**PROGRESSIVE PLAGIARISM DETECTION BASED ON HASHING**

A PROJECT REPORT

*Submitted by*

A.KURTHIS BANU (810015205035)

P.MOWNIKA (810015205044)

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**



DEPARTMENT OF INFORMATION TECHNOLOGY

ANNA UNIVERSITY, CHENNAI 600 025

APRIL 2019

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“PROGRESSIVE PLAGIARISM DETECTION BASED ON HASHING”** is a bonafide work of **“A.KURTHIS BANU, Reg. No. 810015205035 and P.MOWNIKA, Reg. No. 810015205044”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

.

**SIGNATURE SIGNATURE**

**Dr.D.VENKATESAN Mrs.K.UMA MAHESWARI**

**HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR**

Department of Information Technology, Department of Information Technology,

University College of Engineering, University College of Engineering,

Anna University-BIT Campus, Anna University-BIT Campus,

Tiruchirappalli-620 024. Tiruchirappalli-620 024.

Submitted for the VIVA-VOCE to be held on

**Internal Examiner External Examiner**

DECLARATION

We hereby declare that the work entitled **“PROGRESSIVE PLAGIARISM DETECTION BASED ON HASHING”** is submitted in partial fulfilment of the requirement for the award of the degree in B.Tech. University College of Engineering, BIT Campus, Anna University, Tiruchirappalli, is record of our own work carried out by us during the academic year 2018 – 2019 under the supervision and guidance of **Mrs.K.UMA MAHESWARI,** Assistant Professor, Department of Information Technology, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli. The extent and source of information are derived from the existing literature have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any degree, either in this or any other University.

A.KURTHIS BANU (81001205035)

P.MOWNIKA (810015205044)

I certify that the declaration made above by the candidate is true.

Signature of the Guide,

**Mrs.K.UMA MAHESWARI**

**ASSISTANT PROFESSOR,**

Department of Information Technology,

University College of Engineering,

Anna University-BIT Campus,

Tiruchirappalli-620 024.

**ACKNOWLEDGEMENT**

It is a great opportunity to express our sincere thanks to all the people who have contributed to the successful completion of our project work through their support encouragement and guidance.

Our first and foremost thanks goes to **Dr.T.SENTHIL KUMAR,** Dean, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli for providing us the necessary facilities and supportive atmosphere for the completion of this project work.

It is our privilege to render our sincere thanks to **Dr.D.VENKATESAN,** Head of the Department of Information Technology, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli for providing us with excellent lab facilities and constantly encouraging us to pursue new goals and ideas.

We wish to record our heartfelt gratitude to our esteemed guide **Mrs.K.UMA MAHESWARI,** Assistant Professor, Department of Information Technology, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli for his excellent guidance, enterprising and valuable suggestions, encouragement and inspiration offered throughout the project.

It is our responsibility to thank our project co-ordinator **Mrs.V.M.PRIYADHARSHINI,** Assistant Professor, Department of Information Technology, deserves a special vote of thanks for his constant inspiration that she has been all through the project period. We also thank the faculty members of the Department of Information Technology, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli for their remarkable help in completing the project.

We thank all our friends who have very understood, co-operative and appreciative and also understood with us as pillar of support during our good and bad times. On the whole, we express our heartfelt greatfulness to our parents without whom we cannot be shaped up this in our carrier. Last but never the least once again we thank one and all who have helped us either directly or indirectly in completing this dissertation.

**ABSTRACT**

In this project user can investigate methods for plagiarism detection that aim to identify the uniqueness level of journal papers and find the plagiarism percentage of the journal papers. It deals with intrinsic plagiarism and extrinsic plagiarism and find the percentage of plagiarism of a journal papers. HashSet and hash-based algorithm (MD5) is used to find the plagiarism level of journals. Information retrieval(IR based approach) sometimes takes more time when the dataset is large and massive. Query expansion to deal with the situations in which original documents are obfuscated. This approach may fail to identify similarity between document pairs when the original text has been rewritten. In this system we provide efficient method for finding the plagiarism level and it is proposed for journal papers. HashSet are nothing but it is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set. HashSet determines the processing order of the blocks to find the duplicate entities. HashSet stores the elements by using a mechanism called hashing. HashSet contains unique elements only. HashSet allows null value. Even if you provide duplicate values as input and add in the set, only one instance of that value will be available. Even if the execution time is limited this detection processes larger dataset and maintain its quality.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | V |
|  | **LIST OF FIGURES** | Viii |
|  | **LIST OF TABLES** | Ix |
|  | **LIST OF ABBREVIATIONS** | X |
| **1** | **INTRODUCTION** | 1 |
|  | 1.1 Introduction plagiarism detection | 1 |
|  | 1.2 Suspicious documents | 2 |
|  | 1.3 Similarity analysis | 2 |
|  | 1.4 Preprocessing | 3 |
|  | 1.4.1 Conversion | 4 |
|  | 1.5 Hash Function | 5 |
|  | 1.6 HashSet | 5 |
| **2** | **LITERATURE SURVEY** | 5 |
|  |  |  |
| **3** | **SYSTEM ANALYSIS** | 9 |
|  | 3.1 Existing System | 10 |
|  | 3.1.1 Disadvantage | 12 |
|  | 3.1.2 Limitation | 14 |
|  | 3.2 Proposed System | 17 |
|  | 3.2.1 Adavantage of Proposed System | 18 |
|  | 3.2.2 Architecture Description | 18 |
| **4** | **SYSTEM SPECIFICATION** | 18 |
|  | 4.1 Requirements Analysis and Specification | 21 |
|  | 4.2 Resource Requirements | 21 |
|  | 4.3 Functional Requirements | 21 |
|  | 4.4 NON-Functional Requirements | 22 |
|  | 4.5 Technology Overview | 22 |
|  | 4.5.1 Java Technology | 23 |
|  | 4.5.2 Java Programming Language | 23 |
|  | 4.5.3 Java Platform | 25 |
|  | 4.6 History of Java | 27 |
|  | 4.7 Package Java.IO Description | 27 |
|  | 4.8 Java Swing | 27 |
|  | 4.8.1 The Java 2D API Packages | 27 |
|  | 4.8.2 Java Swing |Look and Feel | 28 |
|  | 4.9 Apache Tomcat | 32 |
| **5** | **SYSTEM DESIGN-UML DIAGRAMS** | 41 |
|  | 5.1 UML Diagrams | 41 |
|  | 5.2 Goals | 41 |
|  | 5.3 Data Flow Diagram | 41 |
| **6** | **MODULE DISCRIPTION** | 41 |
|  | 6.1 Admin Functionality | 43 |
|  | 6.2 User Functionality | 45 |
|  | 6.3 Checking Module | 46 |
| **7** | **CONCLUSION** | 47 |
|  | **REFERENCES** | 49 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 3.1 | Existing System Architecture | 3 |
| 3.2 | Proposed System Architecture | 13 |
| 4.1 | Java Virtual Machine | 16 |
| 4.2 | Java Platform | 24 |
| 5.1 | Data Flow Diagram | 24 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| 4.1 | Software Requirements | 5 |
| 4.2 | Hardware Requirements | 6 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| MD5 | Message Digest 5 |
| SHA | Secure Hash Algorithm |
| IR | Information Retrieval |
| CRC | Cyclic Redundancy Check |
| NSA | National Security Agency |
| CUIS | Concept Unique Identifiers |
| API | Application Programming Interface |
| RAM | Random Access Memory |
| JVM | Java Virtual Machine |
| GUI | Graphical User Interface |
| RMI | Remote Method Invocation |
| JDBC | Java Database Connectivity |
| DFD | Data Flow Diagram |
| RTF | Rich Text Format |

UML Unified Modelling Language

**CHAPTER 1**

**1. INTRODUCTION**

**1.1** **Introduction plagiarism detection**

The issue of plagiarism is not new; however increased ease of access to electronic material via the web is always a concern among the academic community. Although there is no direct evidence that students electronically cut and paste material into assignments, or purchase essays from cheat sites, the potential for these kinds of problems exists.

