Republic of the Philippines
Western Mindanao State University
College of Computing Studies
DEPARTMENT OF COMPUTER SCIENCE
Zamboanga City

Development of Thesis Repository System with Document Similarity Feature for the College of Computing Studies

A Thesis presented to the faculty of
Department of Computer Science
College of Computing Studies

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Mark Anthony D. Tubat
Researcher

Ms. Lucy Felix-Sadiwa, MSCS
Adviser

April 5, 2023

Republic of the Philippines
Western Mindanao State University
College of Computing Studies
DEPARTMENT OF COMPUTER SCIENCE
Zamboanga City

Approval Sheet

The Thesis attached hereto, entitled **"Development of Thesis Repository System with Document Similarity Feature for the College of Computing Studies"**, prepared and submitted by Mark Anthony D. Tubat, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science, is hereby **recommended for Oral Examination**.

Ms. Lucy Felix-Sadiwa, MSCS
Adviser

APPROVED by the Oral Examination Committee on **April 5, 2023,** with a rating of **PASSED**.

Engr. Marvic Lines, MEnggEd-ICT
Chairperson

Mr. Jaydee C. Ballaho
Member

Mr. Salimar B. Tahil, MEnggEd - ICT
Member

ACCEPTED in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science**

Ms. Lucy Felix-Sadiwa, MSCS
Head, Department of Computer Science

RODERICK P. GO, Ph.D.
Dean, College of Computing Studies

Acknowledgement

I am very pleased to have this opportunity to give my utmost thanks to the people who became part of my journey in completing this thesis.

To Mrs. Analyn D. Tubat and Mrs. Lorna B. Tan for their patience, unwavering love, and financial aid, and for being my mothers who are always encouraging and reminding me that God is always there for everyone;

To Mr. Bernardo B. Tubat for the love, inspiration, and for being my father. Although he cannot read this anymore, I know that you are safe in God's hands;

To Ms. Lucy Felix-Sadiwa, my adviser throughout my thesis journey, for encouraging me to continue and providing valuable insights regarding my project;

To the rest of the panelists, Engr. Marvic Lines, Mr. Salimar B. Tahil, and Mr. Jaydee C. Ballaho for their comments, suggestions, and insightful questions;

To my friends, Mr. Ronald Dale Fuentebella, Mr. Jayson Beltran, Mr. Wenefredo Tejero, Mr. Emmanuel Toledo, Ms. Jane Stephanie Domingo, and Mr. Theo Jay M'lleno Sanson for accompanying me throughout this journey, laughter, support, brilliant insights;

To Ms. Shaira Pincas, for her love, encouragement and for staying and accompanying me since the beginning;

And finally, to our creator, Almighty God, for giving me struggles that I can withstand enough to make me strong and independent. God is always there when I feel lost, he gives me wisdom when I am in doubt, and he encourages me to trust and continue this journey;

I am sincerely grateful for without your steadfast encouragement, support, and direction, this thesis project could have not been finished. I dedicate the successful completion of this thesis project to everyone who has helped make it possible.

Abstract

With the yearly increase in the number of students and many graduating, there is also a growing amount of academic research documents. The College of Computing Studies (CCS) has issues regarding a need for a platform that handles both a repository of all thesis documents existing in the college and a document similarity-checking feature integrated into a single application. These features are equally important because they show academic integrity and honesty and improve document management efficiency. To address these issues, the researcher developed a web application with a digital repository of all thesis documents and a document similarity-checking feature. The study utilized three techniques, the Term Frequency-Inverse Document Frequency (TF-IDF), Cosine Similarity, and K-Nearest Neighbor. TF-IDF was used to vectorize the documents; this method calculates the numerical representation of a word in a paper based on its frequency (term frequency) and the inverse document frequency (IDF) of the expression across the corpus. Cosine Similarity is a similarity metric that compares the vectorized query document to the vectorized corpus of documents. Cosine returns 0 and 1, with 0 indicating that the compared documents were entirely different and 1 indicating that the papers were similar. K – Nearest Neighbor (K-NN) was used to display the five (5) most similar documents to a query found in the repository. The k-NN method is known as useful in classification issues. Several Natural Language Processing Techniques were also used to maintain and enhance the document similarity capability of the system. The results show positive feedback using several tests to test the effectiveness and efficiency of the web application. The application was rated as pleasing and user-friendly. The system's primary functions were also judged as working and reliable tools. The research thus concludes that the study was successful in terms of the objectives and the developed web application could play a vital role in an academic setting.

Keywords: Term Frequency-Inverse Document Frequency, Cosine Similarity, K-nearest neighbor, Natural Language Processing, Digital Repository, Document Similarity

Table of Contents

List of Figures

List of Tables

CHAPTER I
INTRODUCTION

Background of the Study

The academic environment is experiencing difficulties due to the need for an effective digital repository with integrated document similarity checking and management. These difficulties impact both students and teachers. Critical theses are difficult for students to get, which could lead to the loss of crucial knowledge and a lack of interest in further study. On the other hand, professors need help locating significant theses in the vast amount of knowledge available. Due to the difficulty in determining if a proposed study has already been conducted, students may unintentionally suggest existing studies, which might result in plagiarism. These issues are addressed by this thesis project, which creates a digital repository system with a document similarity function. A centralized platform that organizes, stores, and retrieves digital data is referred to as a digital repository system. Document similarity, used to compare theses in a digital repository, is the degree of similarity between two or more documents.

The project utilized Natural Language Processing techniques (NLP) such as Term Frequency-Inverse Document Frequency (TF-IDF) approach and Cosine Similarity to vectorize and compare the documents. The system also uses the k-nearest neighbor (k-NN) method, which displays the top five documents matching a query. This study aims to contribute to the improvement of academic research and writing for students and teachers in the College of Computing Studies (CCS). Creating an intuitive digital repository system with a document similarity function can increase the efficacy and efficiency of accessing relevant information and decrease the incidences of unintentional plagiarism.

Statement of the Problem

Every year, CCS students are required to complete their thesis before graduating. There are several steps to go through to complete it. Students need to submit three thesis titles; after choosing the title among the three titles, they will then undergo a proposal defense and, lastly, the final defense.

The lack of a centralized platform to organize, store and retrieve these studies led to several disadvantages for both students and teachers. It can be challenging for teachers to identify whether a proposed researches has already been done because it takes a lot of time to find relevant theses in a large amount of data available. It makes it difficult to manage academic resources effectively and efficiently. The lack of a digital repository might make maintaining and preserving academic documents challenging, increasing the chance of losing important data.

The absence of a system with document similarity features may cause unintended plagiarism, whereby students may unintentionally suggest existing studies without understanding them; this problem also leads to thesis duplication and redundancy.

Objectives

The study's main objective is to develop an online repository of thesis documents with similarity checking that aims to improve the process of checking thesis documents with unintentional plagiarism and encourage the originality of thesis ideas.
Specifically, the study will:

- Adapt the following algorithms or methods to implement the similarity feature of the online thesis repository system:
    - Term Frequency-Inverse Document Frequency (TF-IDF) method to vectorize the documents in the repository.
    - Cosine Similarity technique to compare the vectorized documents to the query.
    - K-nearest neighbor (k-NN) technique to present the five documents that are the most comparable to a query.
- Apply natural language processing techniques such as removing stopwords, changing words to lowercase, filtering out words with fewer than three characters, removing numerals, removing punctuation, and removing dates to enhance the system's performance.
- Develop the web application using Django and other Python libraries to allow users to manage, access, and use the similarity-checking feature.
- Evaluate the effectiveness of the techniques and methods used to implement the system.

Scope and Limitations

The web application provides features such as account registration, login, submission of thesis and checks its similarity results, accepting/rejecting documents, commenting, and document and account management.

The application's end users are students and teachers of the College of Computing Studies. Each end-user has different roles and can access features in the system based on the accounts they have.

There are also several limitations to the system. First, the system does not evaluate thesis proposals based on rubrics or criteria. Second, the submitted thesis document's title and contents are the only parts that are compared for similarities. Third, the system cannot verify each chapter individually; it can only check the entire document file and title. The algorithm used for text similarity was limited in its ability to account for the synonyms of words. It also doesn't display the similarity percentage of the specific words/paragraphs. Finally, the document similarity feature does not compare texts outside of the corpus; it only compares documents that have been added to the repository, and it can't identify if AI wrote the manuscript.

Significance of the Study

Completing this study would help students and teachers of the College of Computing Studies regarding thesis document management and unintended plagiarism.

The system makes it simpler for students to manage their research projects by providing a single platform to access and manage their thesis projects. By comparing their work with existing studies, the system's document similarity feature can help students identify relevant research studies and avoid accidental plagiarism. This function reinforces academic integrity and encourages students to produce original work.

The method offers teachers a more effective and efficient approach to managing thesis work. By comparing research proposals with previous studies, they can quickly determine their importance and ensure that students submit original work. Also, the system makes it simpler for teachers to track students' progress, offer feedback, and assess their work.

CHAPTER II
REVIEW OF RELATED LITERATURE

Related Studies

A digital library is not like any other library found in the city or school; this type of library is situated in a building or an infrastructure. The digital library is located and built digitally **based on the name itself.** Anders Björklund described digital as electronic technology that generates, stores, and processes data. He also deduced that digital can be online and offline [1].

