ASSIGNMENT NO. B2

Problem Statement:

Write a web application using Scala/ Python/ Java /HTML5 to check the plagiarism in the given text paragraph written/ copied in the text box. Give software Modeling, Design, UML and Test cases for the same using Analysis Modeling (Static Modeling, Object Structuring, Dynamic Modeling)..

Learning Objectives:

1. To study how to create a web application..
2. TO learn to implement plagirism tool in web application.

Learning Outcomes:

Learnt about web application and able to plagiarised the codes or files.

Software and Hardware Requirements:

1. 64-bit operating System(Linux)
2. Apache Tomcat
3. JSP pages
4. Browser (Google Chrome 0.2.149 or higher/ any other equivalent)
5. Compiler (JSPC Compiler 2.0.2 or higher/ any other equivalent)
6. Modelio 3.6.1

Theory

Modelio

Modelio is an open source UML tool developed by Modeliosoft, based in Paris, France. It supports the UML2 and BPMN standards. Modelio supports UML2 Profiles for XSD, WSDL and BPEL SoaML for service modelling in distributed environments, and BPMN for business process modelling.

Web Applications

In computing, a web application or web app is a client-server software application in which the client (or user interface) runs in a web browser. Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility.

Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a mobile app. HTML5 introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.

Applications

Examples of browser applications are simple office software (word processors, online spreadsheets, and presentation tools), but can also include more advanced applications such as project management, computer-aided design, video editing and point-of-sale.

Plagiarism

Plagiarism is the "wrongful appropriation" and "stealing and publication" of another author's "language, thoughts, ideas, or expressions" and the representation of them as one's own original work.Plagiarism is considered academic dishonesty and a breach of journalistic ethics. It is subject to sanctions like penalties, suspension, and even expulsion. Plagiarism is not a crime per se but in academia and industry, it is a serious ethical offense, and cases of plagiarism can constitute copyright infringement. Thus, plagiarism and copyright infringement might overlap somewhat, but they are not equivalent concepts, and many types of plagiarism do not fall under the category of copyright infringement. Copyright infringement is defined by copyright law and may be adjudicated by courts. Plagiarism is not defined by law, but rather by institutions (including professional associations, educational institutions, and commercial entities, such as publishing companies) and punishment for plagiarism is not set forth by the legal system.

Requirements Modeling:

During the requirements modeling phase, a requirements model is developed in which the functional requirements of the system are defined in terms of actors and use cases. A narrative description of each use case is developed. User inputs and active participation are essential to this effort. If the requirements are not well understood, a throwaway prototype can be developed to help clarify the requirements

Analysis Modeling:

In the analysis modeling phase, static and dynamic models of the system are developed. The static model defines the structural relationships among problem domain classes. The classes and their relationships are depicted on class diagrams. Object-structuring criteria are used to determine which objects should be considered for the analysis model. A dynamic model is then developed in which the use cases from the requirements model are realized to show the objects that participate in each use case and how they interact with each other. Objects and their interactions are depicted on either communication diagrams or sequence diagrams. In the dynamic model, state-dependent objects are defined with statecharts. A statechart is a graphical representation of a finite state machine in the form of a hierarchical state transition diagram.

Design Modeling:

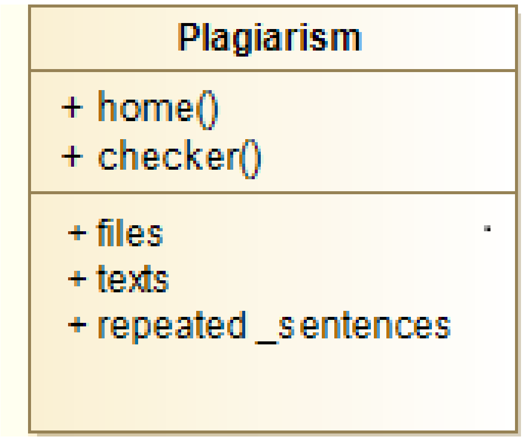
In the design modeling phase, the software architecture of the system is designed; that is, the analysis model is mapped to an operational environment.

The analysis model (which emphasizes the problem domain) is mapped to the design model (which emphasizes the solution domain). Subsystem structuring criteria are provided to structure the system into subsystems, which are considered as aggregate or composite objects. Special consideration is given to designing distributed subsystems as configurable components that communicate with each other using messages. Each subsystem is then designed. For sequential systems, the emphasis is on the object-oriented concepts of information hiding, classes, and inheritance. For the design of concurrent systems, such as real-time, client/server, and distributed applications, it is necessary to consider concurrent tasking concepts in addition to object-oriented concepts.