Plagiarism is the process of taking other people’s words or ideas and pretending that they are own. To steal and pass off ideas or words of other people’s as one’s own. Plagiarism can also be defined as the act of deliberately trying to camouflage your academic tutor by submitting content which is not your own work. Plagiarism generally refers to the unacknowledged copying of existing information, such as documents and programs.

Plagiarism detection is the process of locating instances of plagiarism within a document. The widespread use of computers and the advent of the internet has made it easier to plagiarize the work of other people. Most cases of plagiarism are found in academia, where documents are typically reports or essay. However, plagiarism can be found in virtually any field, including scientific papers, novels, source code and art designs.

Plagiarism also demotivates students who see their efforts as being undermined by the unfair advantage gained by others. Tutors who fail to deal with it make this situation even worse but some institutions' procedures and regulations may be so onerous and draconian that no action is taken or local arrangements are developed - leading to possible inconsistent and unfair treatment.

**Types :**

**Direct plagiarism** - copying and pasting someone else's work, or making changes to someone else's work to pass it off as their own.

**Self-plagiarism** - when a student submits all or part of their own previous work without getting permission from all involved [professors](https://simple.m.wikipedia.org/wiki/Professor).

**Mosaic plagiarism** - when parts of other works are copied without using [quotation marks](https://simple.m.wikipedia.org/wiki/Quotation_mark). It can also be when a student uses the same structure and meaning of an original passage and only uses [synonyms](https://simple.m.wikipedia.org/wiki/Synonym).

**Accidental plagiarism** - it can also happen when a student [paraphrases](https://simple.m.wikipedia.org/w/index.php?title=Paraphrase&action=edit&redlink=1) information without giving [attribution](https://simple.wiktionary.org/wiki/attribution). [Mash-up](https://simple.m.wikipedia.org/wiki/Mash-up) - two or more data sources that have been turned into one. They may be texts, songs, graphics, and video from various media.

**1.2 Suspicious documents**

Suspicious documents are those that cannot be disinfected at the time of the scan or which simply have unusual characteristics. Here we assume our input file to be the suspicious documents. Plagiarism percentage is checked for our input file by using hashing technique. The input files we take are journal papers and so the uniqueness of the journal papers are checked by using the plagiarism checker.

**1.3 Similarity analysis**

The similarity itself means the level of matching between the given input documents and the documents existing in the journal database. HashSet is used in the process of analysis the identities that exhibit atmost similarity. The suspicious document is the provided input document and the input document and the input document is been subjected to preprocessing techniques and the content is checked for plagiarism.

**1.4** **Preprocessing**

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. There are various methods like data cleaning, data integration, data transformation, data reduction and data reduction. In data cleaning the data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data. In data integration the data with different representations are put together and conflicts within the data are resolved. In data transformation the data is normalized, aggregated and generalized. In data reduction this step aims to present a reduced representation of the data in a data warehouse. In data discretization it involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

Thus preprocessing in the input file includes cleaning the document and removing the unnecessary data that reduces the efficiency of the similarity check. The techniques used for preprocessing are the pdf stripper and stopwords and stemming techniques

**1.4.1 Conversion**

The documents is now subjected to text conversion. The entire input document is converted to text format. Where the images are removed including the stopwords and stemming techniques it makes similarity check much more accurate and easy. The prominent step in the similarity check and plagiarism detection is the conversion of pdf to text in preprocessed format.

**1.5 Hash function**

A hash function is any function that can be used to map a data set of an arbitrary size to a data set of a fixed size, which falls into the hash table. The values returned by a hash function are called hash values, hash codes, hash sums, or simply hashes. Hash functions accelerate table or database lookup by detecting duplicated records in a large file.

A hash function is a mathematical function that converts an input value into a compressed numerical value – a hash or hash value. ... The length of the output or hash depends on the hashing algorithm.

Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects.

* In universities, each student is assigned a unique roll number that can be used to retrieve information about them.
* In libraries, each book is assigned a unique number that can be used to determine information about the book, such as its exact position in the library or the users it has been issued to etc.

To achieve a good hashing mechanism, it is important to have a good hash function with the following basic requirements:

1. Easy to compute: it should be easy to compute and must not become an algorithm itself.
2. Uniform distribution: it should provide a uniform distribution across the hash table and should not result in clustering.
3. Less collisions: collisions occur when pairs of elements are mapped to the same hash value.

**1.5.1 Types of Hashing**

There are many different types of hash algorithms such as ripemd, tiger, and more, but the most common type of hashing used for file integrity checks are MD5, SHA-2 and CRC32.

MD5 - an MD5 hash function encodes a string of information and encodes it into a 128-bit fingerprint. MD5 is often used as a checksum to verify data integrity. However, due to its age, MD5 is also known to suffer from extensive hash collision vulnerabilities, but it’s still one of the most widely used algorithms in the world.

SHA-2 – SHA-2, developed by the National Security Agency (NSA), is a cryptographic hash function. SHA-2 includes significant changes from its predecessor, SHA-1. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256.

CRC32 – a Cyclic Redundancy Check (CRC) is an error-detecting code often used for detection of accidental changes to data. Encoding the same data string using CRC32 will always result in the same hash output, thus CRC32 is sometimes used as a hash algorithm for file integrity checks. These days, CRC32 is rarely used outside of zip files.

**Message Digest 5 algorithm**

The MD5 function is a cryptographic algorithm that takes an input of arbitrary length and produces a Message Digest that is 128 bits long. The digest is sometimes also called the "hash" or "fingerprint" of the input. MD5 was designed by well-known cryptographer Ronald Rivest in 1991.

Message digests, also known as hash functions, are one-way functions; they accept a message of any size as input, and produce as output a fixed-length message digest.

MD5 is the third message digest algorithm created by Rivest. All three (the others are MD2 and MD4) have similar structures, but MD2 was optimized for 8-bit machines, in comparison with the two later formulas, which are optimized for 32-bit machines. The MD5 algorithm is an extension of MD4, which the critical review found to be fast, but possibly not absolutely secure. In comparison, MD5 is not quite as fast as the MD4 algorithm, but offered much more assurance of data security.

**1.6 HashSet**

HashSet extends abstract set and implements the set interface. It creates a collection that uses a hash table for storage. A hash table stores information by using a mechanism called hashing. In hashing, the informational content of a key is used to determine a unique value, called its hash code. The hash code is then used as the index at which the data associated with the key is stored. The transformation of the key into its hash code is performed automatically.

HashSet stores the elements by using a mechanism called hashing.

HashSet contains unique elements only.

HashSet allows null value.

HashSet class is non synchronized.

HashSet doesn't maintain the insertion order. Here, elements are inserted on the basis of their Hash code.

HashSet is the best approach for search operations.

**CHAPTER 2**

**LITERATURE SURVEY**

Nawab, R.M.A., Stevenson, m. And clough, p. (2017)[1] presented “An IR-based approach utilising query expansion for plagiarism detection in med line”. They discussed about the identification of duplicated and plagiarised passages of text has become an increasingly active area of research. In this paper we investigate methods for plagiarism detection that aim to identify potential sources of plagiarism from med line, particularly when the original text has been modified through the replacement of words or phrases. A scalable approach based on information retrieval is used to perform candidate document selection - the identification of a subset of potential source documents given a suspicious text - from med line. Query expansion is performed using the ulms Metathesaurus to deal with situations in which original documents are obfuscated. Various approaches to word sense disambiguation are investigated to deal with cases where there are multiple concept unique identifiers (CUIS) for a given term. Results using the proposed IR-based approach outperform a state-of-the-art baseline based on Kullback-leibler distance

# Daegu, Korea [2] presented “Automatic source code plagiarism detection” here they discussed about Plagiarism is one form of academic dishonesty, which is often done by students in programming classes. In a large class, detecting plagiarism manually is both difficult and time-consuming, especially due to the numerous modifications of the source code to conceal the cheating. we designed and developed deimos, a prototype of a source code plagiarism detector, which can be extended to handle other programming languages, simply by implementing new scanners and parsers. Deimos works in two steps: (1) parsing source code and transforming it into tokens, and then (2) comparing each pair of token strings obtained in the first step using running Karp-Rabin greedy string tiling algorithm. Instructor can access deimos via a web application interface that receives input parameters, triggers a background process, and displays the result. The web interface offers user friendliness while the background process prevents timeout and reduces bandwidth consumption. This approach was chosen since deimos is intended to be used for processing more than 100 source code. The web application was implemented using php, while java was used to implement the backend application, which is responsible for the background process unit test, functional test, and nonfunctional test has been conducted. Detection time is 1 hour for processing 100 samples of beginner's source code taken from real assignment of our programming class where the average length of source code is 150 lines. This code similarity detector could also be used for other pedagogical tools, such as autograder, which checks consistency of source code based on a template or solution.