The use of these online repositories has increased during the past few years. As a result of technological advancements and the internet, an increasing number of colleges and academic institutions are switching to digital platforms for archiving and disseminating research results. Many of these repositories are accessible to the general public, providing scholars worldwide with access to a wealth of intellectual material.

According to Pinfield et al., research data management is a significant challenge for organizations. A great deal of born-digital data is being produced in various forms rapidly in universities [2]. Singh looks at how open-access repositories have grown in India. His findings suggest that higher education and research organizations continue to follow trends toward creating open-access repositories [3].

With the growing popularity of digital libraries among organizations, it's evident that it has several benefits and drawbacks. LISBDNETWORK discussed the advantages and disadvantages of the digital library in the article. The ability to access information from anywhere at any time, the ability to store a vast amount of material in a small space, and the simplicity of sharing resources with other libraries are just a few of the advantages digital libraries have over traditional libraries. However, there are also several drawbacks, including copyright issues, slow Internet speeds, high upfront infrastructure costs, difficulties in discovering specific content, and difficulties in simulating a typical library environment. Despite these limitations, it is clear that digital libraries remain an essential tool for modern researchers and information seekers [4].

With many documents in an organization or in school, it's possible to duplicate them, intentionally or unintentionally. Document similarity is a technique used to gauge how similar two documents are. Document similarity or distance between documents is one of the main areas of information retrieval, according to Dmitriy Selivanor in his assessment of some of the approaches to document similarity [5]. In addition to information retrieval, document classification, document clustering, and many other text-analysis tasks depend on it. Text similarity is one of the active research and application subjects in natural language processing, according to Baeldung's article [6].

Term Frequency-Inverse Document Frequency

Term frequency-inverse document frequency, often known as TF-IDF, is a method for calculating and evaluating how essential a word is to a document among a group of papers, according to Stecanella and Scott [7,8].

This is a well-known method and is helpful in many ways, for example:

- Information retrieval

  Most search engines you have used to look up something utilize TF-IDF scores as part of their algorithm. To provide results that are most pertinent to your search, TF-IDF was created for document search.

- Keyword Extraction

  The automated technique of removing the most relevant words and expressions from text is keyword extraction. TF-IDF is useful in extracting keywords from the text. Identifying which words in a document have the highest score are the most relevant to that document.

Ganesan defines Term Frequency (TF) as the frequency with which a term or set of terms appears in a document. The most straightforward calculation is just to tally how many times each word appears. A term may be used more frequently based on the length of a text. We cannot infer that longer documents are more important than shorter ones, even though a word may appear more frequently in longer texts. Due to this, term frequency is typically normalized by dividing it by the total number of terms in the document [9].
If TF is about how frequently the term appeared in a document, Inverse Document Frequency (IDF) is about the weight of terms that frequently appear within a *corpus* (collection of documents) [10].

According to Ganesan, the more frequently a term is used throughout documents, the lower its score. The word loses significance as the score drops. IDF is usually computed as follows:

$$IDF = log\frac{N}{DF_t}$$

Where N represents the overall number of documents in your text collection, DFt represents the number of documents that contain the term t, and t is any word from your lexicon [9].

Daniël Heres' study explores machine learning approaches to improve source code plagiarism detection. After comparing various tools, he created a brand new one called

InfiniteMonkey. Using machine learning and conventional information retrieval techniques; he compared data sets from two sources. He found that the problem was successfully solved by the n-gram model with tf-idf weighting and cosine similarity, which could also be applied to visualization [11].

In their article, Shahzad Qaiser and Ramsha Ali explore the shortcomings of the well-liked TF-IDF method for text analysis in the age of big data. They clarify that other better variants of the method have been developed by researchers, including Adaptive TF-IDF, which uses hill-climbing to increase efficiency, and a cross-language variant that uses statistical translation. The authors also suggest using genetic algorithms to enhance TF-IDF. However, they point out that this did not result in appreciable advancements. The article also mentions the use of PageRank by search engine goliaths like Google to deliver more pertinent results. The authors contend that TF-IDF can be improved upon by mixing it with other methods like Naive Bayes [12].

Kim and Gil developed a method to classify research documents into relevant categories depending on their closely related topics. They extracted crucial information from each article using TF-IDF with latent Dirichlet Allocation (LDA), then sorted the articles into topical groups using the K-means clustering technique. The researchers concluded that their classification systems, with the aid of high-performance computing approaches, can pre-classify research articles by keywords and subjects. The classified research articles will then be used to search the articles in users' fascinating searches. areas effectively and rapidly [13].

Cosine Similarity

Cosine Similarity is a metric used to evaluate the resemblance of two vectors. Cosine similarity is often used in natural language processing (NLP) and information retrieval. Faith Karabiber claimed that cosine similarity is the measurement of the similarity of the direction of the vectors ignoring the differences in their magnitude or scale [13].

The mathematical definition of cosine similarity is the vectors' dot product divided by their magnitude. For instance, the following formula can be used to determine how similar two vectors, A and B, are:

$$similarity(A, B) = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

where:

- θ is the angle between the vectors,
- $A \cdot B$ is dot product between A and B and calculated as

$$A \cdot B = A^T B = \sum_{i=1}^{n} A_i B_i = A_1 B_1 + A_2 B_2 + \ldots + A_n B_n$$

- $\|A\|$ represents the L2 norm or magnitude of the vector which is calculated as $\|A\| = \sqrt{A_1^2 + A_1^2 \ldots A_1^n}$.

The similarity might range from -1 to +1 in value. More cosine similarity is shown by larger cosine values, which are produced by smaller angles between vectors. For instance:

- The angle between two vectors with the same orientation is 0, and their cosine similarity is 1.
- The angle between perpendicular vectors is 90 degrees, and their cosine similarity is 0.
- The angle between opposing vectors is 180 degrees, and their cosine similarity is -1.

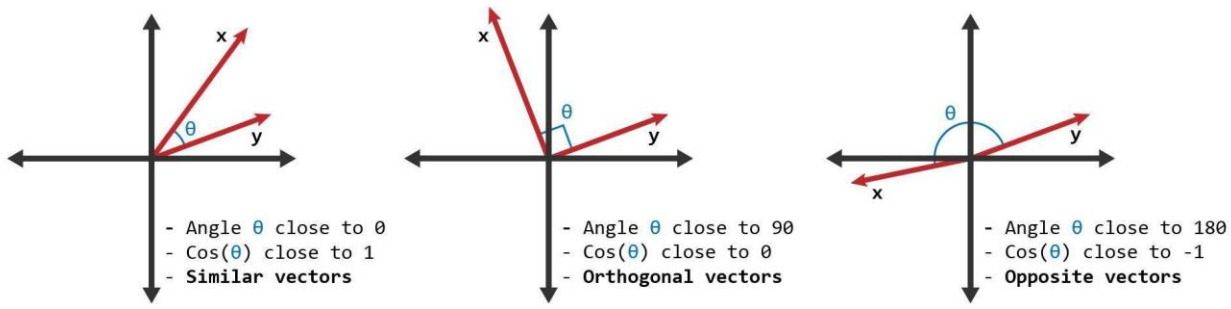


Figure 2. 1 Graphic Evaluation of two vectors with similarities close to 1, 0, and -1

Applications like data science and machine learning frequently use cosine similarity. Examples include measuring the similarity of documents, recommendation systems, and images [13].

Ryan M. Jones conducted a study to evaluate the effectiveness of cosine similarity in predicting the relationship between paired citing and cited sentences. The study found a word recognition of 65% and a performance accuracy of 4% using cosine similarity. It also used TF-IDF to calculate each radical phrase in its study. The total TF-IDF score was then standardized to account for different sentence lengths. The cosine similarity between each citing sentence and each sentence in the cited article was calculated using standardization. In addition, the study revealed five key issues that lower the similarity score, including anaphoric allusions and uneven use of affixes and acronyms [14].

K-Nearest Neighbor

The K-Nearest Neighbour is the best method for classification problems. It is also one of the most straightforward based on supervised learning techniques and machine learning algorithms [15]. The performance of k-NN on heterogeneous datasets, where data can be represented as a mixture of numerical and categorical features, is examined by Najat Ali, Daniel Neagu, and Paul Trundle. They employed a variety of metrics, such as single measurements (Euclidean and Manhattan) and various combinations of similarity metrics. Their tests' findings demonstrated that Euclidean distance is not a suitable metric for categorizing a heterogeneous data collection comprising numerical and binary features using k-NN. According to their

additional findings, combining the outcomes of numerical and binary similarity measures is a promising way to achieve better results than utilizing just one measure. They also noted that the results from the three examples of the supplied weights with k-NN are similar. This may indicate that the algorithm is somewhat resistant to the effects of compact heterogeneous features on classification performance [16].

Developing an Electronic Repository for Undergraduate Theses

According to Levy et al. [17], when theses were only delivered in print form, they frequently went unnoticed and were only recalled by the student and their advisor. The undergraduate dissertation may be available to the public. Still, print copies—often the only copies of theses—are usually kept in honor and are difficult, if not impossible, to get. In addition to saving time and money, students who publish their final projects electronically also, and perhaps most importantly, make their work available to potential collaborators and businesses throughout the world. Students can quickly look at examples of theses in research techniques seminars to understand their subject's formatting and writing styles.

