```xml
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<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">
   <children>
      <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0" prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
         <font>
            <Font size="20.0" />
         </font></TextArea>
      <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text="Enter Text For Keyword Extraction" textFill="#176fc3">
         <font>
            <Font size="24.0" />
         </font>
      </Label>
      <Button fx:id="extractKeyword" layoutX="255.0" layoutY="454.0" mnemonicParsing="false" onAction="#goToOutputScreen" prefHeight="48.0" prefWidth="232.0" text="Go" textFill="#0c74e4">
         <font>
            <Font size="24.0" />
         </font>
      </Button>
      <Button layoutX="651.0" layoutY="538.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="37.0" prefWidth="89.0" text="Back" textFill="#1d53b2">
         <font>
            <Font size="20.0" />
         </font>
      </Button>
   </children>
</Pane>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<BorderPane xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="net.penguincoders.NER.NEROutputController">
   <right>
      <Pane prefHeight="600.0" prefWidth="800.0" BorderPane.alignment="CENTER">
         <children>
            <Label layoutX="253.0" layoutY="60.0" prefHeight="61.0" prefWidth="294.0" text="Extracted Keywords" textAlignment="CENTER" textFill="#187bb8">
               <font>
                  <Font size="28.0" />
               </font>
            </Label>
            <Label layoutX="61.0" layoutY="177.0" prefHeight="51.0" prefWidth="87.0" text="Person" textFill="#0da4d9">
               <font>
                  <Font size="24.0" />
               </font>
            </Label>
            <Label layoutX="308.0" layoutY="177.0" prefHeight="51.0" prefWidth="161.0" text="Organization" textFill="#0da4d9">
               <font>
                  <Font size="24.0" />
               </font>
            </Label>
            <Label layoutX="615.0" layoutY="177.0" prefHeight="51.0" prefWidth="111.0" text="Location" textFill="#0da4d9">
               <font>
                  <Font size="24.0" />
               </font>
            </Label>
            <Separator layoutX="230.0" layoutY="228.0" orientation="VERTICAL" prefHeight="240.0" prefWidth="0.0" />
            <Separator layoutX="547.0" layoutY="228.0" orientation="VERTICAL" prefHeight="240.0" prefWidth="0.0" />
            <Label fx:id="personLabel" layoutX="48.0" layoutY="239.0" prefHeight="319.0" prefWidth="165.0" wrapText="true" />
            <Label fx:id="organizationLabel" layoutX="287.0" layoutY="245.0" prefHeight="319.0" prefWidth="250.0" wrapText="true" />
            <Label fx:id="locationLabel" layoutX="596.0" layoutY="245.0" prefHeight="319.0" prefWidth="186.0" wrapText="true" />
            <Button fx:id="backButton" layoutX="367.0" layoutY="545.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="35.0" prefWidth="100.0" text="Back">
               <font>
                  <Font size="19.0" />
               </font>
            </Button>
         </children>
      </Pane>
   </right>
</BorderPane>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<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">
   <children>
      <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0" prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
         <font>
            <Font size="20.0" />
         </font></TextArea>
      <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text="Enter Text For Sentiment Analysis" textFill="#176fc3">
         <font>
            <Font size="24.0" />
         </font>
      </Label>
      <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">
         <font>
            <Font size="24.0" />
         </font>
      </Button>
      <Label fx:id="scoreLabel" layoutX="345.0" layoutY="483.0" prefHeight="48.0" prefWidth="403.0">
         <font>
            <Font size="20.0" />
         </font></Label>
      <Button layoutX="632.0" layoutY="551.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="25.0" prefWidth="109.0" text="Back" textFill="#1d44c3">
         <font>
            <Font size="19.0" />
         </font>
      </Button>
   </children>
</Pane>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.AnchorPane?>


<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">
   <children>
      <Label layoutX="161.0" layoutY="83.0" prefHeight="60.0" prefWidth="478.0" text="Natural Language Processing" textFill="#1273c3">
         <font>
            <Font size="32.0" />
