1. **INTRODUCTION**

**1.1 BACKGROUND:**

File comparison system is a tool which will able to tell whether the data in two or more file is similar to what extent to each other , the data in the files may be different with each other though still containig the same meaning and that is the place where file comparing system plays a crucial role it will tell the similarity if it exist in their meaning even though they are syntaticaly different just like human who can infer the real meaning of the sentences written in the data files and tell the similarity.

The comparison of files is always been an area of research to find the similarities and difference between the files and it is keep growing till now, to make the comparison better and better and more accurate.

More precisely this system is known as STS(Semantic Textual Similarity) which means to find degree of similarity between two given sentences and that similarity means similarity based on the meaning of two given sentences.

**Semantic Textual Similarity (STS)** can be defined by a metric over a set of documents with the idea is to finding the semantic similarity between them.

Similarity between the documents is based on the direct and indirect relationships among them . These relationships can be measured and recognized by the presence of semantic relations among them.

**Classification of STS:** We can split out the ways of finding the semantic similarity into three categories

**1) Topological/Knowledge-based.**

**2)Statistical/Corpus Based.**

**3)String based.**

Among all of them Topological/Knowledge-based is considered in present popular system to compare the similarity between the two sentences, Because Topological methods, plays an important role to understand intended meaning of an ambiguous word, which is computationally very hard.

Semantic similarity plays an important role in NLP(natural language processing) and it is one of the fundamental taskes for many NLP applications and its related areas.

One of the popular comparing system that we had in ‘diff’ command in unix based system though there always been some hurdles in this area and one of them is to find the similarities and difference between the files based on their meaning where textual architecture can differ up to any extent for example considering the two sentences "men eats food” and “men eats bread” here both the sentences are similar in meaning as both of them are actually taking about the food consumption of human race but both of them are textually different and for a kindergarten child both are totally different as they don’t have that much understanding of these textual phrases.

Some of the other popular file comparison systems are :-

**1)AptDiff**

**2)DiffMerge**

**3)Diffuse**

**4)ExamDiff**

**5)KDiff3**

though they are currently popular but still facing the issues regarding accuracy in terms of text which have similar meaning but different textual appearance.

**1.2 OBJECTIVES**

1. To develop a system that will be able to compare two or more files data and tell whether they are

similar or and upto what extent they are similar,based on their meaning even after having different textual structure.

**2)** To develop a interactive web interface for the easy interaction of the user who wishes to compare the content of its data files and want to know how similar they are.

**3)** To apply machine learning approache to make project self sufficient to learn from various training data sets and from the future experience of its uses.

**4)** To apply the method of finding Semantic Textual Similarity between two sentences based on overlapping senses, which is one of the new techinque of deducing the Semantic Textual Similarity between two sentences as published in the research paper mentioned in refereces of this document.

**1.3 PURPOSE AND SCOPE**

**1.3.1 PURPOSE:** Since file comparison system which compare the files data based on their meaning rather than the textual structure needs a kind of system that would be able to learn itself from the past experience as like human, about the precision of deducing the similarity and deduce much better than the past model, for that we need something like machine learning or more precisely the subset of machine learning known as “NLP“ or “Natural language Processing”.

Moreover using NLP for the file comparison system would solve the problem of accuracy to a great extent.

Now as there are models which are nowdays using NLP as a tool for comparing the files datasets, more specifically to decide the Semantic Textual Similarity, and the one which is used in this project is using the concept of overlapping senses of words used in sentences that we want to compare.

**Why Overlapping Senses Method:** since in languages there are more than one meaning of a word which is totally depend upon the context in which it is used in the given sentence, so we can declare the two words in two different sentences as similar if they are using the same context and hence can be considered as equivalent, this is the approach that will lead us to determine whether the two documents are similar or not and if they are similar then upto what extent they are similar that would be given on the basis of the similarity score assigned to them.

**1.3.2 SCOPE:** Semantic Textual Similarity is itself a significant field for researchers in present and in near future and so does a utility which decide the similarity between any given sets of documents, since in this present scenario where areas such as quantum computing and other technical aspects are flourishing at a rapid pace and hence increasing the computation power of computers, there is a rapid development in demand for utilising this enhance computation power to deal with problems which are unchasable in past and one of them is making human languages explanatory to electronics gadgets more specifically to computers, so this comparing system can be a part of this set to help computer to decide the difference between the two given inputs in terms of human languages.

Moreover in the Field of Natural Language Processing it plays a significant role as developer for the efficient data sets to train various machine models.

It can be used in performing textual analysis of any social platform and of anything by comparing the level of similarity of sentence used in that analysis.

