

Design Document  
Plagiarism Detection Application  
**Aronee Dasgupta (arone008@umn.edu)**

*To use the application, first create the .class files by running `javac *.java` at the root.*

*Usage - `java PlagiarismDetectionApplication <synonym_filename> <base_filename> <comparision_filename> <tuple_size>`*

1. The main driver class of the application calls several services. This has been done for future extensibility. In case the business logic changes, then that service will only need to be changed. And the main driver application can remain unchanged.
2. I have a Model to represent the input. The input currently takes in the synonyms filename, the two text files which need to be checked for plagiarism and an optional tuple size. This model can be extended, if the application needs more parameters (example – min threshold, metadata, etc).
3. A class to represent a Tuple called the TupleModel. In case in future, the definition of a tuple changes, I can include those new parameters in this model.
4. The InputModel class has a builder pattern. This is done so that in future for extension, when the input model has more variables and doing this with a constructor, would cause the constructor to bloat. The builder class is included as a static inner class.
5. I have separate GenerateService class. This class will generate a HashMap with a word and its synonyms (in a HashSet) as key value pair. The reason of using a HashSet is that the average retrieval time in a HashSet is  $O(1)$  (this is highly dependent on the hash function and the compression map in that function).
6. To read from the file I wrap the BufferedReader in a AtomicReference so that it becomes a immutable object instance and we can have multiple threads reading from a file. Since this operation is atomic, there is synchronization between the read streams (there will be a lock on the read stream).
7. I have a separate class for calculating the plagiarism between two files. If the logic to calculate the plagiarism changes, then it can be done in this class itself. Currently, if the number of words in a tuple is unequal, then I report that those two tuples don't match.