         </font>
      </Label>
      <Label layoutX="54.0" layoutY="274.0" prefHeight="53.0" prefWidth="271.0" text="Sentiment Analysis" textFill="#0d82bc">
         <font>
            <Font size="25.0" />
         </font>
      </Label>
      <Label layoutX="382.0" layoutY="274.0" prefHeight="53.0" prefWidth="370.0" text="Named Entity Recognition" textFill="#0d82bc">
         <font>
            <Font size="25.0" />
         </font>
      </Label>
      <Button fx:id="sentimentAnalysisButton" layoutX="117.0" layoutY="372.0" mnemonicParsing="false" onAction="#sentimentAnalysis" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
         <font>
            <Font size="20.0" />
         </font>
      </Button>
      <Button fx:id="NERButton" layoutX="523.0" layoutY="372.0" mnemonicParsing="false" onAction="#NER" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
         <font>
            <Font size="20.0" />
         </font>
      </Button>
   </children>
</Pane>
```

```java
package net.penguincoders;

import java.util.ArrayList;
import java.util.List;

import twitter4j.Query;
import twitter4j.QueryResult;
import twitter4j.Status;
import twitter4j.Twitter;
import twitter4j.TwitterException;
import twitter4j.TwitterFactory;

public class TweetManager
{
        public static ArrayList<String> getTweets(String topic)
        {
                int count = 0;
                Twitter twitter = new TwitterFactory().getInstance();
                ArrayList<String> tweetList = new ArrayList<String>();
                try
                {
                        Query query = new Query(topic);
                        QueryResult result;
                        do
                        {
                                result = twitter.search(query);
                                List<Status> tweets = result.getTweets();
                                count+=1;
                                for (Status tweet : tweets)
                                {
                                        tweetList.add(tweet.getText());
                                }
                        }
                        while (count != 1);
                }
                catch (TwitterException te)
                {
                        te.printStackTrace();
                        System.out.println("Failed to search tweets: " + te.getMessage());
                }
                return tweetList;
        }
}
```

```java
package net.penguincoders;

import java.util.ArrayList;

public class WhatToThink
{
        public static void main(String[] args)
        {
                String topic = "IPL 2019";
                int count = 0;
                ArrayList<String> tweets = TweetManager.getTweets(topic);
                System.out.println("tweets retrieved Done!!!");
                NLP.initSentiment();
                System.out.println(" init Done!!!");
                for(String tweet : tweets)
                {
                        System.out.println(tweet + " : " + NLP.findSentiment(tweet));
                        count++;
                }
                System.out.println("Done!!! Total = "+count);
        }
}
```

```java
package net.penguincoders;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;

public class NLPUIController
{
        @FXML
        public Button sentimentAnalysisButton;

        @FXML
        public Button NERButton;

        @FXML
        public void sentimentAnalysis(ActionEvent event)throws Exception
        {
                ((Node)event.getSource()).getScene().getWindow().hide();
                FXMLLoader loader = new FXMLLoader(getClass().getResource("Sentiment/SentimentUI.fxml"));
                Parent root = (Parent)loader.load();
                Stage output = new Stage();
                output.setTitle("Sentiment Analysis");
                output.setScene(new Scene(root));
                output.show();
        }

        public void NER(ActionEvent event)throws Exception
        {
                ((Node)event.getSource()).getScene().getWindow().hide();
                FXMLLoader loader = new FXMLLoader(getClass().getResource("NER/NERUI.fxml"));
                Parent root = (Parent)loader.load();
                Stage output = new Stage();
                output.setTitle("Named Entity Recognition");
                output.setScene(new Scene(root));
                output.show();
        }
}
```

```java
package net.penguincoders;

import java.io.FileInputStream;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

public class NLPUI extends Application
{
                @Override
                public void start(Stage stage) throws Exception