It can also plays an import role in document retrieval by using natural language processing modules and by training their machine learning models.

**2 SURVEY OF TECHNOLOGIES:**  This project can be developed in any language as here we uses Semantic Textual Similarity based on machine learning algorithms and since machine learning algorithms can be encoded in any language so does this project, so languages such as Java, C++ or python anyone can be used but here **python is the best fit for this project due to certain reasons:**

**1) Built In libraries for Machine learning:** Being an open source and platform independent python provides a greate variety of libraries for usual and complex both type of tasks, it has very effecient libraries regarding the Natural Language Processing and Machine learning entities which gives us liberty to use them rather than buiding each thing from scratch.

**2) Highly Object Oriented:** Python is one of the popular object oriented programming language in the recent past and at present also,which helps us in using the object oriented methods and concepts quite easily in this language.

**3) Faster rate of Development:** Being an open source and very popular language, python is flourishing like nothing else and which provides us facilites to nuture our code and modify it to the best level, which can be easily done in this language.

**4) Interpreted Than Compiled:** Since python is interpreted by its interpreter rather than compiled which makes it user friendly for the detection of errors in codes whether it would be a logical or syntax error, both can be easily rectified in this language.

**3 REQUIREMENTS AND ANALYSIS**

**3.1 PROBLEM DEFINITION:**

**why we need this:** Since in the advent of this modern era computational power is increasinig at a very high speed and which led us to solve the problem which are not addressed earlier and one of them is the understading of human language for computers and since computer is a bair bones of electronic circuits so its needs quite effort and new techniques to solve this problem.

Now since if we want to interact fully with machines as like humans then in this respect the area of Natural Language Processing is of great importance and in the absence of which it is merely impossible to communicate with computers as like humans.

As Natural Language Processing is the field where we deal with the isssue of processing the human languages for computers to make them capable of understanding our instructions in the form of our language rather than any machine coded instructions, here Semantic Textual Similarity plays an important role as it led machines to differentiate between tthe wo given instructions in the form of human language and also helps to deduce the similarity between them also, all these things creates a need to have a system that would be able to differentiate between the two given sentences and tell whether they are similar or not as this would finally led us to prepare good data sets for models of Natural language Processing to have a better training to the algorithms used in that.

**What it is:** File comparing system is basically a system that would tell us about the differences in the two documents provided based on their meaning rather than textual appearences which basically needs the use of Semantic Textual Similarity to calculate the similarity between the two given sentences.

**Problem Bifurcation:** Since this problem is totally dependent upon how effeciently and effectiveley the STS can be performed and the task of performing or deducing the Semantic Textual Similarity between two sentences can be divide into certain parts which are : Sentence Identification, Tokenization, Creation of Bag of words, Deduction of part of speech and Generation of STS score.

**3.2 REQUIREMENTS SPECIFICATION**

**3.2.1 OPERATIONAL REQUIREMENTS:**

**1) Sufficent Amount of Data Set:** Since finding the STS in this project is based on the phenomenon of Overlapped Senses, that means two words in a sentence is considered to be similar if they are carrying the same sense in the given two sentences, and for that machine learning algorithm is used in this project to train the developed model so that it can be able to find the senses of the words used in the sentences.

Now to train any machine learning model, sufficient amount of Data set is needed, which is also the case with this model, as more the data set is, more effective the prediction of the sense of word in a sentence would be. Therefore there is a need of good amount of Data set to train the model developed for finding the sense of the word used in a sentence.

**2) Regularly Used:** Again as is the case with any machine learning model, the more it get used the more better it would be, providing Sufficient amount of Data Set is not only capable of making any model successfull, but there is a need to use the derived model on a regular basis, as the more it is used the more it will be trained and as in English Language new words and their new senses are getting updated due the heavy use of this language by the current world, so there is a need to use our model regulary so that it can get trained appropriately as per the need of the world and present scenario.

**3.2.2 PROBLEMS:**

1. **New Researches:** As Semantic Textual Similarity is the open and flourishing area of research and new techniques are still emerging to find the better Semantic textual similarity between any two sentences, that means none of the techniques at present can be considered as the paromount technique of all time, and there is a huge posssibility of emerging of new ways to find the better Semantic Textual Similarity between any two sentences, so is the case with our used techinque to find the Semantic Textual Similarity using overlapping senses, it would be highly likely that there would be some method or technique in future that would surpassed the results of model used in this project.
2. **Not At All Full Proof:** Since a machine learning model is used in this project to asses the Semantic Textual Similarity between two sentences, but as its name suggest that it will remain a Machine **Learning** model during his whole life time, it would get better as it is used, but there is no certainity at all about the results or Semantic Textual Similarity deduced by the model for a pair of Sentences that it would be absolutely correct.