Thesis Similarity Detection Application at Banten Jaya University

To identify whether a text or document is comparable to another, Kania, Solihati, & Arzaqi created an application based on a website [18]. The researcher employed a waterfall methodology, and the Winnowing Algorithm was used to determine how similar the texts were. The winnowing method is one of the identifying features of the document technique. This method may assess how similar a group of papers are to one another, considering even slight similarities, using the Jaccard Coefficient to obtain the percentage findings. Although the degree of similarity between text documents reduces with decreasing percentage levels of similarity, plagiarism can be detected if the percentage number is more significant.

Plagiarism Detection in Students' Theses Using the Cosine Similarity Method

Oppi Anda Resta, Aditya, A., and Febry Eka Purwiantono [19] decided to create a system that uses data mining to find similarities in titles, abstracts, or thesis themes to prevent plagiarism. In their work, the cosine similarity approach, preprocessing method, and TF-IDF are used to determine the degree of similarity between the title and abstract of a student's final scientific paper. Based on the threshold value, the results are presented and contrasted with the already-existing final project repository to decide whether scientific work can be approved or denied. According to the test data and training data utilized in the TF-IDF technique in this study, the level of similarity between the training data document and the test data document is 8%. This result shows that the student's thesis is still considered authentic and devoid of plagiarism.

Theses and Capstone Projects Plagiarism Checker using Kolmogorov Complexity Algorithm

Del Rosario & Sareno [20] found that students frequently try to copy other people's work and rely on pre-written answers from the Internet to finish their assignments. To address this problem, the researcher created a plagiarism detector to register materials, enable users to access them, and determine how similar the two documents were. The developed system consists of three essential modules: Document Search, which allows users to browse records;

Document Registration, which enables administrators to add and manage the stored papers; and Document Comparison, which serves as the system's plagiarism detection component. One hundred respondents rated the developed system, assessed using the ISO 25010 software quality model for Product Quality. By any standard, the approach produced an "outstanding" mean of 4.70; this shows that the study's objectives were satisfied and achieved.

Development of the Online Repository of Theses and Dissertations of the University of Cebu – Graduate School Library (ucGSLIB)

Dinauanao's [21] goal was to design and create an online repository that adheres to the industry standards for database management, search queries, user application interfaces, and data integrity. He claimed that graduate student theses and dissertations are unavailable online at the University of Cebu (UC). The majority of library patrons need help locating these items. The objective of this study was to provide an online repository for UCGS theses and dissertations. Descriptive and quasi-experimental designs were combined. Information obtained through surveys, interviews, and content analysis was used to develop the system.

Design and Development of an Online Repository System for Thesis and Special Problem Manuscripts

According to Mesa [22], the University of the Philippines (UP) is experiencing problems with students submitting hardbound and digital versions of their theses and particular problem manuscripts due to plagiarism or missing manuscripts. This is due to the lack of a mechanism to safeguard and organize such documents. The researcher created a project in response to these problems that would be helpful to the school's administrators, its students, and other stakeholders. The technology would allow users to learn about the research projects carried out by UP alums. Based on the survey and interviews, the study presents a fantastic outcome.

Synthesis

The rapid advancement of technology and the Internet caused digital libraries' popularity. The digital library has several advantages over traditional libraries, such as flexible access, availability of information, the ability to store a vast amount of data with limited space, and its simplicity of sharing information. Aside from these advantages, there are also some drawbacks, including copyright issues, high upfront infrastructure costs, difficulties in terms of specific content filtering, simulation difficulties, and slow internet speeds. Document similarity is part of information retrieval ideas in tasks such as document clustering, text analysis, and document classification.

For information, the College of Computing Studies (CCS) used a web application to handle the document similarity issues of the school. Turnitin is an online plagiarism software used by many educators and students to check the similarity of their work. It compares the submitted paper to many academic papers, websites, and other sources stored in the database.

Various studies regarding document similarity or text analysis were conducted. However, some only used the title and abstract to identify or classify if the documents were similar. There was also some integrated document similarity feature to the digital library but not use the repository as a corpus to be compared in a query document. In this information, the introduced

projects still needed to be completed. With this information, the researcher used the popular TF-IDF method and Cosine Similarity in the applications' document similarity feature. The study's technique for classifying the five most similar documents to the query was the k-nearest neighbor (k-NN). The web application was successfully developed and tested and was positively received and used by the users.

Existing System	Web-based	Digital Repository	Document Similarity (Compared to the repository)	Document Management	Peer Review
Development of Thesis Repository System with Document Similarity Feature for the College of Computing Studies	√	√	√	√	√
Prepostseo Plagiarism comparison	√	×	√	×	×
Turnitin Similarity Checker	√	√	√	√	×

Table 2. 1 System Comparison Table

Table 2.1 above compares three different but related and existing systems with the features listed above.

Prepostseo is similar to the two listed systems regarding web applications and document similarity features.

Turnitin and the **developed system** of the researcher have only the peer review as the difference. Turnitin doesn't have a feature wherein the panelists can accept or reject the uploaded documents of the students.

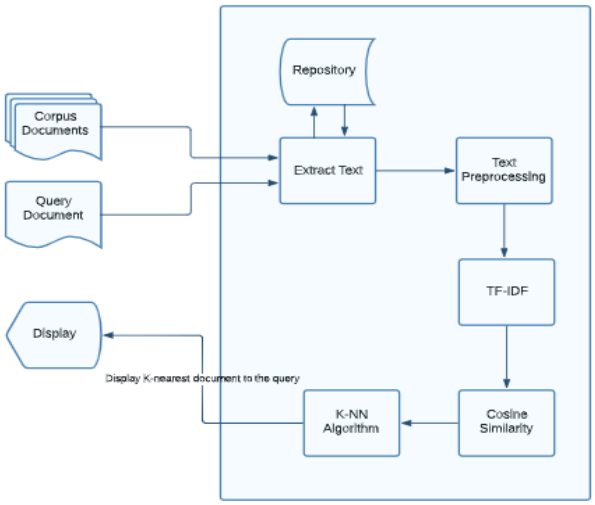


Figure 2. 2 Conceptual Framework

Figure 2.2 shows the conceptual framework of the study. The figure above presents the basic flow of the system. It shows the documents as the expected input and displays k nearest documents to the query.

Definition of Terms

Terms	Definitions
1. College of Computing Studies (CCS)	The college where the study is focus, located in Zamboanga City, Philippines
2. K-nearest neighbor (k-NN) algorithm	A machine learning algorithm that is used for classification and regression analysis. It assigns a class label to an instance by looking at the k-nearest neighbors of that instance and choosing the most common class label among those neighbors.
3. Cosine Similarity	A measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. It is used in information retrieval, text mining and natural language processing to determine the similarity between two documents or sets of words.
4. Digital Repository	A digital library that is used to store, preserve and manage digital content, such as books, articles, images, and audio/video files, for long-term access and use.
5. Term frequency-inverse document	A numerical statistic that is used to reflect how important a word is to a document in a collection or corpus. It is calculated as the product of the term frequency (TF) and the inverse document

frequency (TF-IDF)	frequency (IDF).
6. Classification	A type of machine learning algorithm that is used to categorize items into different classes or categories. It is used to predict the class label of a given instance based on the values of its features.
7. Natural Language Processing (NLP)	A field of study that focuses on the interaction between computers and human (natural) languages, in order to enable the computers to understand, interpret, and generate human language content.
8. Document Similarity	A measure of how similar two or more documents are to each other, based on their content and structure. It is used in information retrieval, text mining and natural language processing to determine the similarity between two or more documents.
9. Threshold	A value that is used to make decisions or take actions based on some criteria. In the context of machine learning, a threshold is used to determine when a prediction should be classified as positive or negative, based on some pre-defined limit.

Table 2. 2 Definition of Terms

CHAPTER III
METHODOLOGY

Research Design

An applied research design was used in this study. The research aims to solve a specific practical problem: the need for an efficient digital repository system for theses in an academic setting. The research objectives are clearly defined, and specific methods and techniques are outlined for achieving these objectives, including natural language processing and document similarity methods. The ultimate goal is to create a digital repository system with document similarity features to improve access to information and reduce the incidence of unintentional plagiarism.

Respondents

The College of Computing Studies' fourth- and fifth-year computer science students currently enrolled in the thesis subject served as the study's primary respondents. The end users of this developed system were students and teachers as the panelists. Random sampling was used to choose the sample for this investigation. For this study, a total of 15 individuals are chosen.

Research Instruments

A self-administered questionnaire online through Google Forms was used to evaluate the developed system. Google Forms is a free online tool to design and share surveys and quizzes quickly. The researcher decided to use Google Forms since it is practical, user-friendly, and makes data management and analysis simple. The data was gathered anonymously to protect the participants' confidentiality and privacy.

These are the overview of the self-administered questionnaire, which can be found in *Appendix A*:

- **Section 1**: Information about the survey questionnaire, including the purpose of why the study is being conducted, along with the respondent's consent to participate in the survey.
- **Section 2**: Personal information of the respondents, which includes the name(optional), college, course, and year level.
- **Section 3**: Direction on how to evaluate the software. This is the start of the first factor of the system, the design.
- **Section 4**: Questions about the functionality aspect of the system; this factor consists of the major and minor functions.
- **Section 5**: The system's capacity to carry out its planned functions without interruptions or breakdowns is called reliability. The availability, accessibility, uptime, and stability of the system are all expressly mentioned.
- **Section 6**: The usability factor evaluates how simple and effective a system is to use for the purpose for which it was designed.
- **Section 7**: The efficiency factor determines the effectiveness and speed with which users can execute activities.
- **Section 8**: This part caters to the comments or recommendations of the end-users.