# [Basel Halak](https://www.researchgate.net/profile/Basel_Halak) [Mohammed el-Hajjar](https://www.researchgate.net/profile/Mohammed_El-Hajjar) [3] presented “Plagiarism detection and prevention techniques in engineering education ” they focused on this paper plagiarism seriously damages the education process in a number of ways; it prevents students from developing the skills of creative thinking and critical analysis; it undermines the trust between lectures and students, and if goes undetected, it can impact the reputation of the academic institution and devalue its degrees. In this paper, we present two techniques for plagiarism detection and prevention. The first method is based on the allocation of a unique assignment for each student, and the second approach is based on the use of individual presentation of coursework findings. These techniques are applied to three courses at the master level in the university of southampton, where we show that they are effective at reducing plagiarism and improving students’ understanding.

Thorsten Papenbrock, Arvid Heise, and Felix Naumann [4] presented

# “Progressive duplicate detection ” here they deal with duplicate detection is the process of identifying multiple representations of same real world entities. Today, duplicate detection methods need to process ever larger datasets in ever shorter time: maintaining the quality of a dataset becomes increasingly difficult. We present two novel, progressive duplicate detection algorithms that significantly increase the efficiency of finding duplicates if the execution time is limited: they maximize the gain of the overall process within the time available by reporting most results much earlier than traditional approaches. Comprehensive experiments show that our progressive algorithms can double the efficiency over time of traditional duplicate detection and significantly improve upon related work.

# Dhivyabharathi GV, Kumaresan S[5] presented “A survey on duplicate record detection in real world data ” here deal with often, in the real world, entities have two or more representations in databases. Duplicate records do not share a common key and/or they contain errors that make duplicate matching a difficult task. Errors are introduced as the result of transcription errors, incomplete information, lack of standard formats, or any combination of these factors. In this paper, we present a thorough analysis of the literature on duplicate record detection. We cover similarity metrics that are commonly used to detect similar field entries, and we present an extensive set of duplicate detection algorithms that can detect approximately duplicate records in a database. We also cover multiple techniques for improving the efficiency and scalability of approximate duplicate detection algorithms. We conclude with coverage of existing tools and with a brief discussion of the big open problems in the area.

# [Brian martin](http://www.uow.edu.au/arts/sts/bmartin/pubs/94jie.html) [6] presented “Plagiarism: a misplaced emphasis” Plagiarism is conventionally seen as a serious breach of scholarly ethics, being a theft of credit for ideas in a competitive intellectual marketplace. This emphasis overlooks the vast amount of institutionalized plagiarism, including ghostwriting and attribution of authorship to bureaucratic elites. There is a case for reducing the stigma for competitive plagiarism while exposing and challenging the institutionalized varieties.

# [Caroline Lyon](https://www.researchgate.net/profile/Caroline_Lyon) James Malcolm[7] presented “Demonstration of the ferret plagiarism detector” they introduces a demonstration of ferret, a plagiarism detection tool developed at the university of hertfordshire, and describes the re-implementation of the system to improve its maintainability. Both text and graphical user interfaces are now provided to the core ferret algorithm, and in the new version the input of tokens may be modified to suit the type of source document.

# “Data Profiling” by Thorsten Papenbrock[8]presented “pay-as-you go entity resolution,” IEEE transactions on knowledge and data engineering (TKDE), vol. 25, no. 5, 2012, has been proposed the data profiling is the improving data. That is improve data quality and improve understanding of data for the users.

“Data Cleansing” by Arvid Heise[9] presented “Record Linkage: making maximum use of the discriminating power of identifying information communications of the ACM”, vol. 5, no. 11, 2011, has been proposed the data cleansing is to detecting the all files. Data cleansing is the process of detecting and corrupt and inaccurate from a datasets or database and replacing, modifying, deleting in datasets.

“Duplicate Detection” by Felix Naumann[10] presented “An introduction to duplicate detection” Morgan & Claypool, 2010, has been proposed by three techniques. That is profiling, mining and cleansing. It’s describes the key analysis, link analysis, data type analysis inverse predicate discovery synonym predicate discovery auto-completion

Aristides Gionis Piotr Indyky Rajeev Motwaniz[11]presented “Similarity search in high dimensions via Hashing”. The nearest- or near-neighbor query problems arise in a large variety of database applications, usually in the context of similarity searching. Of late, there has been increasing interest in building search/index structures for performing similarity search over high-dimensional data, e.g., image databases, document collections, time-series databases, and genome databases. Unfortunately, all known techniques for solving this problem fall prey to the \curse of dimensionality." that is, the data structures scale poorly with data dimensionality; in fact, if the number of dimensions exceeds 10 to 20, searching in k-d trees and related structures involves the inspection of a large fraction of the database, thereby doing no better than brute-force linear search. It has been suggested that since the selection of features and the choice of a distance metric in typical applications is rather heuristic, determining an approximate nearest neighbor should suffice for most practical purposes. In this paper, we examine a novel scheme for approximate similarity search based on hashing. The basic idea is to hash the points.

Rupali Vairagade, Shantanu Surve, Karan Naik, Ganesh Sonawane, Jayant Athawale[12] presented “A survey of sorted neighbourhood indexing technique for deduplication”. The process of matching records which are from several databases and which refer to the same entities is known as record linkage. When you apply this process on a single database, this process is known as de-duplication. Matched data are becoming an important in many application areas, because they can contain information that is not available, or it is too costly to acquire. Thus, removing duplicate records from a single database is an important step in the data cleaning process as duplicates can severely affect the outcomes of any subsequent data processing or data mining. Because of the constant increase in the size of today’s databases, the difficulty of the matching process has become one of the major issue for record linkage and de-duplication. Recently, various indexing methods have been developed for record linkage and de-duplication. Thus, by removing the obvious non-matching pairs these methods have been aimed at reducing the number of record pairs to be compared during the matching process. This paper presents the survey of sorted neighbourhood indexing technique.

# 

# 

# CHAPTER 3

**SYSTEM ANALYSIS**

**3.1 Existing System**

An IR-based approach utilising query expansion for plagiarism detection in MEDLINE Rao Muhammad Adeel Nawab, Mark Stevenson and Paul clough

the identification of duplicated and plagiarised passages of text has become an increasingly active area of research. In this paper we investigate methods for plagiarism detection that aim to identify potential sources of plagiarism from med line, particularly when the original text has been modified through the replacement of words or phrases. A scalable approach based on information retrieval is used to perform candidate document selection - the identification of a subset of potential source documents given a suspicious text - from med line. Query expansion is performed using the ulms Metathesaurus to deal with situations in which original documents are obfuscated. Various approaches to word sense disambiguation are investigated to deal with cases where there are multiple Concept Unique Identifiers (CUIS) for a given term. Results using the proposed IR-based approach outperform a state-of-the-art baseline based on Kullback-leibler distance.