So, it is a good way to calculate the STS between any two sentences or documents but it is not at all Full Proof and no algorithm can be, at present.

**3.3 Planning and Scheduling:**

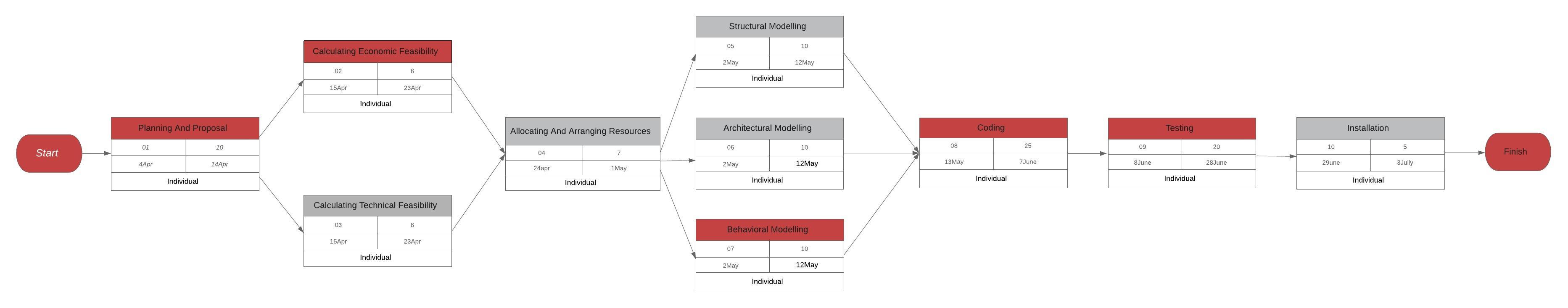
**Gantt Chart**

A Gantt chart is popular type of chart that illustrates a project schedule. Gantt Chart illustrates the start and finish dates of the terminal elements and summary elements of a project. Terminal element and summary comprise the work breakdown structure of the project.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **4Apr-30Apr** | **31Apr-9May** | **10May-12June** | **13June-12Jully** | **13Jully-18Jully** | **18Jully-23Jully** |
| **Develop project proposal** | 27 days |  |  |  |  |  |
| **Analysis** |  | 10 days |  |  |  |  |
| **Designing** |  |  | 30 days |  |  |  |
| **Coding** |  |  |  | 29days |  |  |
| **Unit Testing** |  |  |  |  | 5 days |  |
| **Implementation** |  |  |  |  |  | 5 days |

**Gantt Chart**

**Pert Chart**

****

**3.4 SOFTWARE AND HARDWARE REQUIREMENTS:**

**3.4.1 HARDWARE REQUIREMENTS:**

**Client Side**

|  |  |
| --- | --- |
| Processor | Dual Core or above |
| RAM | 1 GB |
| Disk space | 500 GB |
| Monitor | 15” |
| Others | Keyboard, mouse, Internet Connection |

**Server Side**

|  |  |
| --- | --- |
| Processor | I3 or above |
| RAM | 4 GB |
| Disk space | 500 GB |
| Monitor | 15” |
| Others | Keyboard, mouse, Internet Connection |

**3.4.2. Software Requirements:**

To develop this project there are certain software requirements that needs to be fulfilled and these are as follows:

**1) Anaconda Distribution 5.3.0 or higher :** This is needed to provide python version 3.6 or higher and other supporitive libraries built for machine learning and othere powerfull uses of python language.

**2) Jupyter Notebook or Jupyter Lab:** It is a web based user interface, which works as an IDE for the supportive kernels, and it is needed to prepare the notes and for trying dry code runs, moreover it is a full fleged utility to work interactively with codes.

**3) Visual Studio Code:** It is an Ide which is needed to help in creating effecient code files with proper extensions provided in it for python and other languages such as Html and javascript, and the whole project actual compilation would be performed here only in this project.

**4) Selenium Automated Testing Suite:** This is needed for performing the Automated testing of the project developed as a whole and as well as different units of the project. Moreover for the use of selenium there is also a need of corresponding web browser driver, which is needed to bind the selenium automated testing suite to the web browser that user want to use.

**5) Machine learning Libraries ‘Ntlk’ and ‘scikit learn’:** They are needed for performing the task of deducing the semantic textual similarity between the two sentences and to train the developed model for the algorithims used in this project.

**6) Web Browser:** This is needed to perform the testing procedure at the time of project development more specifically the elements of the web interface developed.