                {
                        //Create the FXMLLoader
                        FXMLLoader loader = new FXMLLoader();
                        //SentimentUIController controller = new SentimentUIController();
                        //loader.setController(controller);
                        //Path to FXML File
                        String fxmlPath = "src/net/penguincoders/NLPUI.fxml";
                        FileInputStream fxmlStream = new FileInputStream(fxmlPath);
                        //Parent root = FXMLLoader.load(getClass().getResource("src/net/penguincoders/Sentiment/SentimentUI.fxml"));
                        //Create the pane and load all details
                        Pane root = (Pane) loader.load(fxmlStream);
                        //Create the scene
                        Scene scene = new Scene(root);
                        //Set the scene to the stage
                        stage.setScene(scene);
                        stage.setTitle("Natural Language Processing");
                        stage.show();
                }

                public static void main(String[] args)
                {
                        launch(args);
                        //System.out.println("Done executing.");
                }
}
```

```java
package net.penguincoders;

import java.util.Scanner;

public class SimpleNLP
{
        public static void main(String args[])
        {
                Scanner ob = new Scanner(System.in);
                System.out.println("Enter text for analysis: ");
                String s = ob.nextLine();
                NLP.initSentiment();
                System.out.println("Sentiment: "+NLP.findSentiment(s));
                NLP.initNER();
                NLP.namedEntityRecognition(s);
                ob.close();
        }
}
```

```java
package net.penguincoders.NER;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

import edu.stanford.nlp.pipeline.CoreDocument;
import edu.stanford.nlp.pipeline.CoreEntityMention;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextArea;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;
import net.penguincoders.NLP;

public class NERUIController implements Initializable
{
        static
        {
                NLP.initNER();
        }

        @FXML
        public TextArea inputText;

        @FXML
        public Button extractKeyword;

        @FXML
        public BorderPane outputPane;
        @FXML
        public void goToOutputScreen(ActionEvent event) throws IOException
        {
                ((Node)event.getSource()).getScene().getWindow().hide();
                FXMLLoader loader = new FXMLLoader(getClass().getResource("NEROutput.fxml"));
                Parent root = (Parent)loader.load();
                NEROutputController neroc = loader.getController();
                //neroc.setPersonLabel("Hello Person 1");
                CoreDocument doc = NLP.namedEntityRecognition(inputText.getText());
                String person="",organization="",location="";
                for (CoreEntityMention em : doc.entityMentions())
                {
                        if(em.entityType().equalsIgnoreCase("PERSON"))
                                person += em.text()+"\n";
                        if(em.entityType().equalsIgnoreCase("ORGANIZATION"))
                                organization+=em.text()+"\n";
                        if(em.entityType().equalsIgnoreCase("LOCATION") || em.entityType().equalsIgnoreCase("COUNTRY") || em.entityType().equalsIgnoreCase("CITY"))
                                location+=em.text()+"\n";
                }
                neroc.setPersonLabel(person);
                neroc.setLocationLabel(location);
                neroc.setOrganizationLabel(organization);
                Stage output = new Stage();
                output.setTitle("Named Entity Recognition");
                output.setScene(new Scene(root));
                output.show();
    }

        @FXML
        public void gotToPreviousScreen(ActionEvent event)throws Exception
        {
                ((Node)event.getSource()).getScene().getWindow().hide();
                FXMLLoader loader = new FXMLLoader(getClass().getResource("../NLPUI.fxml"));
                Parent root = (Parent)loader.load();
                Stage output = new Stage();
                output.setTitle("Natural Language Processing");
                output.setScene(new Scene(root));
                output.show();
        }
        @Override
        public void initialize(URL arg0, ResourceBundle arg1) {
                // TODO Auto-generated method stub

        }

}
```