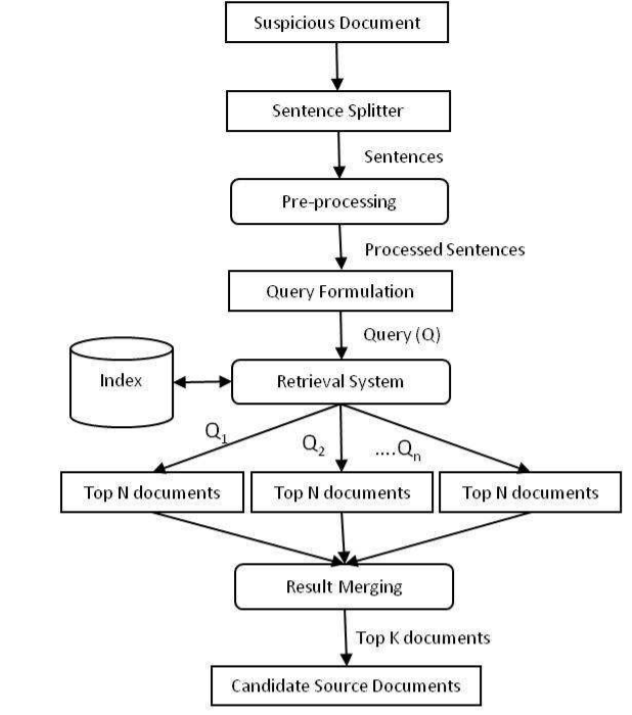
* In progressive plagiarism detection basic two approaches user uses are,
* The adaptive and progressive technique.
* Adaptive technique like prosed algorithm usage for estimating the comparision candidates quality.
* Progressive technique like pay-as-you-go algorithm for information integration.
* User don’t combine our progressive approaches with scalable approaches for duplicate detection.

**3.1.1 Disadvantages**

* A user has only limited, maybe unknown time for data cleansing and wants to make best possible use of it. Then, simply start the algorithm and terminate it when needed. The result size will be maximized.
* A user has little knowledge about the given data but still needs to configure the cleansing process.
* A user needs to do the cleaning interactively to, for instance, find good sorting keys by trial and error. Then, run the progressive algorithm repeatedly; each run quickly reports possibly large results.
* All presented hints produce static orders for the comparisons and miss the opportunity to dynamically adjust the comparison order at runtime based on intermediate results.

**3.1.2 Limitations**

* + Time complexity.
  + Difficult implementation.
  + Collusion-related plagiarism.
  + Ineffectual results.
  + Syntactic effect.



**Fig 3.1 Existing System Architecture**

**3.2 Proposed System**

In this system the prestored documents in the journal database is subjected to conversion of hash code and entirely stored in the HashSet. Preprocessing techniques like stemming and Stopwords are used in the removal of unwanted contact for efficient conversion to hash code. The conversion is performed using md5 algorithm. Now the provided suspicious input file is checked for plagiarism by comparing the hash code converted format of the input document and the journal database document. The input file is also converted to hash code using md5 algorithm. The similarity check is executed in the hash set between the input files and the database and the percentage of plagiarism is produced as the output.

**3.2.1 Advantage of proposed system**

To the best of our knowledge, this work represents the first effort to systematically detect plagiarism level existing in the journal database.

We have evaluated our system using various journal databases that holds input files from various aspects.

It identifies the similarities of the journal papers.

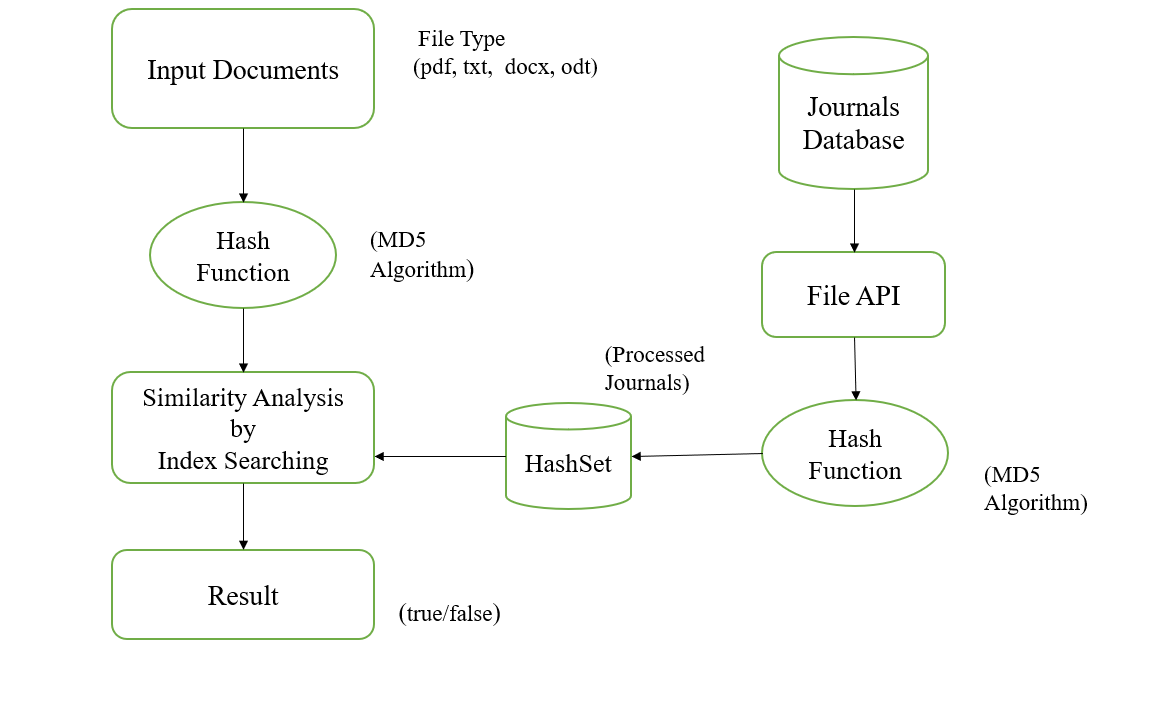
It also offers a proof that the journal papers are not plagiarized.

It validates the uniqueness level of the journal papers.

It also finds the paraphrased contents.

Efficient time complexity in the process of plagiarism detection is the major advantage of the proposed system.

Architecture diagram



**Fig 3.2 Proposed System Architecture**

**3.2.2 Architecture Description**

**Input documents**

The input document is the database where all the predefined input files are stored .The file stored will be selected and checked for plagiarism against the entire journal database.

**Hash function**

A hash function is a mathematical function that converts an input value into a compressed numerical value – a hash or hash value. The length of the output or hash depends on the hashing algorithm. Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects.

**Message Digest 5 algorithm**

The MD5 message digest hashing algorithm processes data in 512-bit blocks, broken down into 16 words composed of 32 bits each. The output from md5 is a 128-bit Message Digest value.

Computation of the MD5 digest value is performed in separate stages that process each 512-bit block of data along with the value computed in the preceding stage. The first stage begins with the message digest values initialized using consecutive [hexadecimal](https://whatis.techtarget.com/definition/hexadecimal) numerical values. Each stage includes four message digest passes which manipulate values in the current data block and values processed from the previous block. The final value computed from the last block becomes the MD5 digest for that block.

**HashSet**

This class implements an efficient abstraction for storing sets of distinct elements. This class is identical to the set class except for the fact that it uses a hash table as its underlying representation. The advantage of the HashSet class is that it operates in constant time, as opposed to the *o*(log *n*) time for the set class.

The main advantage of HashSet is it will not let you add duplicate values. Even if you provide duplicate values as input and add in the set, only one instance of that value will be available.

**CHAPTER 4**

**SYSTEM SPECIFICATION**

**4.1 Requirement Analysis and Specification**

The requirement engineering process consists of feasibility study, requirements elicitation and analysis, requirements specification, requirements validation and requirements management requirements elicitation and analysis is an iterative process that can be represented as a spiral of activities, namely requirements discovery, requirements classification and organization, requirements negotiation and requirements documentation.

**4.2 Resource Requirements**

Software requirements is a sub-field of software engineering that deals with the elicitation, analysis, specification and validation of requirements for software requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders such as users. Requirements analysis is critical to the success of the development project requirements must be actionable, measurable, testable related to identified business needs or opportunities and defined to a level of detailed sufficient for system design.

**4.3 Functional Requirements**

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality.

In this system following are the functional requirements: The Admin has to login by using valid user name and password. After login Successful the admin gets the input file and checks for any plagiarized content against all the available journals in the database. The admin analysis the entire database and then proceeds with the process of plagiarism detection.