**7) Linux OS:** since any operating system can be used for the project development but the open source linux is quite better in terms of integrating the above mentioned softwares effeciently.

**8) Lucidchart:** This is a website which provides easy diagramming tool for the development UML Diagrams and other figures used in this projet at no cost.

**9) Libre office:** This is an open source office package which is needed for the development of documents used in this project.

**3.5 PRELIMINARY PRODUCT DESCRIPTION:**

File Comparing System is a project which aims to produce an utility cum web interactive application for determining the textual similarity for a given set of documents, it aims to avail multiple facilities to the user at one go, these function or facilities can be listed as follows:

Sentence splitting: First of all the documents given will be splitted into appropriate sentences which means not only the full stop will be treated as the teremination of sentence in a document but other punctutation marks and sentence symbols are also treated as stop words to decide the exact or more appropriate splitting of documents texts into sentences.

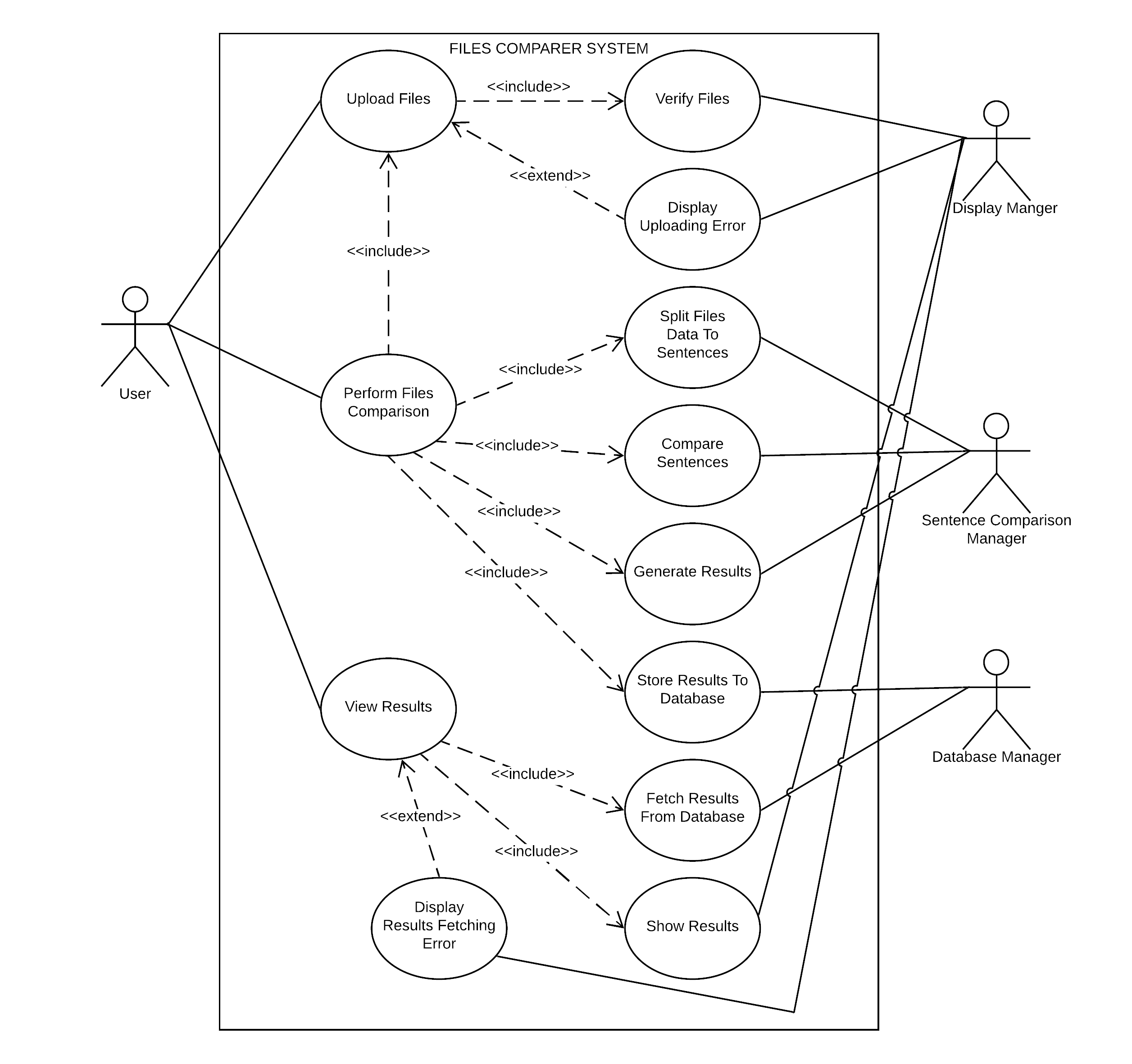
After this splitting, each sentence will be stored in a list which will contain all the sentences splitted for a particular document.