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<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">
   <children>
      <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0" prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
         <font>
            <Font size="20.0" />
         </font></TextArea>
      <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text="Enter Text For Keyword Extraction" textFill="#176fc3">
         <font>
            <Font size="24.0" />
         </font>
      </Label>
      <Button fx:id="extractKeyword" layoutX="255.0" layoutY="454.0" mnemonicParsing="false" onAction="#goToOutputScreen" prefHeight="48.0" prefWidth="232.0" text="Go" textFill="#0c74e4">
         <font>
            <Font size="24.0" />
         </font>
      </Button>
      <Button layoutX="651.0" layoutY="538.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="37.0" prefWidth="89.0" text="Back" textFill="#1d53b2">
         <font>
            <Font size="20.0" />
         </font>
      </Button>
   </children>
</Pane>
```

```java
package net.penguincoders.NER;

import java.net.URL;
import java.util.ResourceBundle;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;

public class NEROutputController implements Initializable
{

        @FXML
        Label personLabel;

        @FXML
        Label locationLabel;

        @FXML
        Label organizationLabel;

        @FXML
        Button backButton;

        public void setPersonLabel(String input)
        {
                personLabel.setText(input);
        }

        public void setLocationLabel(String input)
        {
                locationLabel.setText(input);
        }

        public void setOrganizationLabel(String input)
        {
                organizationLabel.setText(input);
        }

        @FXML
        public void gotToPreviousScreen(ActionEvent event)throws Exception
        {
                ((Node)event.getSource()).getScene().getWindow().hide();
                FXMLLoader loader = new FXMLLoader(getClass().getResource("NERUI.fxml"));
                Parent root = (Parent)loader.load();
                Stage output = new Stage();
                output.setTitle("Named Entity Recognition");
                output.setScene(new Scene(root));
                output.show();
        }
        @Override
        public void initialize(URL arg0, ResourceBundle arg1) {
                // TODO Auto-generated method stub

        }

}
```

```
package net.penguincoders.NER;

import java.io.FileInputStream;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

public class NERUI extends Application
{

        @Override
        public void start(Stage stage) throws Exception
        {
                //Create the FXMLLoader
                FXMLLoader loader = new FXMLLoader();
                //NERUIController controller = new NERUIController();
                //loader.setController(controller);
                //Path to FXML File
                String fxmlPath = "src/net/penguincoders/NER/NERUI.fxml";
                FileInputStream fxmlStream = new FileInputStream(fxmlPath);
                //Create the pane and load all details
                Pane root = (Pane) loader.load(fxmlStream);
                //Create the scene
                Scene scene = new Scene(root);
                //Set the scene to the stage
                stage.setScene(scene);
                stage.setTitle("Named Entity Recognition");
                stage.show();
        }

        public static void main(String[] args)
        {
                launch(args);
                //System.out.println("Done executing.");
        }