Rating	Verbal Interpretation
5	Strongly Agree
4	Agree
3	Slightly Agree
2	Slightly Disagree
1	Strongly Disagree

Table 3. 1 Five-Point Likert Scale

The developed system was assessed using a five-point Likert scale. Researchers utilize the Likert scale, a unidimensional scale, to gather respondents' views and opinions. Researchers frequently use this psychometric scale to learn how people feel about a particular brand, item, or target market.

Validity of the Instrument

The survey questionnaire was submitted and reviewed by the research adviser before conducting the survey. The final revision was made after verifying the contents and recommendations of the research adviser.

Data Gathering Procedures

The researcher gathered the thesis documents of the past students of CCS online. These documents were used as a corpus and made accessible to the students through the system repository feature.

The documents gathered were first converted to text files before being saved in the database for further preprocessing for similarity checking. The researcher also utilized the built website to save the uploaded documents of the students. In doing this step, it will help more in expanding the corpus of documents that would be used to be compared to the query.

The researcher also used the survey discussed above after the respondents utilized the system. This step was conducted first for them to rate the system well based on their experienced.

Statistical Treatment

The following are the statistical tools used by the researcher in this study:

- **Mean** – this represents central tendency and shows what the data collection's average value is. It was determined by summing up the dataset's values and dividing the result

by the total number of values. This gave a comprehensive comprehension of the whole set of facts.

- **Median** – this is the center value of the data collection. When half of the values are above and half are below, it is represented by the median. The midpoint was used to calculate it after the data set was sorted from lowest to highest.
- **Mode** - this is particularly useful in identifying the study's most common response or outcome.
- **Frequency Distribution** – this was to show the frequency of responses per rating.
- **Range** – this was to measure the range used in interpreting the mean score. This is done by getting the difference between the maximum and minimum values found in the Likert scale.

Weight	Mean Range	Interpretation
5	4.20 – 5.00	Strongly Agree
4	3.40 – 4.19	Agree
3	2.60 – 3.39	Slightly Agree
2	1.80 – 2.59	Disagree
1	1.00 – 1.79	Strongly Disagree

Table 3. 2 Range and Equivalent Interpretation

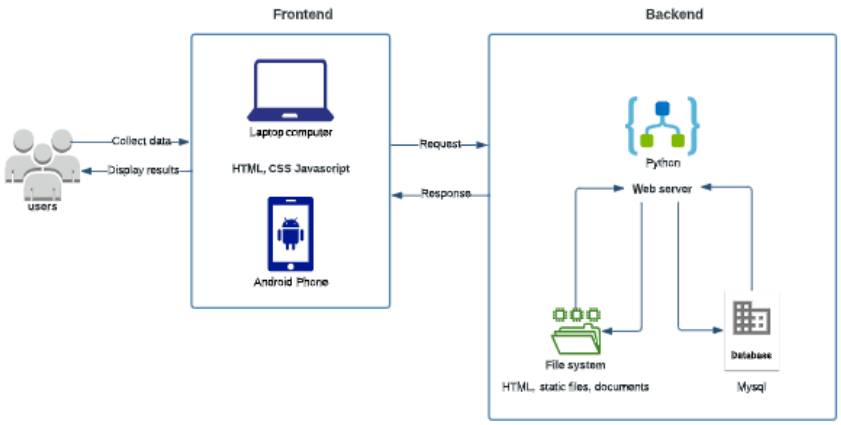


Figure 3. 1 System Architecture

The researcher included both frontend and backend elements in the system architecture. The frontend component, created with JavaScript, HTML, and CSS, offered a simple user interface for gathering and displaying results.

The backend component, which held the logic for data processing and information retrieval, was linked to the frontend component. A file system comprising HTML, CSS, pictures, and documents was merged with the Python-built backend component. The backend component was also linked to a MySQL database, which provided a place to store data and made it possible to retrieve it efficiently.

Software Development Lifecycle

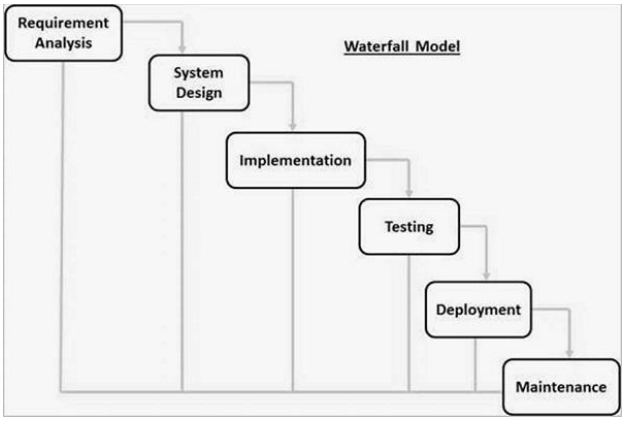


Figure 3. 2 Waterfall Model

Figure 3.2 presents the software development model the researcher used to develop the system. The researcher employed the waterfall model because it clearly outlines the project's structure, which aided in task organization and ensured that all necessary steps were taken.

Phase #1: Requirements Gathering: The gathering of requirements started by researching existing systems similar to the study. The researcher examined those systems' features, functionalities, and algorithms to identify their strengths and limitations. The researcher also studied the literature on the term frequency-inverse document frequency (TF-IDF) method, the cosine similarity metric, and the k-nearest neighbor (k-NN) algorithm to understand their applications in text analysis. After researching the systems and algorithms, the researcher determined the qualities the system needs to have. The researcher also identified the hardware and software requirements to run the system effectively.

Phase #2: Design and Planning: The researcher began designing the system after the requirements needed were finalized. At this phase, diagrams were developed to specify the connections between the system's entities and to map out the flow of each algorithm employed in the system. The researcher also created user interface prototypes to envision how the system would look and work.

Phase #3: Implementation: The researcher started the implementation step following the design and planning phase. The researcher used HTML, CSS, JavaScript, and Bootstrap to design and develop the web application's front end. For the backend, it was written in Python using Django framework.

The digital repository and the document similarity feature, the system's two key components, were created utilizing the algorithms mentioned in the objectives. The documents were vectorized using the term frequency-inverse document frequency (TF-IDF) approach, and the vectorized documents were compared using the cosine similarity measure. The five documents most similar to a query are displayed using the k-nearest neighbor (k-NN) technique. Ultimately, writing and testing the code, finding bugs, and improving the system's performance was all part of the implementation step.

Phase #4: Testing: The researcher carried out beta testing, which comprised assessing the built application's functionality, usability, reliability, and efficiency to make sure it was operating as anticipated. Various test scenarios were also conducted, including testing the document comparison capability with paraphrased and formatted documents and contrasting the outcomes with Turnitin, a well-known online plagiarism detection tool, and Prepostseo, another plagiarism detection program.

Phase #5: Deployment: The developed system was deployed online and was used by the end users.

Phase #6: Maintenance: The developed system was monitored for possible bugs and supported the end users.

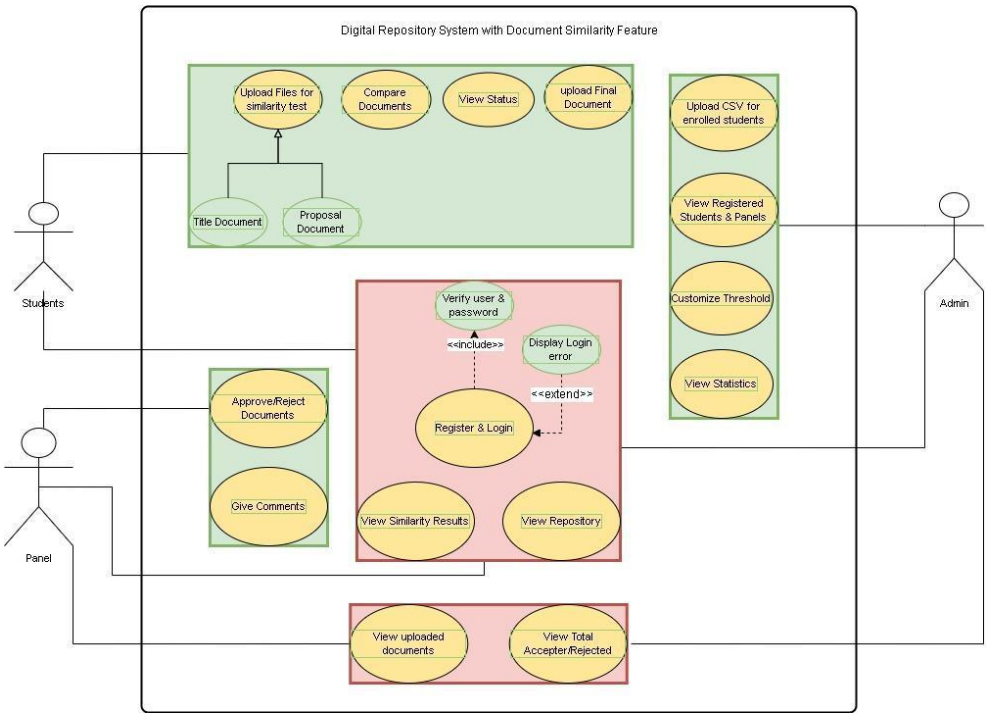


Figure 3. 3 Use Case Diagram

Figure 3.3 depicts how the system and the users interact with each other. It also shows the actions each user can do. The student User, Panel User, and Admin User are the three

primary actors in the use case diagram for the thesis repository with a document similarity checking system. The following are the functional requirements of each user:

The following actions can be carried by the student, panel, and admin user in the system:
Student:

- Register & Login: A student can register and log in to the system using their credentials.
- View repository: A student user can view the finished thesis's repository.
- Upload files for the similarity test: A student user can upload the title and proposal documents for the similarity test.
- Upload a final document: A student user can upload the final document to be added to the repository.
- Compare two documents: A student user can compare two documents and view the similarity report.
- View status: A student user can view the status of the uploaded document (accepted or rejected).