* **Tokenization:** Once the splitting is performed in a successfull manner , then these sentences are further splitted to tokens more generally words to create a bag of array so that various machine learning algorithms can be used over that data sets to find the sense of that token in a given sentence, this process will be then iterated to all the sentences in the document given to the utility.
* **Similarity Scores:** Once the process of tokenization has been completed then the process of applying machine learning algorithm to decide which sentence is similar to which one will proceed and that would be completed after alloting each sentence a similarity scores with respect to other sentence of documents.

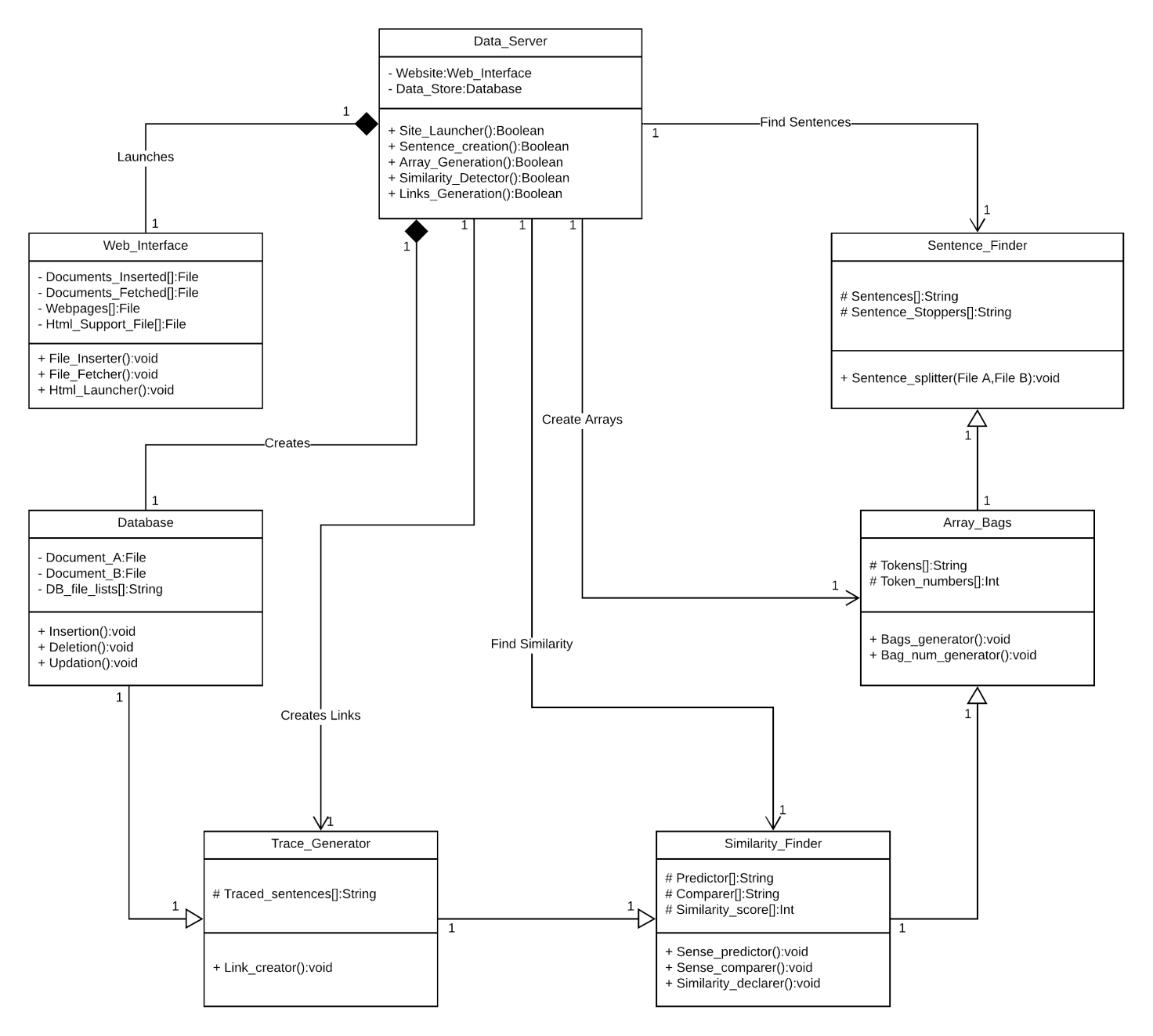
Sentences with low similarity score nearly equal to zero would be considered as equivalent and then these type of sentences would be stored in a different array.