**4.4 Non– Functional Requirements**

Non – Functional requirements, as the name suggests, are those requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability response time and store occupancy. Alternatively, they may define constraints on the system such as the capability of the Input Output devices and the data representations used in system interfaces. Many non-functional requirements relate to the system as whole rather than to individual system features. This means they are often critical than the individual functional requirements. The following non-functional requirements are worthy of attention.

**4.5 Software Requirements**

|  |  |
| --- | --- |
| Operating system | Windows10 |
| Front end | Java swing |
| Back end | PHP |
| Language | Java |
| Database | sql server 2008 |

**4.6 Hardware Requirements**

|  |  |
| --- | --- |
| Processor | Core i5 8th gen |
| Hard disk | 1 TB |
| RAM | 4GB |

**4.7 TECHNOLOGY OVERVIEW**

## **4.7.1 Java Technology**

Java technology is both a programming language and a platform.

**4.7.2 Java Programming Language**

### The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

Simple

Object oriented

Portable

Distributed

High performance

Most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



**Fig 4.1 Java Virtual Machine**

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

**4.7.2.1 Java Platform**

A platform is the hardware or software environment in which a program runs. We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it’s a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

* The Java Virtual Machine (Java VM)
* The Java Application Programming Interface (Java API)

Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages*.* Depicts a program that’s running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability. The most common types of programs written in the Java programming language are appletsandapplications*.* If you’ve surfed the Web, you’re probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs. An application is a standalone program that runs directly on the Java platform. A special kind of application known as a serverserves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet*.* A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

**The essentials**, Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on. **Applets**.The set of conventions used by applets. **Networking**, URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses. **Internationalization**, Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

**Security**, Both low level and high level, including electronic signatures, public and private key management, access control, and certificates. **Software components**, Known as JavaBeansTM, can plug into existing component architectures. **Object Serialization**, Allows lightweight persistence and communication via Remote Method Invocation (RMI). **Java Database Connectivity (JDBCTM),** Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



**Fig 4.2 Java platform**

We can’t promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages.

# 4.8 History of Java

**The history of java** is very interesting. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of java starts with green team. Java team members (also known as **green team**), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was suited for internet programming. Later, java technology was incorporated by netscape.

The principles for creating java programming were "simple, robust, portable, platform-independent, secured, high performance, multithreaded, architecture neutral, object-oriented, interpreted and dynamic".

Currently, java is used in internet programming, mobile devices, games, e-business solutions, etc. There are given the significant points that describe the history of java.

1) [James Gosling](https://www.javatpoint.com/james-gosling-father-of-java), **Mike Sheridan**, and **Patrick Naughton** initiated the java language project in June 1991. The small team of sun engineers called **green team**.

2) Originally designed for small, embedded systems in electronic appliances like set-top boxes.

3) Firstly, it was called **"Greentalk"** by James gosling, and file extension was .gt.

4) After that, it was called **oak** and was developed as a part of the green project.

## Why java named "oak"?

5) **Why oak?** Oak is a symbol of strength and chosen as a national tree of many countries like U.S.A., France, Germany, Romania, etc.

6) In 1995, oak was renamed as **"java"** because it was already a trademark by oak technologies.

**4.8.1 Java Platform**

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

1. Runtime environment
2. API(Application Programming Interface)

Java code can be run on multiple platforms, for example, windows, Linux, Sun Solaris, MAC/OS, etc. Java code is compiled by the compiler and converted into byte code. This byte code is a platform-independent code because it can be run on multiple platforms, i.e., write once and run anywhere (wora).

Java is an [object-oriented](https://www.javatpoint.com/java-oops-concepts) programming language. Everything in java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.

## **4.8.2 Package Java.IO Description**

This package provides for system input and output through data streams, serialization and the file system. Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a null pointer exception to be thrown.

**Following are the important classes in java.io package:**

* [Buffered input stream](https://www.geeksforgeeks.org/java-io-bufferedinputstream-class-java/)
* [Buffered output stream](https://www.geeksforgeeks.org/java-io-bufferedoutputstream-class-java/)
* [Buffered reader](https://www.geeksforgeeks.org/java-io-bufferedreader-class-java/)
* [Buffered writer](https://www.geeksforgeeks.org/io-bufferedwriter-class-methods-java/)
* [Byte array input stream](https://www.geeksforgeeks.org/io-bytearrayinputstream-class-java/)
* [Byte array output stream](https://www.geeksforgeeks.org/io-bytearrayoutputstream-class-java/)
* [Char array reader](https://www.geeksforgeeks.org/java-io-chararrayreader-class-java/)
* [Console](https://www.geeksforgeeks.org/java-io-console-class-java/)
* Data input stream
* [Data output stream](https://www.geeksforgeeks.org/dataoutputstream-in-java/)
* [File](https://www.geeksforgeeks.org/file-class-in-java/)
* [File descriptor](https://www.geeksforgeeks.org/java-io-filedescriptor-java/)
* [File input stream](https://www.geeksforgeeks.org/java-io-fileinputstream-class-java/)
* [File output stream](https://www.geeksforgeeks.org/creating-a-file-using-fileoutputstream/)
* [File permission](https://www.geeksforgeeks.org/java-io-filepermission-class-java/)
* [File reader and file writer](https://www.geeksforgeeks.org/file-handling-java-using-filewriter-filereader/)
* [Filter input stream](https://www.geeksforgeeks.org/java-io-fileinputstream-class-java/)
* [Filter output stream](https://www.geeksforgeeks.org/java-io-filteroutputstream-class-java/)
* [Filter reader](https://www.geeksforgeeks.org/java-io-filterreader-class-java/)
* [Filter writer](https://www.geeksforgeeks.org/java-io-filterreader-class-java/)
* [Input stream](https://www.geeksforgeeks.org/java-io-inputstream-class-in-java/)
* [Input stream reader](https://www.geeksforgeeks.org/java-io-inputstreamreader-class/)
* [Line number input stream](https://www.geeksforgeeks.org/java-io-linenumberinputstream-class-java/)
* [Line number reader](https://www.geeksforgeeks.org/java-io-linenumberreader-class-java/)
* Object input stream
* Object input stream. Get field
* Object output stream
* Object output stream. Put field
* [Object stream class](https://www.geeksforgeeks.org/java-io-objectstreamclass-java/)
* Object stream field
* [Output stream](https://www.geeksforgeeks.org/java-io-outputstream-class-java/)
* [Output stream writer](https://www.geeksforgeeks.org/java-io-outputstreamwriter-class-methods/)
* [Piped input stream](https://www.geeksforgeeks.org/java-io-pipedinputstream-class-java/)
* [Piped output stream](https://www.geeksforgeeks.org/java-io-pipedoutputstream-class-java/)
* [Piped reader](https://www.geeksforgeeks.org/java-io-pipedreader-class-java/)
* [Piped writer](https://www.geeksforgeeks.org/java-io-pipedwriter-class-java/)
* Print stream
* Print writer
* [Pushback input stream](https://www.geeksforgeeks.org/java-io-pushbackinputstream-class-java/)
* [Pushback reader](https://www.geeksforgeeks.org/java-io-pushbackreader-class-java/)
* [Reader](https://www.geeksforgeeks.org/java-io-reader-class-java/)
* [Writer](https://www.geeksforgeeks.org/java-io-writer-class-java-2/)

**4.8.3 Java Swing**

Java swing is a lightweight java Graphical User Interface (GUI) widget toolkit that includes a rich set of widgets. It is part of the Java Foundation Classes (JFC) and includes several packages for developing rich desktop applications in java. Swing includes built-in controls such as trees, image buttons, tabbed panes, sliders, toolbars, color choosers, tables, and text areas to display http or Rich Text Format (RTF). Swing components are written entirely in java and thus are platform-independent.

## **4.8.5 The Java 2D API Packages**

The java 2D API classes are organized into the following packages:

* Java.awt
* Java.awt.geom
* Java.awt.font
* Java.awt.color
* Java.awt.image
* Java.awt.image.renderable
* Java.awt.print

# 4.8.6 Java Swing | Look and Feel

Swing is **GUI widget toolkit** for java. It is an API for providing graphical user interface to java programs. Unlike awt, swing components are written in java and therefore are platform-independent. Swing provides platform specific look and feel and also an option for pluggable look and feel, allowing application to have look and feel independent of underlying platform.