Panel:

- Register & Login: A panel user can log in to the system using their credentials.
- View repository: A panel user can view the finished thesis's repository.
- Approve/Reject document: A panel user can approve or reject the uploaded documents.
- Give comments: A panel user can give comments and feedback on the uploaded documents.
- View uploaded documents: A panel user can view the uploaded documents and their status.
- View total accepted/Rejected: A panel user can view the total number of accepted and rejected documents.

Admin:

- Register & Login: An admin user can register and log in to the system using their credentials.
- View repository: An admin user can view the finished thesis's repository.
- View uploaded documents: An admin user can view the uploaded documents and their status.
- View total accepted/Rejected: An admin user can view the total number of accepted and rejected documents.
- View registered students/panels: An admin user can view the list of registered students and panel users.
- View statistics: An admin user can view the system's statistics, such as the total number of uploaded documents, etc.
- Upload CSV or Excel file for enrolled students: An admin user can upload a CSV or Excel file containing the list of enrolled students. The file format can be seen in *Figure Appendix B.23. Excel format of Officially Enrolled Students*.
- Upload File for Repository: An admin user can upload a file to the repository.
- Customize similarity threshold: An admin user can customize the similarity threshold used for the similarity test.

Entity Relation Diagram

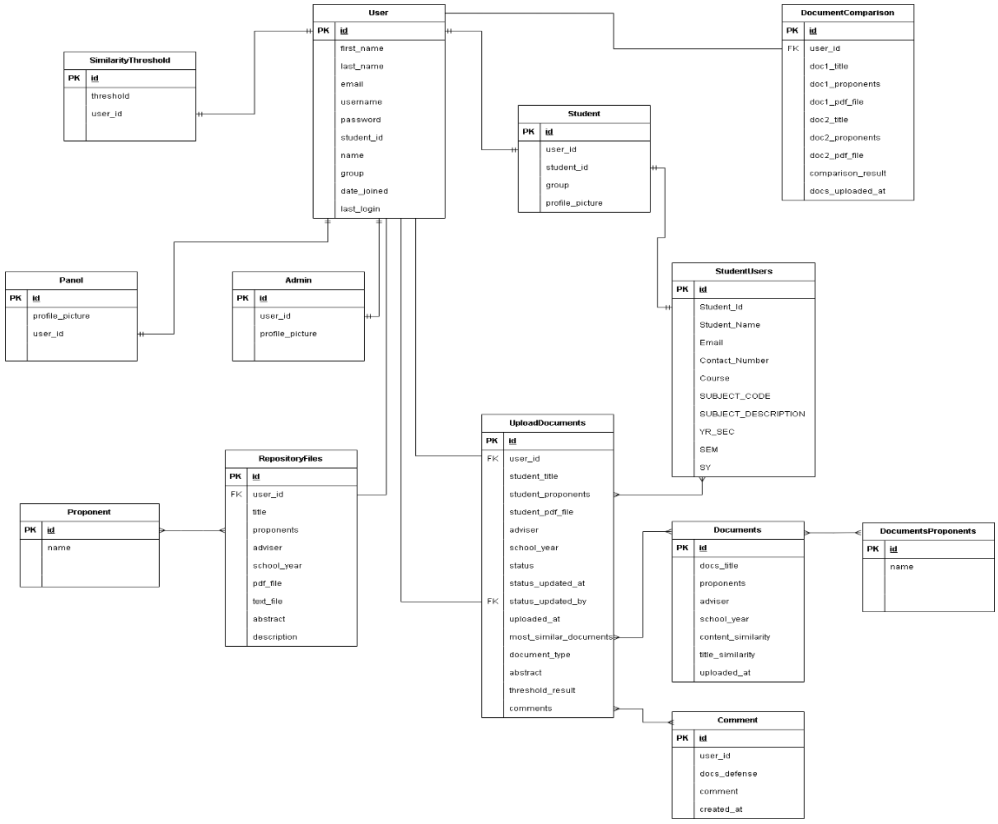


Figure 3. 4 System's ERD

Algorithms

In this study, the researcher used the widely known Term Frequency-Inverse Document Frequency (TF-IDF) for vectorizing the raw data and used Cosine Similarity to compare the vectorized data. These methods were used and incorporated into the document similarity feature of the system. Every document was preprocessed before undergoes in this process.

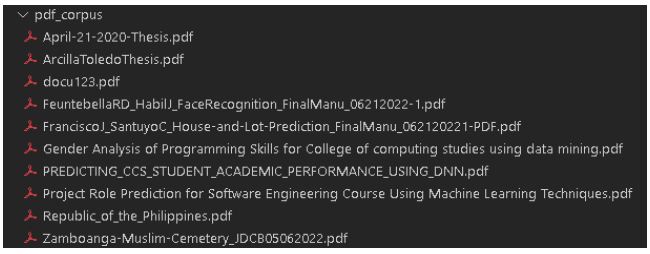


Figure 3. 5 Corpus of Documents

Figure 3.5 displays the documents gathered and used for training the tf-idf model. The total number of documents was 10, and these documents were gathered from the previous CCS students. Each document found in the corpus had different topics and title.

Document Preprocessing

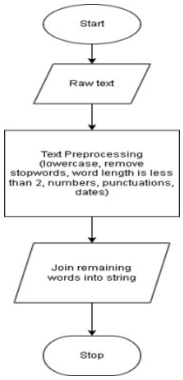


Figure 3. 6 Document Preprocessing Flowchart

Document preprocessing was the researcher's first step in preparing the gathered text data. This is an essential process because it was used to optimize and remove the noise of the documents. The following are the preprocessing steps that the researcher took:

- Case Folding

In this stage, the researcher changed all of the data's letters to lowercase. This is carried out to guarantee that similar words, such as "Hello" and "hello," are handled as one word in subsequent analysis.

- Digit and Punctuation Removal

Punctuation and numbers are removed since they have no bearing on text analysis and could make finding key phrases or recurring patterns more difficult.

- Stopwords Removal

Stopwords include "the" and "and" that has no vital context or meaning in a document. Moreover, words with a length of less than three characters were also removed.

- Date Removal

To further clean the text data, the system also removed the dates, which contain no important information.

Term Frequency – Inverse Document Frequency

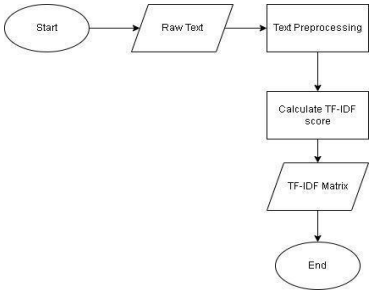


Figure 3. 7 TF-IDF Flowchart

Figure 3.7 shows the process after preparing the raw data for vectorization. The TF-IDF of each document was calculated and converted into a matrix of numerical data, where each row represents a document and each column describes a unique word in the corpus. A document-term matrix or bag-of-words matrix is the name of the resulting matrix. An element in the matrix represents the importance of each word in the matrix, and it is calculated using the inverse document frequency and the term frequency (the number of times the word appears in the document) (a measure of how common or rare the word is across all documents in the corpus).

```
  (0, 6573) 0.0012395558163642074
  (0, 3911) 0.0029162884713839008
  (0, 6473) 0.0029162884713839008
  (0, 8768) 0.0029162884713839008
  (0, 8319) 0.0010844648646925O5
  (0, 1313) 0.0010844648646925O5
  (0, 2922) 0.0010844648646925O5
  (0, 7705) 0.0012395558163642074
  (0, 5621) 0.0012395558163642074
  (0, 6530) 0.0009641669766879445
  (0, 7730) 0.0012395558163642074
  (0, 6628) 0.0012395558163642074
  (0, 3935) 0.0012395558163642074
  (0, 8643) 0.0008658764453647587
  (0, 1128) 0.0007107854936930S6
  (0, 6572) 0.0012395558163642074
  (0, 668)  0.0008658764453647587
  (0, 7271) 0.0008658764453647587
  (0, 6403) 0.001458144235691954
  (0, 7335) 0.005832576942767816
  (0, 5288) 0.005832576942767816
  (0, 4546) 0.002348318955734131
  (0, 4990) 0.016266972970387573
  (0, 3088) 0.005832576942767816
  (0, 7924) 0.010207009649843677
  : :
  (9, 4376) 0.13802583655455114
  (9, 2435) 0.06127900549494709
  (9, 8661) 0.029625857601952443
  (9, 1039) 0.27874486016382527
  (9, 6536) 0.0019552337694S7551
  (9, 7894) 0.0007374687992888185
  (9, 5870) 0.037704618766121 29
  (9, 1395) 0.012119341746253271
  (9, 8213) 0.0033664838184036864
  (9, 1894) 0.010099451455211 06
  (9, 9079) 0.008079561164168848
  (9, 7331) 0.009426154691530322
  (9, 2102) 0.008079561164168848
  (9, 2553) 0.0033664838184036864
  (9, 7909) 0.018179012619379906
  (9, 2109) 0.0033664838184036864
  (9, 1992) 0.004039780582084424
  (9, 8582) 0.0026931870547229494
  (9, 7803) 0.004713077345765161
  (9, 5417) 0.0026931870547229494
  (9, 8921) 0.0026931870547229494
  (9, 6214) 0.0026931870547229494
  (9, 8239) 0.667237092807610 7
  (9, 5833) 0.27874486016382527
  (9, 6992) 0.0013465935273614747
```

