bona-fide

A Plagiarism Detection Software…

Our Team...

Aadarsh Sahoo (17CS30041) {Project Design & BackEnd}

Ayush Kumar (17CS10007) {Project Design & FrontEnd}

https://github.com/AadSah/bona-fide.git

The Need...

- The widespread use of computers and the advent of the Internet has made it easier to plagiarize the work of others.
- Most cases of plagiarism are found in academia, where documents are typically essays, reports, research papers, etc.
- Plagiarism can be found in virtually any field, including novels, scientific papers, art designs, and source code.
- Manual detection requires substantial effort and excellent memory, and is impractical.
- Software-assisted detection allows vast collections of documents to be compared to each other, making successful detection much more likely.

Some Statistics...

- 36% of undergraduates admit to “paraphrasing/copying few sentences from Internet source without footnoting it.”
  - 24% of graduate students self report doing the same
- 38% admit to “paraphrasing/copying few sentences from written source without footnoting it.”
  - 25% of graduate students self report doing the same
- 14% of students admit to “fabricating/falsifying a bibliography”
  - 7% of graduate students self report doing the same
- 7% self report copying materials “almost word for word from a written source without citation.”
  - 4% of graduate students self report doing the same
- 7% self report “turning in work done by another.”
  - 3% of graduate students self report doing the same

Source::https://www.plagiarism.org/article/plagiarism-facts-and-stats

Our Goal...

To promote and sustain the
‘Value’ of
‘Intellectual Property’ and
‘Originality in Idea and Expression’.

Development Environment...

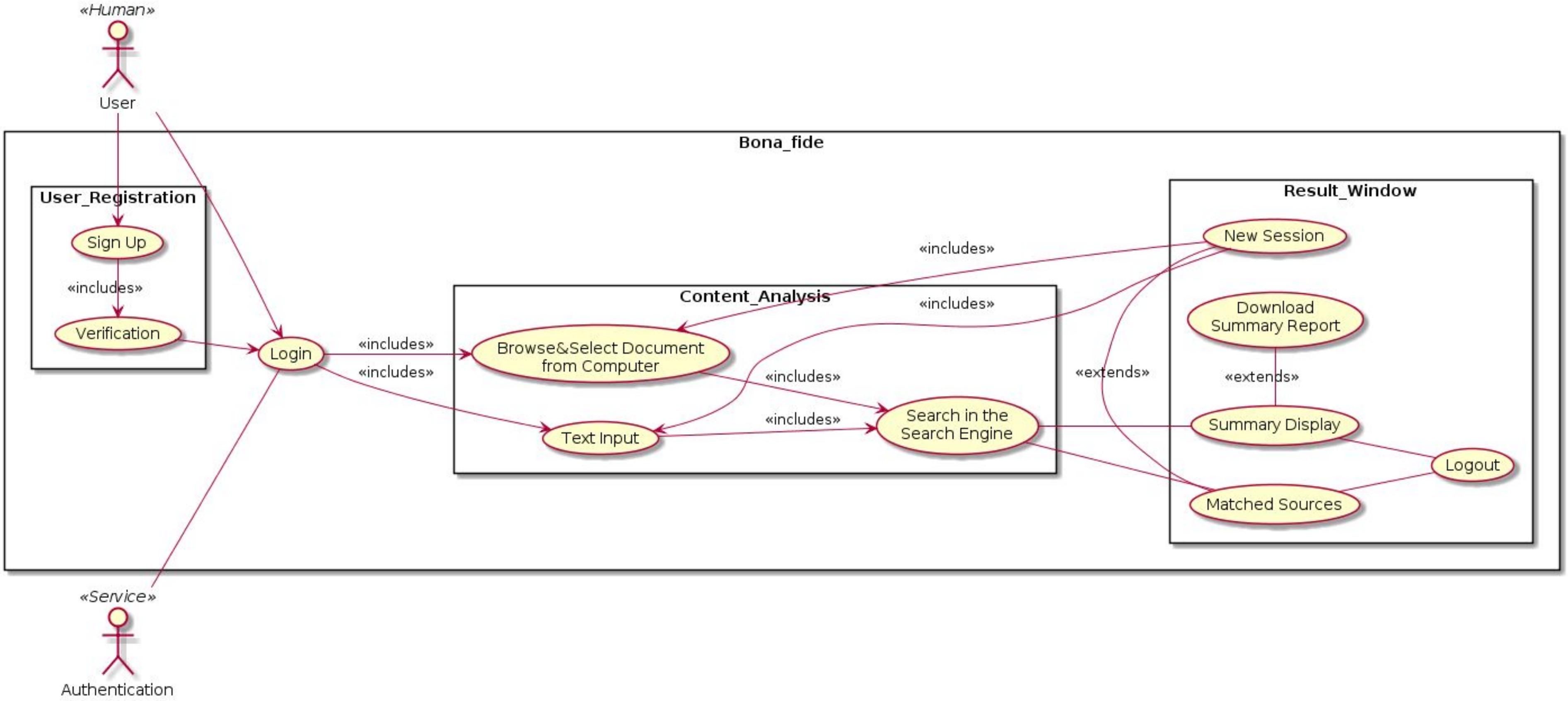
Most of the Code is in **Java (JDK8)**, **python3** is used for implementing some backend functionalities only:

- **FrontEnd Development:**
    - Developed using **Netbeans IDE 8.2**.
    - Toolkits: **Java Swing**, **Java AWT**(Abstract Window Toolkit)
- **Database:**
    - **MySQL** Oracle Corporation
    - Database is used for Storing User Login Credentials only

Development Environment...

- **BackEnd Development:**
  - **Fuzzywuzzy**: A python library which uses Levenshtein Distance to calculate the differences between sequences in a simple-to-use package, github: https://github.com/seatgeek/fuzzywuzzy
  - **Google API for python**: Used for performing the required searches over the Internet for Analyzing the User Content. website: https://developers.google.com/api-client-library/python/
  - **BeautifulSoup**: A python package used for parsing HTML and XML documents for Scraping web-data.
  - **Java.io**: Used for performing all the file handling operations for the software.

UML::Use-Case Diagram...



Overall Working (Plag-Detection Flow)...

- User uploads/writes the File to be Analyzed…
- The Content in the File is broken up into paragraphs/sentences and are made ready for the Search Operation…
- Each Line is Searched using the Google API…
  - Only the **Top URL** is obtained (for Quick Analysis Function)...
  - The **Top Three(3) URLs** are obtained (for Detailed Analysis Function)...
- The Web-Content from the respective URLs is Scrapped…
  - The User-File-Content is matched with the scraped data and the respective Levenshtein distance is noted...
- The Plagiarism Percentage is calculated accordingly with view of the total number of paragraphs/lines present in the user document...

Challenges...

- **Synonym Matching:** Changing some of the words to their corresponding synonyms to get themselves screened from the application software.(Implementing NLP should help)...
- **Non-Verbatim Plagiarism:** The user may rewrite, translate or otherwise redraft the content from the source and deceive the software system. This problem arises because Plagiarism detectors analyze the words, they don’t analyze the content i.e. it can’t see if you copied the idea or information even if you didn’t copy the words. So this may let go of some serious plagiarised content undetected.
- **Common Phrasing:** The document or content being analyzed is very likely to contain many common phrases in the English language, which may be reported by the software system as a match even though that might be just a coincidence.(Intelligent comparison system should help)…

Demo Time!...