Initially there were very few options for colors and other settings in java swing, that made the entire application look boring and monotonous. With the growth in java framework, new changes were introduced to make the UI better and thus giving developer opportunity to enhance the look of a java swing application.

**“Look” refers to the appearance of GUI widgets and “Feel” refers to the way the widgets behave**.

**4.9 Apache Tomcat**

It is often referred to as tomcat server, is an open-source [java servlet container](https://en.wikipedia.org/wiki/Servlet_container) developed by the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF). Tomcat implements several [java](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition)  specifications including [java servlet](https://en.wikipedia.org/wiki/Java_Servlet), [Java Server Pages](https://en.wikipedia.org/wiki/JavaServer_Pages) (JSP), [java el](https://en.wikipedia.org/wiki/Unified_Expression_Language), and [web socket](https://en.wikipedia.org/wiki/WebSocket), and provides a "pure [java](https://en.wikipedia.org/wiki/Java_(programming_language))" [http](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [web server](https://en.wikipedia.org/wiki/Web_server) environment in which [java](https://en.wikipedia.org/wiki/Java_(programming_language)) code can run.

Tomcat is developed and maintained by an open community of developers under the auspices of the [apache software foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation), released under the [apache license](https://en.wikipedia.org/wiki/Apache_License) 2.0 license, and is [open-source software](https://en.wikipedia.org/wiki/Open-source_software).

**CHAPTER 5**

**SYSTEM DESIGN-UML DIAGRAMS**

**5.1 UML Diagrams**

UML stands for unified modeling language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the object management group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, visualization, constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**5.2 Goals**

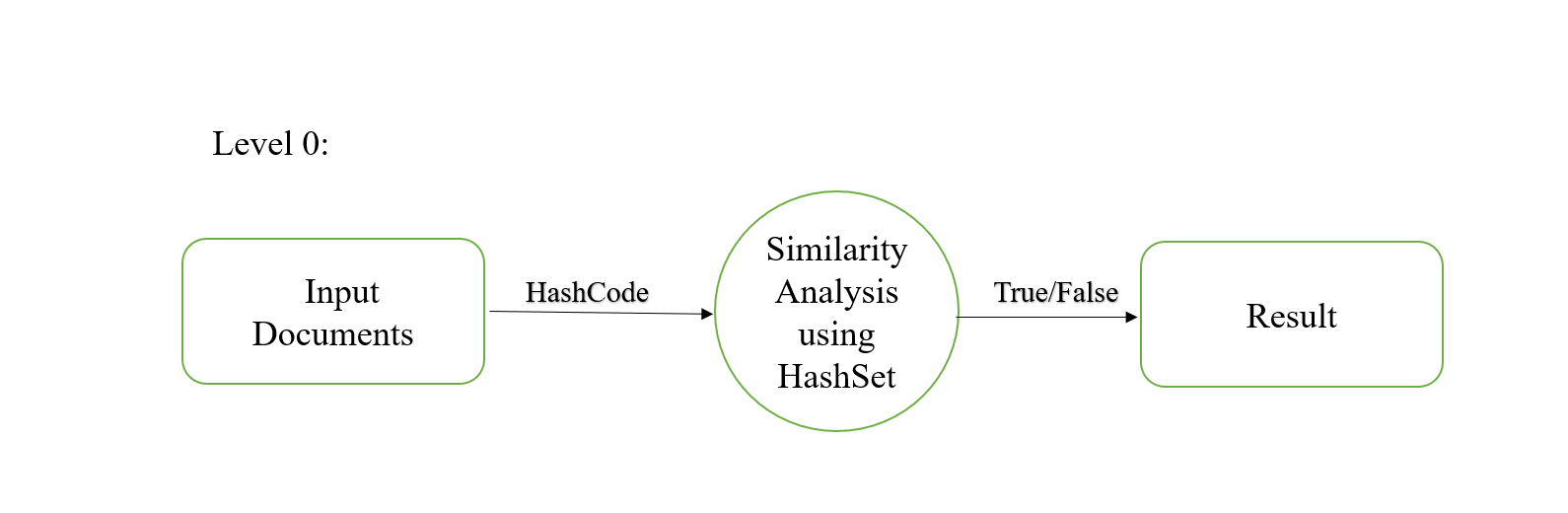
The primary goals in the design of the UML are as follows:

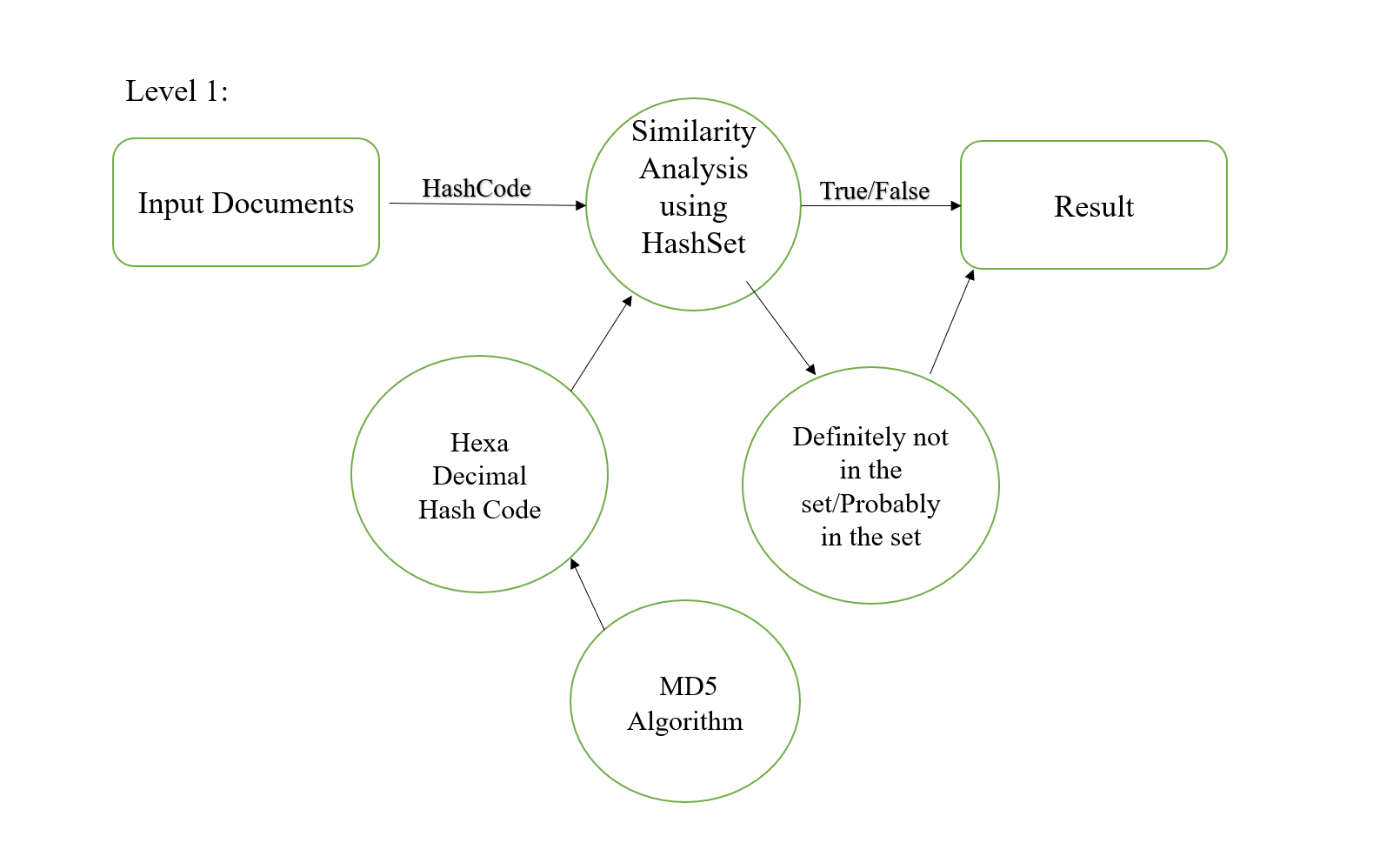
1. Provide users a ready-to-use, expressive visual modeling language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**5.3 Data flow diagram**