Figure 3. 8 TF-IDF Matrix

Cosine Similarity

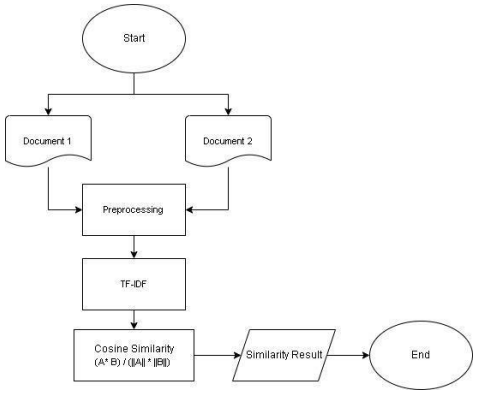


Figure 3. 9 Cosine Similarity Flowchart

Figure 3.9 depicts the system's flow while incorporating the cosine similarity to compare the vectorized documents.

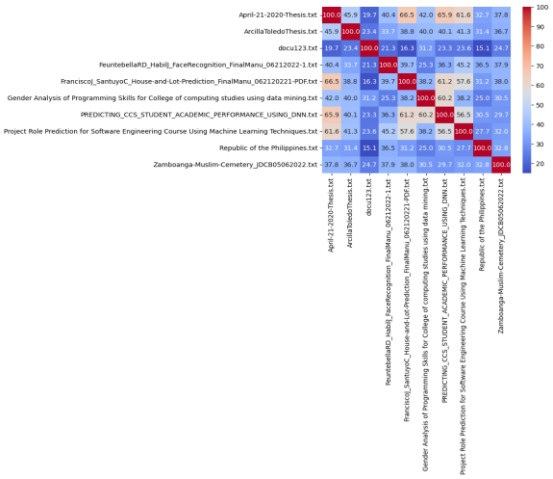


Figure 3. 10 Cosine Similarity Results

Figure 3.10 shows the result when the documents in the corpus were compared to each other. It shows the different similarities between each document. When the document was compared to itself, it offered a 100 percent which showed as dark red.

K-Nearest Neighbor (K-NN)

This study used the k-NN technique to classify the nearest documents to a query. The idea was to use the closest data points to a given query point in a vector space.

	April-21-2020-Thesis.txt
April-21-2020-Thesis.txt	100.000000
ArcillaToledoThesis.txt	45.909885
docu123.txt	19.692215
FeuntebellaRD_HabilJ_FaceRecognition_FinalManu_...	48.409930
FranciscoJ_SantuyoC_House-and-Lot-Prediction_Fi...	66.494816
Gender Analysis of Programming Skills for Colle...	41.959655
PREDICTING_CCS_STUDENT_ACADEMIC_PERFORMANCE_USI...	65.850096
Project Role Prediction for Software Engineerin...	61.619767
Republic of the Philippines.txt	32.734111
Zamboanga-Muslim-Cemetery_JDCB05062022.txt	37.826300

Figure 3. 11 Document Similarity Results

Using the document named “April-21-2020-Thesis.txt” as a query document, it was compared to the corpus of documents. Figure 3.11 shows the results of the document similarity against the corpus of documents. The highest similarity score was 100% when compared to itself and the lowest was docu123.txt, with a similarity score of 19.69%.

	document	percentage
0	April-21-2020-Thesis.txt	100.000000
4	FranciscoJ_SantuyoC_House-and-Lot-Prediction_F...	66.494816
6	PREDICTING_CCS_STUDENT_ACADEMIC_PERFORMANCE_US...	65.850096
7	Project Role Prediction for Software Engineeri...	61.619767
1	ArcillaToledoThesis.txt	45.909885

Figure 3. 12 K-NN Results

Figure 3.12 displays the five nearest documents based on it's similarity score. The k-NN technique was used in this phase to output the 5 similar documents to the query. The result was displayed in descending order from a 100% similarity score to 45%.

Development

The plugins listed below were installed in a virtual environment where the software was created using Visual Studio Code IDE. The system was then tested on a local server before the researchers uploaded the web application to the Railway hosting service with the domain name "https://web-production-fb5b.up.railway.app/".

Software Requirements

The software requirements for the user depend on the internet browser to access the web application, as it was deployed online.

The packages used for the development and deployment of the website are described in the requirements.txt in the workspace. The following are the packages used:

- asgiref==3.6.0
- boto3==1.26.59
- botocore==1.29.59
- click==8.1.3
- colorama==0.4.6
- DateTime==5.0

- Django==4.1.5
- django-cors-headers==3.13.0
- django-crispy-forms==1.14.0
- django-storages==1.13.2
- et-xmlfile==1.1.0
- fuzz==0.1.1
- fuzzywuzzy==0.18.0
- gunicorn==20.1.0
- jmespath==1.0.1
- joblib==1.2.0
- Levenshtein==0.20.9
- logging42==0.0.8
- loguru==0.6.0
- mysqlclient==2.1.1
- nltk==3.8.1
- numpy==1.24.1
- openpyxl==3.0.10
- pandas==1.5.3
- pathlib==1.0.1
- Pillow==9.4.0
- PyPDF2==3.0.1
- python-dateutil==2.8.2
- python-Levenshtein==0.20.9
- pytz==2022.7.1
- rapidfuzz==2.13.7
- regex==2022.10.31
- s3transfer==0.6.0
- scikit-learn==1.2.1
- scipy==1.10.0
- six==1.16.0
- sqlparse==0.4.3
- threadpoolctl==3.1.0
- tqdm==4.64.1
- tzdata==2022.7
- urllib3==1.26.14
- whitenoise==6.3.0
- win32-setctime==1.1.0
- zope.interface==5.5.2

Hardware Requirements

The following are the hardware requirements used in developing the web application:

- Processor: Intel(R) Celeron(R) N4100 CPU @ 1.10GHz, 1101 Mhz, 4 Core(s), 4 Logical Processor(s)

- Installed Physical Memory (RAM): 4.00 GB
- Storage: 230 GB

Development Tools

- Visual Studio Code: It is a multilingual programming IDE. Microsoft has created flexible software that supports a range of plugins, packages, and languages. This IDE is robust, especially for web developers, as it supports HTML, CSS, PHP, and JavaScript.

Front End Frameworks

- Bootstrap: Bootstrap is a free, open-source front-end framework for designing responsive and mobile-first websites. Twitter developed it, and was first released in 2011. Bootstrap is built with HTML, CSS, and JavaScript and is one of the most popular front-end frameworks web developers use today.

Back End Frameworks

- Django: Django is a high-level Python web framework that enables the rapid development of secure and maintainable websites.
- MySQL: MySQL is a robust and reliable database that developers and organizations have widely adopted due to its ease of use, scalability, and strong community support. It can handle large amounts of data and provides a range of features for data management and retrieval, making it an ideal choice for many web-based applications.

Implementation Plan

The researcher used a hosting platform to deploy the developed system to make the end users test the system and gather thoughts and feedback.

CHAPTER IV
RESULTS AND DISCUSSION

In this chapter, the researcher conducted different tests and results to evaluate the study's objectives. Figures were used to present the results of the testing.

Algorithms or Methods Assessment

To assess if the methods of the systems document similarity feature were successfully adapted, the researcher conducted a test using a document as a test case. The query document for the testing was entitled "Gravekeeper: Zamboanga Muslim Cemetery Web-Based Geo Mapping System employing Quantum Geospatial Information System." The document's title and abstract were used and compared to the corpus of documents. There were two versions of the query document, the original version of the title and abstract and the paraphrased version.

"Gravekeeper": Zamboanga Muslim Cemetery Web-Based GeoMapping System using Quantum Geographic Information System

Abstract

Over the past year's cemeteries are still constantly growing. With the cemetery still using traditional method of storing the deceased record, management cannot keep up in tracking the location of the deceased, and cannot properly maintain records and archives. This research then intends to develop an information management system for proper storing of data with integration of a quantum geographic information system for grave mapping and location tracking using spatial algorithm. The fully functional system would help cemetery caretaker or management, visitors, and researcher alike in making cemetery manageable. The study employed an applied research design, specifically a research and development design approach. Pre-evaluation and Beta-testing was made before and after the development, with beta-testing as a primary basis for the success of the system development. Data were collected using a survey questionnaire on 20 convenient residents near the target cemetery. Results showed that the developed system is beneficial in improving the cemetery management. Areas that mainly benefited from the system are grave tracking, data saving and record retrieving efficiency, and services options. Results also showed the openness of the cemetery visitors and caretaker in adapting a new tool in making cemetery visit less problematic. Furthermore, results made it clear that the lack of proper system management tool is what keeps the cemetery from being manageable. Therefore, further development in management tool is recommended in making the cemetery more manageable. Specifically, using a geographic information system technology. Keywords: Information Management System, Applied Research, Research and Development Design, Grave Tracking, Retrieving Efficiency, QGIS, Quantum Geographic Information System, Spatial Algorithm

Figure 4. 1 Query Document Original Version

Figure 4.1 depicts the original version of the query document. Using the process discussed in the last chapter, the original version was cleaned, vectorized, and compared to the corpus of documents.