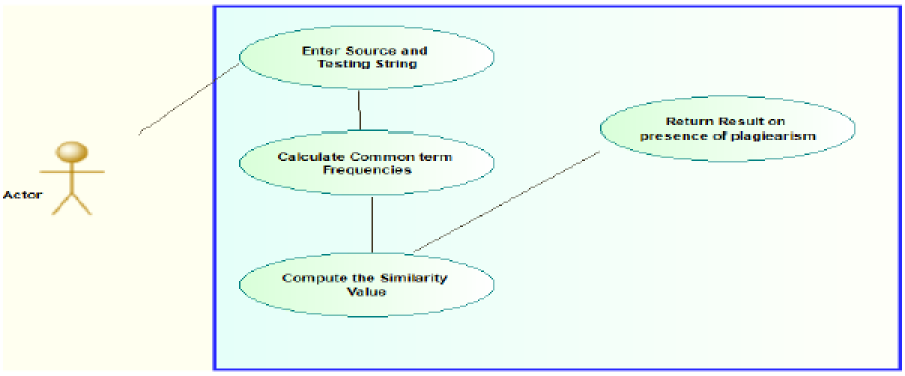
Mathematical Model

```
S = {s, e, I, o, f, DD, NDD, success, failure}
where,
s = {Initial state of system}
  = {apache tomcat started.}
e = End state of the system

I = {input of system}
I = (Text to be evaluated for plagiarism)

o = {Output of system}
o = It will show is entered text is plagiarised or not

DD = {Deterministic Data }
DD = Text in existing file

NDD = {Non-Deterministic Data }
NDD = What text user will enter in textbox

f = {check(),write()}
where,
check()={Compares the text given by user with the available text}
write()={Displays the plagiarism in percentage for each text}

Success = {Desired outcome generated }
success = Plagiarism in percentage is displayed.

Failure = {Desired outcome not generated }
failure = Error message is displayed.
```

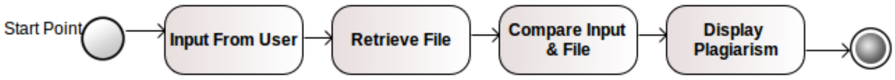
Class diagram



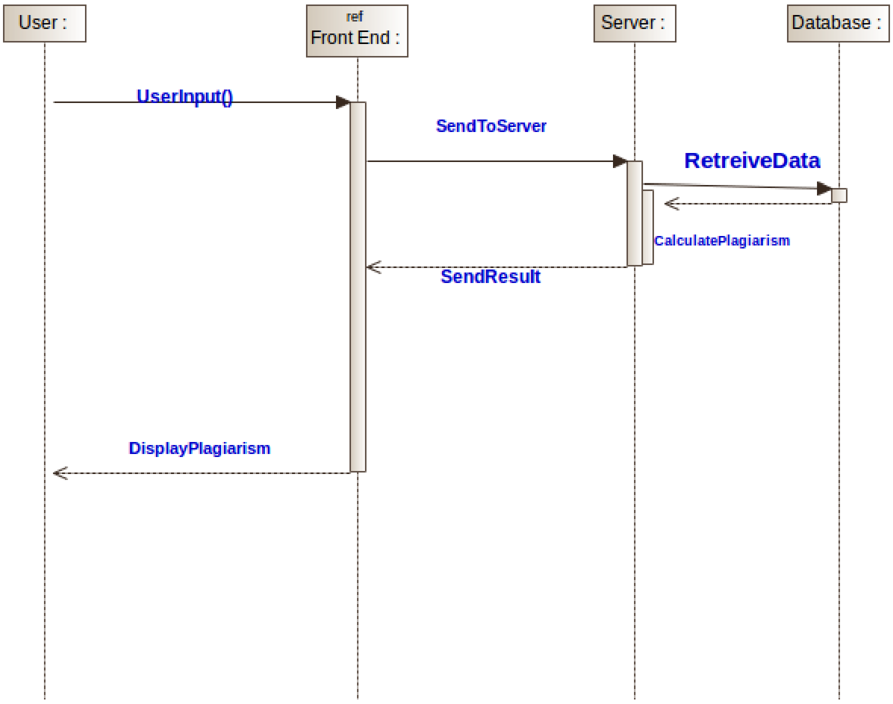
Use-case diagram


Actor
Enter Source and Testing String
Calculate Common term Frequencies
Compute the Similarity Value
Return Result on presence of plagiearism

State diagram



Sequence diagram



Algorithm

1. Start
2. Enter the texts for comparison.
3. Input the text to be checked for plagiarism.
4. Compare the input text with the available text line by line.
5. Display the plagiarism in percentage.
6. Stop

Positive Testing

Input	**Expected Output**	**Actual Output**
Correct path of both the files	Plagiarism displayed	Same as expected
Both files are same	100% Plagiarism	Same as expected

Negative Testing

Input	**Expected Output**	**Actual Output**
Incorrect path of file	Invalid path	Same as expected

Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. PRECONDITIONS:

1. Comparison between entered text and the existing file text.

White Box testing

White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

1. Input i.e text entered by user and the text in file.

Conclusion

We have successfully created web application to check plagiarism in the given text.