A Data Flow Diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFD that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFD remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

**Data Flow Diagram**





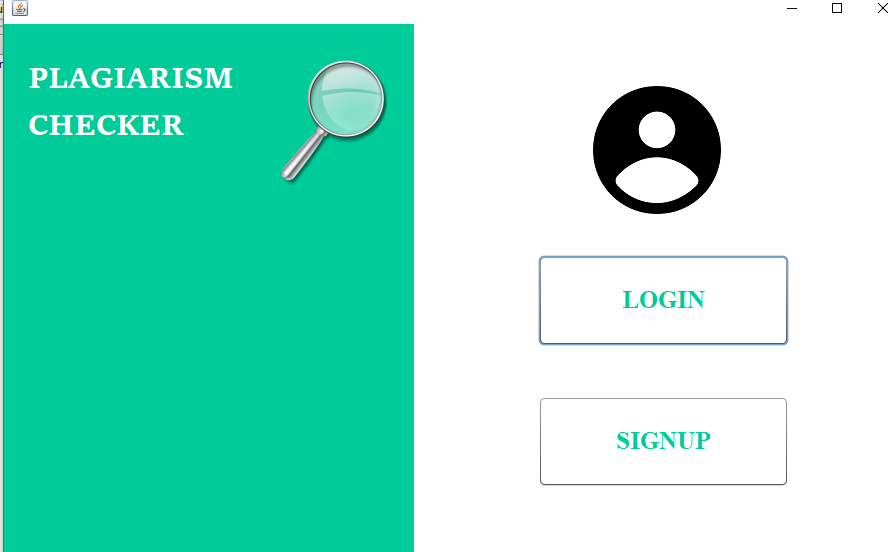
**CHAPTER 6**

**MODULE DESCRIPTION**

For detecting plagiarism level of journal papers, it has three modules to test them functionality:

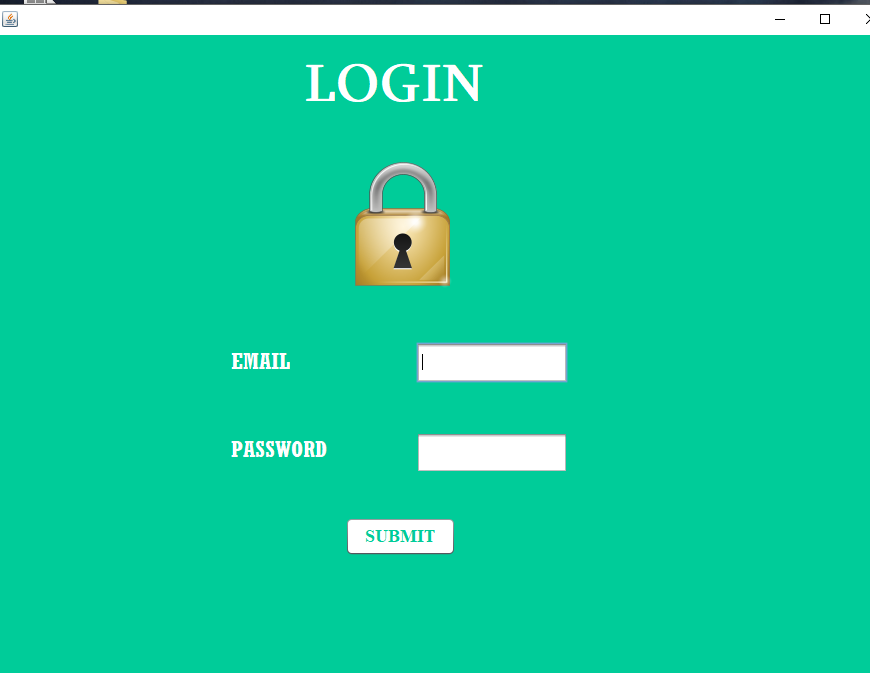
* + - 1. Admin module
      2. User module
      3. Checking module

**6.1 Admin Functionality:**

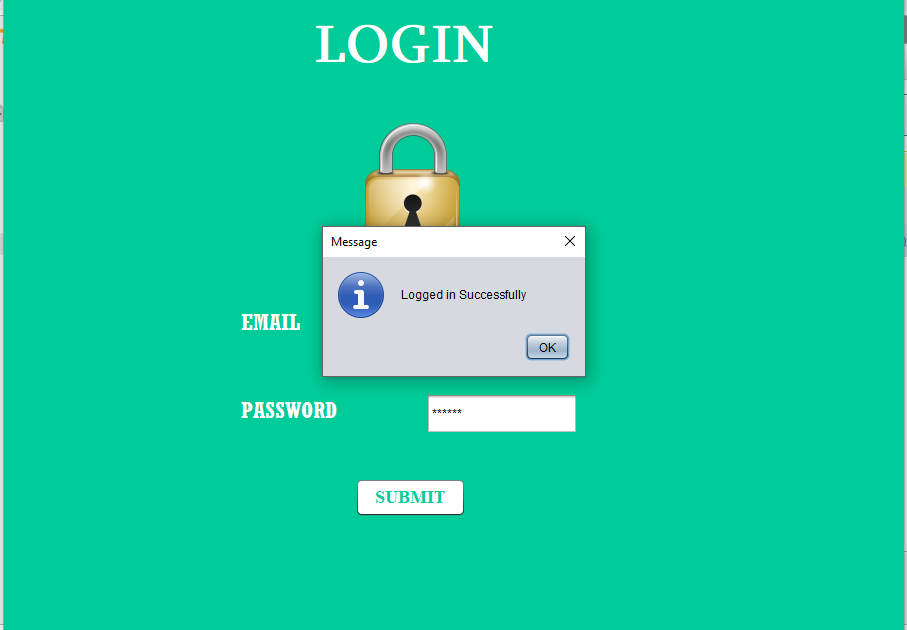


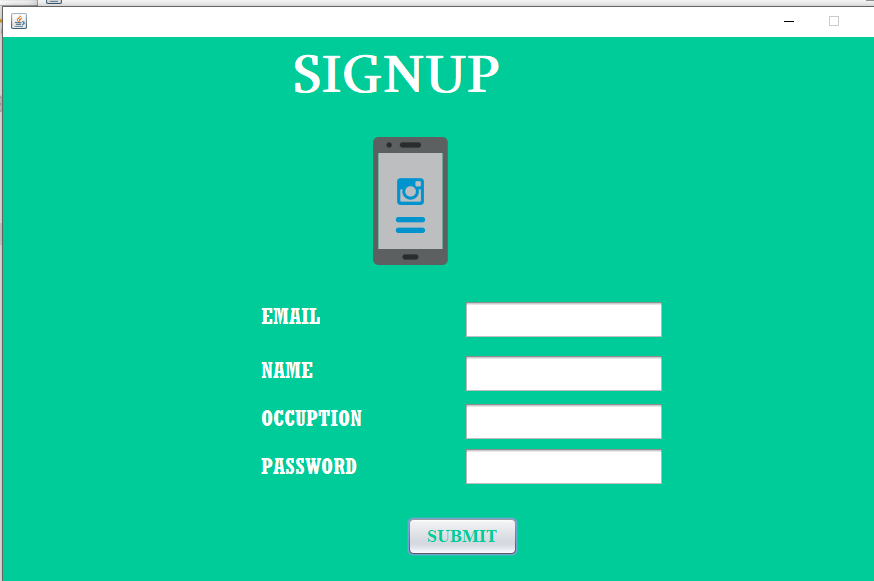
The first module is the home page here the user has the option of login and Signup. The login enables the uses to proceed to plagiarism checking. The signup is where a new user registers and joins the system.