}
```

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<BorderPane xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="net.penguincoders.NER.NEROutputController">
   <right>
      <Pane prefHeight="600.0" prefWidth="800.0" BorderPane.alignment="CENTER">
         <children>
            <Label layoutX="253.0" layoutY="60.0" prefHeight="61.0" prefWidth="294.0" text="Extracted Keywords" textAlignment="CENTER" textFill="#187bb8">
               <font>
                  <Font size="28.0" />
               </font>
            </Label>
            <Label layoutX="61.0" layoutY="177.0" prefHeight="51.0" prefWidth="87.0" text="Person" textFill="#0da4d9">
               <font>
                  <Font size="24.0" />
               </font>
            </Label>
            <Label layoutX="308.0" layoutY="177.0" prefHeight="51.0" prefWidth="161.0" text="Organization" textFill="#0da4d9">
               <font>
                  <Font size="24.0" />
               </font>
            </Label>
            <Label layoutX="615.0" layoutY="177.0" prefHeight="51.0" prefWidth="111.0" text="Location" textFill="#0da4d9">
               <font>
                  <Font size="24.0" />
               </font>
            </Label>
            <Separator layoutX="230.0" layoutY="228.0" orientation="VERTICAL" prefHeight="240.0" prefWidth="0.0" />
            <Separator layoutX="547.0" layoutY="228.0" orientation="VERTICAL" prefHeight="240.0" prefWidth="0.0" />
            <Label fx:id="personLabel" layoutX="48.0" layoutY="239.0" prefHeight="319.0" prefWidth="165.0" wrapText="true" />
            <Label fx:id="organizationLabel" layoutX="287.0" layoutY="245.0" prefHeight="319.0" prefWidth="250.0" wrapText="true" />
            <Label fx:id="locationLabel" layoutX="596.0" layoutY="245.0" prefHeight="319.0" prefWidth="186.0" wrapText="true" />
            <Button fx:id="backButton" layoutX="367.0" layoutY="545.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="35.0" prefWidth="100.0" text="Back">
               <font>
                  <Font size="19.0" />
               </font>
            </Button>
         </children>
      </Pane>
   </right>
</BorderPane>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.BorderPane?>

<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">
   <children>
      <TextArea fx:id="inputText" layoutX="63.0" layoutY="126.0" prefHeight="270.0" prefWidth="616.0" promptText="Enter Your text here" wrapText="true">
         <font>
            <Font size="20.0" />
         </font></TextArea>
      <Label layoutX="63.0" layoutY="61.0" prefHeight="57.0" prefWidth="424.0" text="Enter Text For Sentiment Analysis" textFill="#176fc3">
         <font>
            <Font size="24.0" />
         </font>
      </Label>
      <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">
         <font>
            <Font size="24.0" />
         </font>
      </Button>
      <Label fx:id="scoreLabel" layoutX="345.0" layoutY="483.0" prefHeight="48.0" prefWidth="403.0">
         <font>
            <Font size="20.0" />
         </font></Label>
      <Button layoutX="632.0" layoutY="551.0" mnemonicParsing="false" onAction="#gotToPreviousScreen" prefHeight="25.0" prefWidth="109.0" text="Back" textFill="#1d44c3">
         <font>
            <Font size="19.0" />
         </font>
      </Button>
   </children>
</Pane>
```

```java
package net.penguincoders.Sentiment;

import java.io.FileInputStream;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

public class SentimentUI extends Application
{
        @Override
        public void start(Stage stage) throws Exception
        {
                //Create the FXMLLoader
                FXMLLoader loader = new FXMLLoader();
                //SentimentUIController controller = new SentimentUIController();
                //loader.setController(controller);
                //Path to FXML File
                String fxmlPath = "src/net/penguincoders/Sentiment/SentimentUI.fxml";
                FileInputStream fxmlStream = new FileInputStream(fxmlPath);
                //Parent root = FXMLLoader.load(getClass().getResource("src/net/penguincoders/Sentiment/SentimentUI.fxml"));
                //Create the pane and load all details
                Pane root = (Pane) loader.load(fxmlStream);
                //Create the scene
                Scene scene = new Scene(root);
                //Set the scene to the stage
                stage.setScene(scene);
                stage.setTitle("Sentiment Analysis");
                stage.show();
        }

        public static void main(String[] args)
        {
                launch(args);
                //System.out.println("Done executing.");
        }
}
```

```
package net.penguincoders.Sentiment;

import java.net.URL;
import java.util.ResourceBundle;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextArea;
import javafx.stage.Stage;
import net.penguincoders.NLP;
import javafx.fxml.Initializable;

public class SentimentUIController implements Initializable
{
        static
        {