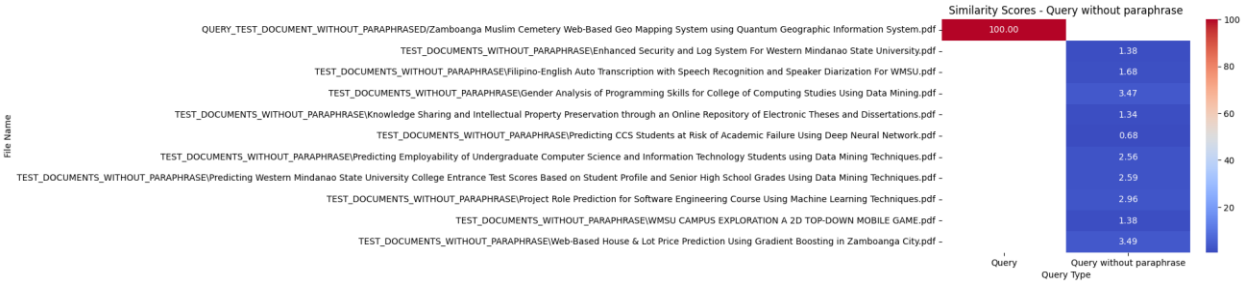


Figure 4. 2 Similarity Results of Query Document Original Version

Figure 4.2 displays the document similarity test results using the query document's original version. The following is the breakdown of the results:

- Enhanced Security and Log System For Western Mindanao State University: **1.37**

  The low similarity score of 1.37% between this document and the query indicates that it might not be relevant to the query.

- Filipino-English Auto Transcription with Speech Recognition and Speaker Diarization For WMSU: **1.67**

  Also, this document's low similarity score of 1.6797 with the query suggests that it might need to be more relevant to the query.

- Gender Analysis of Programming Skills for College of Computing Studies Using Data Mining: **3.46**

  The document may be related to the query, given that it has a relatively high similarity score of 3.46%.

- Knowledge Sharing and Intellectual Property Preservation through an Online Repository of Electronic Theses and Dissertations: **1.34**

  This document may not be related to the query document as it has a low similarity score of 1.34%.

- Predicting CCS Students at Risk of Academic Failure Using Deep Neural Network: **0.68**

  This document is not highly relevant to the question, as evidenced by its extremely low similarity score of 0.68%.

- Predicting Employability of Undergraduate Computer Science and Information Technology Students using Data Mining Techniques: **2.56**

This document may be pertinent to the query given that it has a relatively high similarity score of 2.56%.

- Predicting Western Mindanao State University College Entrance Test Scores Based on Student Profile and Senior High School Grades Using Data Mining Techniques: **2.58**

  This document's reasonably high similarity score of 2.58% indicates that it might be related to the query.

- Project Role Prediction for Software Engineering Course Using Machine Learning Techniques: **2.95**

  This document may be highly pertinent to the query, given its high similarity score of 2.95%.

- Web-Based House & Lot Price Prediction Using Gradient Boosting in Zamboanga City: **3.49**

  This document may be highly pertinent to the question, given its high similarity score of 3.49%.

- WMSU CAMPUS EXPLORATION: A 2D TOP-DOWN MOBILE GAME: **1.38**

  The similarity score for this document is only 1.38%. Like other documents, it may not be related to the query document.

```
Similarity scores for Query without paraphrase vs. 10 documents:
THE 5 MOST SIMILAR DOCUMENTS TO THE QUERY
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Web-Based House & Lot Price Prediction Using Gradient Boosting in Zamboanga City.pdf: 3.492
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Gender Analysis of Programming Skills for College of Computing Studies Using Data Mining.pdf: 3.465
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Project Role Prediction for Software Engineering Course Using Machine Learning Techniques.pdf: 2.955
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Predicting Western Mindanao State University College Entrance Test Scores Based on Student Profile and
Senior High School Grades Using Data Mining Techniques.pdf: 2.588
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Predicting Employability of Undergraduate Computer Science and Information Technology Students using Data
Mining Techniques.pdf: 2.565
```

Figure 4. 3 Top Five Most Similar Documents to the Query Original Document

As shown in the figure above, the five most similar documents to the query are displayed.

- The paper with the highest similarity score of 3.492, Web-Based House & Lot Price Prediction Using Gradient Boosting in Zamboanga City, is the most similar to the search query of the ten documents. The document's content might be connected to the query's subject, as evidenced by the high similarity score.

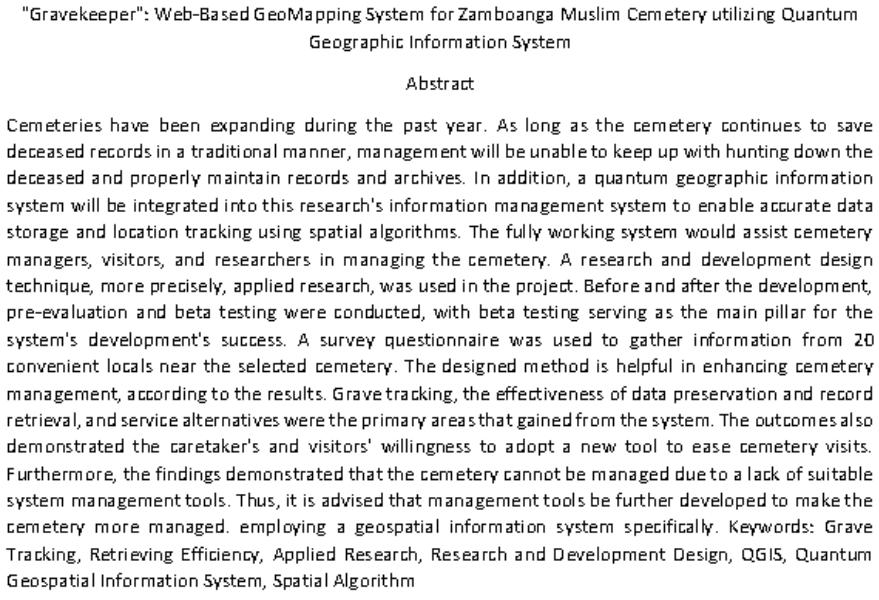
- This paper has a similarity score of 3.465, which is relatively high. Gender Analysis of Programming Skills for College of Computing Studies Using Data Mining. It might have

information or concepts pertinent to the inquiry, such as gender analysis or data mining methods.

- With a similarity score of 2.955, Project Role Prediction for Software Engineering Course Using Machine Learning Techniques is less related to the query than the first two documents. It still has a high similarity score and might include details or concepts relevant to the query, like machine learning strategies.

- The document Predicting Western Mindanao State University College Entrance Test Scores Based on Student Profile and Senior High School Grades Using Data Mining Techniques has a similarity score of 2.588, which is lower than the scores of the first two documents but higher than the scores of the last document in the list. It could include details or concepts about the question, like data mining methods or methods for predicting student performance.

- Predicting Employability of Undergraduate Computer Science and Information Technology Students Using Data Mining Techniques. This document's similarity score is 2.565, just a little lower than that of the preceding document. It might have information or concepts pertinent to the question, like data mining methods or employability prediction.



"Gravekeeper": Web-Based GeoMapping System for Zamboanga Muslim Cemetery utilizing Quantum Geographic Information System

Abstract

Cemeteries have been expanding during the past year. As long as the cemetery continues to save deceased records in a traditional manner, management will be unable to keep up with hunting down the deceased and properly maintain records and archives. In addition, a quantum geographic information system will be integrated into this research's information management system to enable accurate data storage and location tracking using spatial algorithms. The fully working system would assist cemetery managers, visitors, and researchers in managing the cemetery. A research and development design technique, more precisely, applied research, was used in the project. Before and after the development, pre-evaluation and beta testing were conducted, with beta testing serving as the main pillar for the system's development's success. A survey questionnaire was used to gather information from 20 convenient locals near the selected cemetery. The designed method is helpful in enhancing cemetery management, according to the results. Grave tracking, the effectiveness of data preservation and record retrieval, and service alternatives were the primary areas that gained from the system. The outcomes also demonstrated the caretaker's and visitors' willingness to adopt a new tool to ease cemetery visits. Furthermore, the findings demonstrated that the cemetery cannot be managed due to a lack of suitable system management tools. Thus, it is advised that management tools be further developed to make the cemetery more managed. employing a geospatial information system specifically. Keywords: Grave Tracking, Retrieving Efficiency, Applied Research, Research and Development Design, QGIS, Quantum Geospatial Information System, Spatial Algorithm

Figure 4. 4 Query Document Paraphrased Version

Figure 4.4 shows the paraphrased version of the original document. Quilbot.com was used to rephrase the query document. Using the same corpus of documents, the translated version was then compared.

