreNLP("net/penguincoders/Sentiment/Sentimen
t.properties");
20:     }
21:
22:     public static void initNER()
23:     {
24:         pipeline = new StanfordCoreNLP("net/penguincoders/NER/NER.properties
");
25:     }
26:     public static String findSentiment(String input)
27:     {
28:         int mainSentiment = 0;
29:         if (input != null && input.length() > 0)
30:         {
31:             int longest = 0;
32:             Annotation annotation = pipeline.process(input);
33:             for (CoreMap sentence : annotation.get(CoreAnnotations.Sente
ncesAnnotation.class))
34:             {
35:                 Tree tree = sentence.get(SentimentAnnotatedTree.clas
s);
36:                 int sentiment = RNNCoreAnnotations.getPredictedClass
(tree);
37:                 String partText = sentence.toString();
38:                 if (partText.length() > longest)
39:                 {
40:                     mainSentiment = sentiment;
41:                     longest = partText.length();
42:                 }
43:             }
44:         }
45:         String sentiment[] = {"Very Negative", "Negative", "Neutral", "Posit
ive", "Very Positive"};
46:         return sentiment[mainSentiment];
47:     }
48:
49:     public static CoreDocument namedEntityRecognition(String input)
50:     {
51:         CoreDocument doc = new CoreDocument(input);
52:         pipeline.annotate(doc);
53:         for (CoreEntityMention em : doc.entityMentions())
54:         {
55:             System.out.println("\t"+em.text()+"\t"+em.entityType()+"\t"+em
.entityTypeConfidences());
56:         }
57:
58:         return doc;
59:     }
60: }
```