**6.2** **User functionality:**

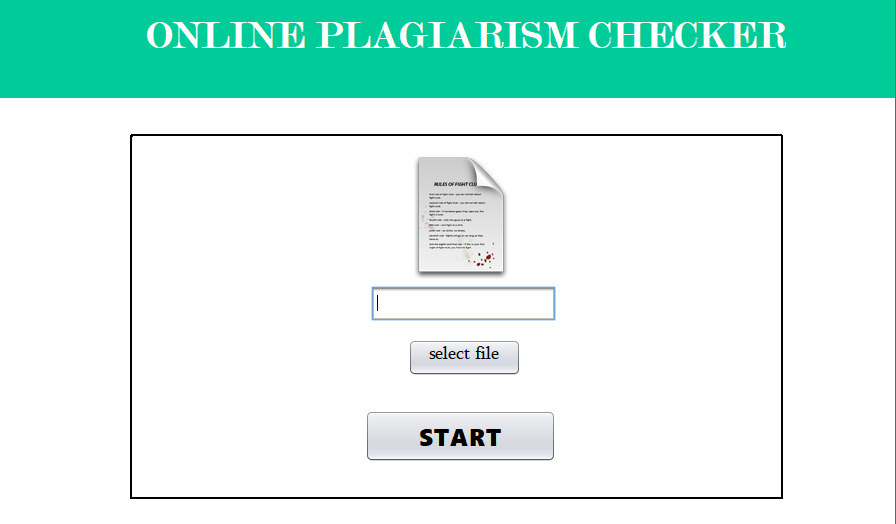


The registered users login with the specific username and password. If an incorrect username and password occurs then the login becomes unsuccessfully.





The new user registering signup with the essential details like email id, password, name of the person, and the specific occupation.



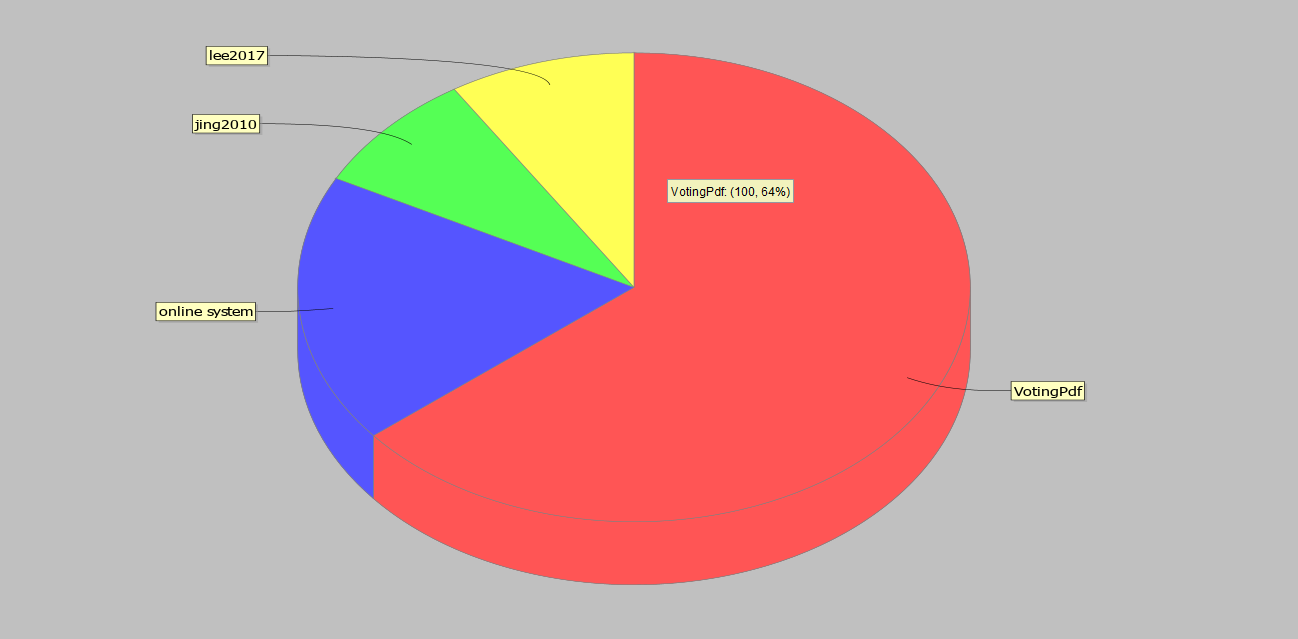
The file which is to be plagiarism checked is know as input file. It is selected and submitted for plagiarism detection.

**6.3 Checking Module**

**Pie chart**

A pie chart displays data, information, and statistics in an easy-to-read 'pie-slice' format with varying slice sizes telling you how much of one data element exists. The bigger the slice, the more of that particular data was gathered.

The main use of a pie chart is to show comparison. When items are presented on a pie chart, we can easily see which documents is the most plagiarized and which is the least plagiarized.



The output screen is displayed in the form of pie chart. The partitions in the chart represents the files from which the input file is plagiarized. The biggest partition is the document or journal from which the maximum content is plagiarized and similarly the corresponding levels of partitions represent corresponding percentage of partitions.

**CHAPTER 7**

**7.1 Conclusion**

Plagiarism is actual straight forward. When you use someone else’s work without crediting them, it is seen as stealing their intellectual property. Just like theft, the penalties for plagiarized work are also severe all over the world. The real problem is that most people are not even aware of what they are doing. there are various tools scriber, plagscan, plagramme etc…

In our Project we are introduce the concept of hash function working using MD5 algorithm in the process of plagiarism detection for journal papers. The HashSet performs the similarity check between the input file and entire journal database.

The level of plagiarism from different files are checked and detected. the detected output is represented in the pie chart format where the complete plagiarized information are expressed explicitly.

**7.2Future Enhancement**

The future work of our system is the replacement of MD5 Algorithm with the SHA algorithm for much more efficiency in plagiarized data detection.

Bloomfilter can be used for the similarity check and the addition of Synonyms checker enhances the process of similarity check and avoids the removal of unique data present in the input file.

**REFERENCES**

[1] Nawab, R.M.A., Stevenson, m. And clough, “An IR-based approach utilising query expansion for plagiarism detection in med line,” IEEE Transactions of Education,vol.Vol.42(2),2017.

# [2] Daegu, Korea,“Automatic source code plagiarism detection,” Journal of Information Ethics,vol.2,2015.

# [3] [Basel Halak](https://www.researchgate.net/profile/Basel_Halak) [Mohammed el-Hajjar](https://www.researchgate.net/profile/Mohammed_El-Hajjar) “Plagiarism detection and prevention techniques in engineering education ” Communication of the ACM ,vol.37(8),2013.

[4] Thorsten Papenbrock, Arvid Heise, and Felix Naumann

# “Progressive duplicate detection ” ,” IEEE Transactions of Education,vol,2015.

# [5]Dhivyabharathi GV, Kumaresan S “A survey on duplicate record detection in real world data ” ACM SIGCSE Bulletin,vol.34,2013.

# [6][Brian martin](http://www.uow.edu.au/arts/sts/bmartin/pubs/94jie.html) “Plagiarism:a misplaced emphasis” Tech.Rep.,2011.

# [7] [Caroline Lyon](https://www.researchgate.net/profile/Caroline_Lyon) James Malcolm “Demonstration of the ferret plagiarism detector” Proceedings of the National Academy of Sciences,vol.112,2015.

# [8] “Data Profiling” Thorsten Papenbrock “pay-as-you go entity resolution,” IEEE transactions on knowledge and data engineering (TKDE), vol. 25, no. 5, 2012.

[9] “Data Cleansing” Arvid Heise “Record Linkage: making maximum use of the discriminating power of identifying information communications of the ACM”, vol. 5, no. 11, 2011.

[10] “Duplicate Detection”Felix Naumann“An introduction to duplicate detection” Morgan & Claypool, 2010.

[11]Aristides Gionis Piotr Indyky Rajeev Motwaniz “Similarity search in high dimensions via Hashing,”Academy of Management Learning and Education, vol.5,2009.

[12] Rupali Vairagade, Shantanu Surve, Karan Naik, Ganesh Sonawane, Jayant Athawale[12] presented “A survey of sorted neighbourhood indexing technique for deduplication, ”Vine, vol.31,2008.