                NLP.initSentiment();
        }
        @FXML
        public TextArea inputText;
        @FXML
        public Button calculateScore;
        @FXML
        public Label scoreLabel;
        @FXML
        public URL location;
        @FXML
        public ResourceBundle resources;

        // Add a public no-args constructor
        public SentimentUIController() {
        }
        @FXML
        public void printLabelOutput()
        {
                String input = inputText.getText();
                String score = NLP.findSentiment(input);
                scoreLabel.setText("Sentiment: "+score);
        }

        @FXML
        public void gotToPreviousScreen(ActionEvent event)throws Exception
        {
                ((Node)event.getSource()).getScene().getWindow().hide();
                FXMLLoader loader = new FXMLLoader(getClass().getResource("../NLPUI.fxml"));
                Parent root = (Parent)loader.load();
                Stage output = new Stage();
                output.setTitle("Natural Language Processing");
                output.setScene(new Scene(root));
                output.show();
        }
        @Override
        public void initialize(URL arg0, ResourceBundle arg1) {
                // TODO Auto-generated method stub
        }
}
```

```xml
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.AnchorPane?>


<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">
   <children>
      <Label layoutX="161.0" layoutY="83.0" prefHeight="60.0" prefWidth="478.0" text="Natural Language Processing" textFill="#1273c3">
         <font>
            <Font size="32.0" />
         </font>
      </Label>
      <Label layoutX="54.0" layoutY="274.0" prefHeight="53.0" prefWidth="271.0" text="Sentiment Analysis" textFill="#0d82bc">
         <font>
            <Font size="25.0" />
         </font>
      </Label>
      <Label layoutX="382.0" layoutY="274.0" prefHeight="53.0" prefWidth="370.0" text="Named Entity Recognition" textFill="#0d82bc">
         <font>
            <Font size="25.0" />
         </font>
      </Label>
      <Button fx:id="sentimentAnalysisButton" layoutX="117.0" layoutY="372.0" mnemonicParsing="false" onAction="#sentimentAnalysis" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
         <font>
            <Font size="20.0" />
         </font>
      </Button>
      <Button fx:id="NERButton" layoutX="523.0" layoutY="372.0" mnemonicParsing="false" onAction="#NER" prefHeight="39.0" prefWidth="89.0" text="Go" textFill="#1a35a3">
         <font>
            <Font size="20.0" />
         </font>
      </Button>
   </children>
</Pane>
```

```java
package net.penguincoders;

import edu.stanford.nlp.ling.CoreAnnotations;
import edu.stanford.nlp.neural.rnn.RNNCoreAnnotations;
import edu.stanford.nlp.pipeline.Annotation;
import edu.stanford.nlp.pipeline.CoreDocument;
import edu.stanford.nlp.pipeline.CoreEntityMention;
import edu.stanford.nlp.pipeline.StanfordCoreNLP;
import edu.stanford.nlp.sentiment.SentimentCoreAnnotations.SentimentAnnotatedTree;
import edu.stanford.nlp.trees.Tree;
import edu.stanford.nlp.util.CoreMap;

public class NLP
{
        static StanfordCoreNLP pipeline;

        public static void initSentiment()
        {
                pipeline = new StanfordCoreNLP("net/penguincoders/Sentiment/Sentiment.properties");
        }

        public static void initNER()
        {
                pipeline = new StanfordCoreNLP("net/penguincoders/NER/NER.properties");
        }
        public static String findSentiment(String input)
        {
                int mainSentiment = 0;
                if (input != null && input.length() > 0)
                {
                        int longest = 0;
                        Annotation annotation = pipeline.process(input);
                        for (CoreMap sentence : annotation.get(CoreAnnotations.SentencesAnnotation.class))
                        {
                                Tree tree = sentence.get(SentimentAnnotatedTree.class);
                                int sentiment = RNNCoreAnnotations.getPredictedClass(tree);
                                String partText = sentence.toString();
                                if (partText.length() > longest)
                                {
                                        mainSentiment = sentiment;
                                        longest = partText.length();
                                }
                        }
                }
                String sentiment[] = {"Very Negative", "Negative", "Neutral", "Positive", "Very Positive"};
                return sentiment[mainSentiment];
        }

        public static CoreDocument namedEntityRecognition(String input)
        {
                CoreDocument doc = new CoreDocument(input);
                pipeline.annotate(doc);
                for (CoreEntityMention em : doc.entityMentions())
                {
                      System.out.println("\t"+em.text()+"\t"+em.entityType()+"\t"+em.entityTypeConfidences());
                }

                return doc;
        }
}
```