* **Trace Generator:** After the completion of the process of calculating the similarity scores there would be a process of linking the sentences in the documents to the sentences which are similar to them in other documents and this would be like once the user will hover the mouse on the sentences which have some similar sentence in the other documents then that similar sentence would be poped at the top of the sentence over which the mouse is hovered and there is also a link embeeded to that sentence so that if user wants to follow the similar documents then he can follow the link for that.
* **Online Interaction:** Once this all process of generation of similarity score would be done at the backend of the web interface, a message will be pop to view the results of the performed comparison and that can be done by clicking the button below that.

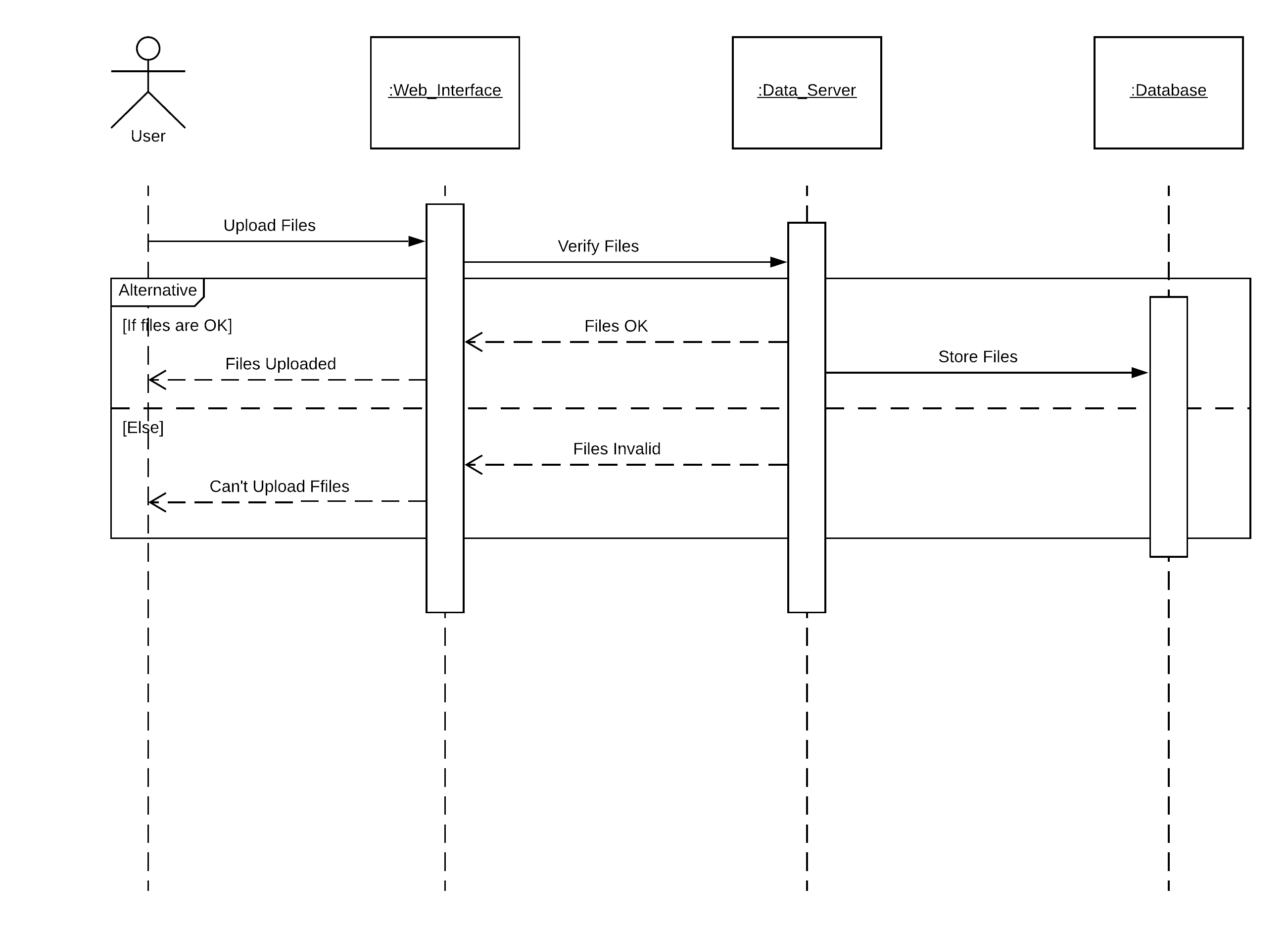
**3.6 Conceptual Models:**



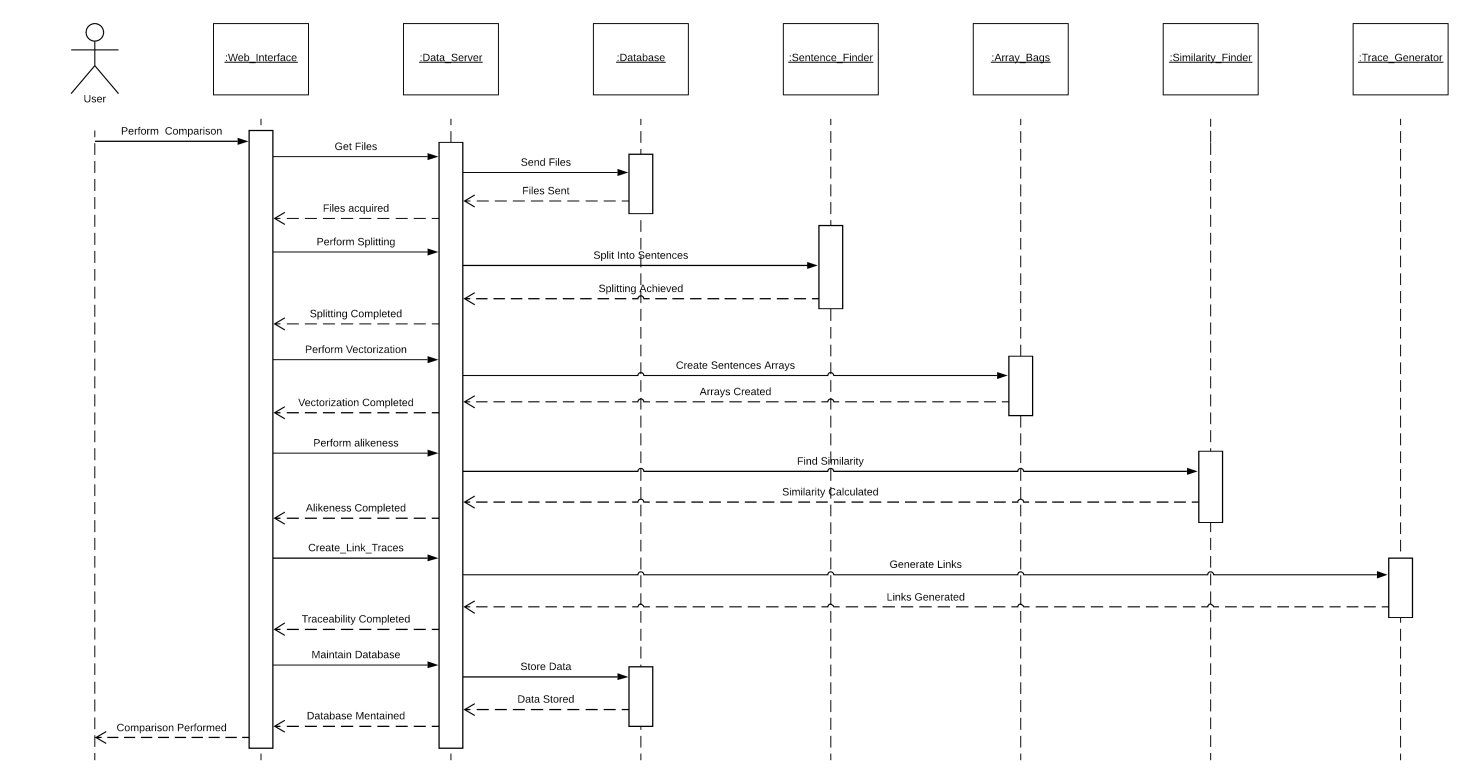
**1. USE CASE DIAGRAM**



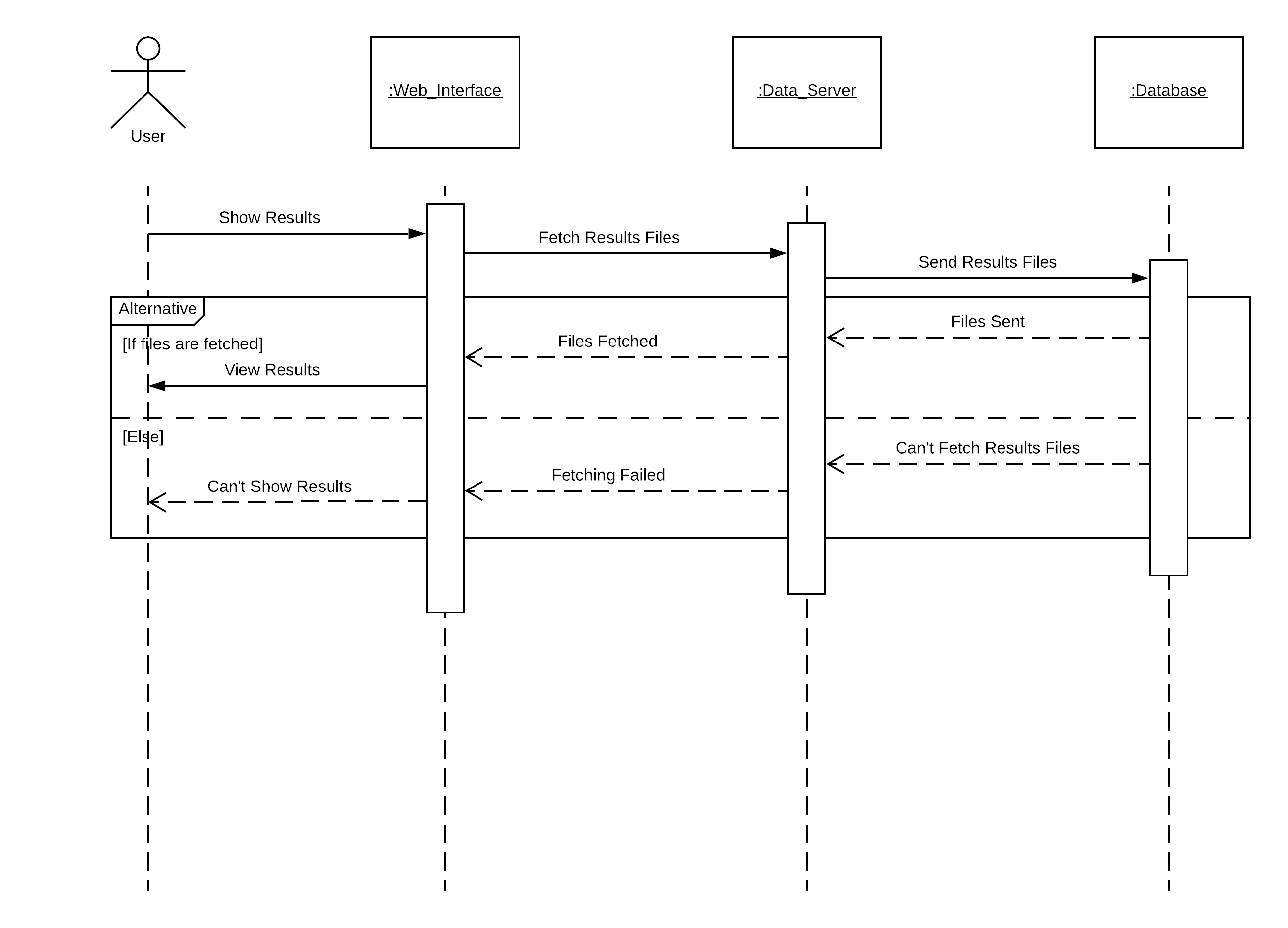
**2. CLASS DIAGRAM**



**3. SEQUENCE DIAGRAM FOR UPLOADING FILES**



**4. SEQUENCE DIAGRAM FOR FILE COMPARISON**



**5 SEQUENCE DIAGRAM FOR VIEWING THE RESULTS**

**4 References:**

Jian Xu, Qin Lu. 2013. PolyUCOMP-CORE TYPED: Computing Semantic Textual Similarity

using Overlapped Senses

The Hong Kong Polytechnic University, Department of Computing, Hung Hom, Kowloon, Hong Kong.

Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task, pages 90–95, Atlanta, Georgia, June 13-14, 2013. c 2013 Association for Computational Linguistics.

Daniel B., Chris Biemann, Iryna Gurevych and Torsten Zesch. 2012. UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures.

Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in

conjunction with the First Joint Conference on Lexical and Computational Semantics (\*SEM 2012).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (\*SEM 2012).

Frane Saric, Goran Glavas, Mladen Karan, Jan Snajder and Bojana Dalbelo Basia. 2012. TakeLab: Systems for Measuring Semantic Text Similarity. Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (\*SEM 2012).

https://towardsdatascience.com

https://medium.freecodecamp.org

https://dataquest.io