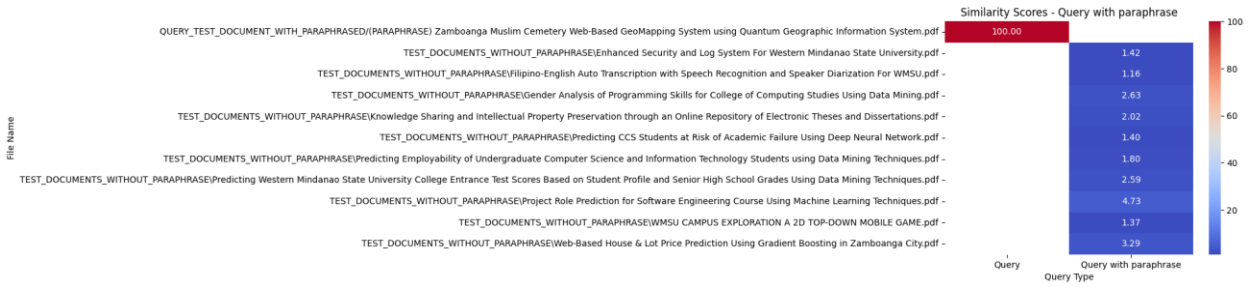


Figure 4. 5 Similarity Results of Query Document Paraphrased Version

The breakdown of the document similarity results using the paraphrased version of the query document was the following:

- Enhanced Security and Log System For Western Mindanao State University: **1.420**

  The similarity score is relatively low, indicating that this document and the query document are incomparable.

- Filipino-English Auto Transcription with Speech Recognition and Speaker Diarization For WMSU: **1.159**

  This document's low similarity score suggests it is irrelevant to the query.

- Gender Analysis of Programming Skills for College of Computing Studies Using Data Mining: **2.627**

  This document's average similarity score indicates it relates to the query.

- Knowledge Sharing and Intellectual Property Preservation through an Online Repository of Electronic Theses and Dissertations: **2.020**

  This document's average similarity score may relate to the query.

- Predicting CCS Students at Risk of Academic Failure Using Deep Neural Network: **1.396**

  This document's low similarity score indicates that it is irrelevant to the query.

- Predicting Employability of Undergraduate Computer Science and Information Technology Students using Data Mining Techniques: **1.803**

  It implies that the information is not pertinent to the query.

- Predicting Western Mindanao State University College Entrance Test Scores Based on Student Profile and Senior High School Grades Using Data Mining Techniques: **2.586**
  Using a paraphrased version of the query indicates that it is partly relevant to the query.

- Project Role Prediction for Software Engineering Course Using Machine Learning Techniques: **4.731**

  When a paraphrased version of the query is utilized, it indicates that it is the most pertinent document to the inquiry.

- Web-Based House & Lot Price Prediction Using Gradient Boosting in Zamboanga City: **3.292**

  When the query is paraphrased, it implies it is relevant to the inquiry.

- WMSU CAMPUS EXPLORATION A 2D TOP-DOWN MOBILE GAME: **1.373**

  When a paraphrased version of the query is utilized, this document's low similarity score indicates that it is not highly relevant to the query.

```
Similarity scores for Query with paraphrase vs. 10 documents:
THE 5 MOST SIMILAR DOCUMENTS TO THE QUERY
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Project Role Prediction for Software Engineering Course Using Machine Learning Techniques.pdf: 4.731
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Web-Based House & Lot Price Prediction Using Gradient Boosting in Zamboanga City.pdf: 3.292
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Gender Analysis of Programming Skills for College of Computing Studies Using Data Mining.pdf: 2.627
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Predicting Western Mindanao State University College Entrance Test Scores Based on Student Profile and
Senior High School Grades Using Data Mining Techniques.pdf: 2.586
TEST_DOCUMENTS_WITHOUT_PARAPHRASE\Knowledge Sharing and Intellectual Property Preservation through an Online Repository of Electronic Theses
and Dissertations.pdf: 2.020
```

Figure 4. 6 Top Five Most Similar Documents to the Query Paraphrased Document

Figure 4.6 displays the top five most similar documents to the original document's paraphrased version. The breakdown is as follows:

- The document with the most significant similarity score, 4.731, is Project Role Prediction for Software Engineering Course Using Machine Learning Techniques.pdf. It is pertinent to the question, indicating that it might know about utilizing machine learning techniques to forecast project responsibilities in software engineering courses.

- Web-Based House & Lot Price Prediction Using Gradient Boosting in Zamboanga City.pdf: With a similarity score of 3.292, this document is the second most similar among the test documents without paragraphs. Given that it uses gradient-boosting techniques to estimate prices, it might be related to the question of paraphrasing.

- It is possible that the content in Gender Analysis of Programming Skills for College of Computing Studies Using Data Mining.pdf, which has a similarity score of 2.627, pertains to data mining techniques used to analyze the programming abilities of students in the College of Computing Studies. It appears pertinent to the query, implying that the paper might include information on the gender gap in programming skills.

- The fourth most similar item is Forecasting Western Mindanao State University College Admission Exam Results Based on Student Profile and Senior High School Grades Using Data Mining Techniques.pdf, which has a similarity score of 2.586. It involves employing data mining to predict college entrance test scores so that they might be connected to the query.

- Knowledge Sharing and Intellectual Property Preservation through an Online Repository of Electronic Theses and Dissertations.pdf has a similarity score of 2.020, indicating that it may contain information on knowledge sharing and intellectual property preservation through such an electronic repository. Even though it may not be directly related to the question, it could include helpful information relevant to the current work.

Natural Language Processing Techniques Evaluation

ID	DOCUMENTS TITLE	RESULTS			
		WITHOUT PARAPHRASE WITHOUT PREPROCESSING	WITHOUT PARAPHRASE WITH PREPROCESSING	WITH PARAPHRASE WITHOUT PREPROCESSING	WITH PARAPHRASE WITH PREPROCESSING
1	Enhanced Security and Log System For Western Mindanao State University (WMSU) Using Real-Time Face Recognition System	100%	100%	76.67%	67.31%
2	Filipino-English Auto Transcription with Speech Recognition and Speaker Diarization For WMSU	100%	100%	92.92%	66.94%
3	Gender Analysis of Programming Skills for College of Computing Studies Using Data Mining	100%	100%	79.86%	72.16%
4	Knowledge Sharing and Intellectual Property	100%	100%	83.33%	70.66%

	Preservation through an Online Repository of Electronic Theses and Dissertations (ETD): The Crimson's Legacy				
5	Predicting CCS Students at Risk of Academic Failure Using Deep Neural Network	100%	100%	81.03%	72.88%
6	Predicting Employability of Undergraduate Computer Science and Information Technology Students using Data Mining Techniques	100%	100%	85.13%	63.23%
7	Predicting Western Mindanao State University College Entrance Test Scores Based on Student Profile and Senior High School Grades Using Data Mining Techniques	100%	100%	87.19%	76.32%
8	Project Role Prediction for Software Engineering Course Using Machine Learning Techniques	100%	100%	89.61%	84.21%
9	Web-Based House & Lot Price Prediction Using Gradient Boosting in Zamboanga City	100%	100%	88.41%	76.77%
10	WMSU CAMPUS EXPLORATION: A 2D TOP-DOWN	100%	100%	71.08%	56.64%

	MOBILE GAME				

Table 4. 1 Test Documents

Figure 4.1 displays the test documents with their equivalent results compared to themselves with/without paraphrasing and preprocessing. This testing assessed whether the NLP techniques discussed in the objectives were successfully integrated and effectively enhanced the system's performance in terms of its main feature, document similarity.

The first column shows the ID of each document, while the second column provides the titles of the documents. There are four columns displaying the results of the similarity checking using different conditions. The first two columns show the results obtained when no paraphrasing and no preprocessing were used, while the last two columns show the results obtained when paraphrasing and preprocessing were used. The results varied when paraphrasing and preprocessing were used. These results suggests that the use of preprocessing technique in preparing for document similarity was helpful in filtering and removing unnecessary data that doesn't contain important data for the document.

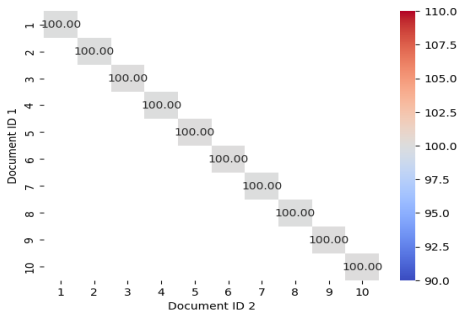


Figure 4. 7 WITHOUT PARAPHRASE AND WITHOUT PREPROCESSING RESULTS HEATMAP

Figure 4.7 presents the results when the document was compared to itself without paraphrasing and preprocessing. The results display that all ten documents have 100% similarity, which was understandable since the paper was the original text and was just compared to itself.

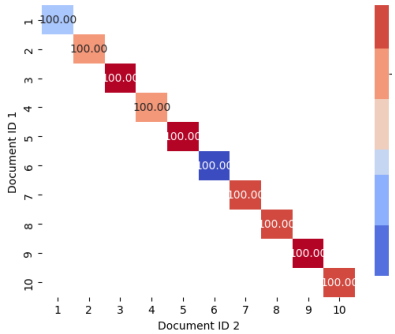


Figure 4. 8 WITHOUT PARAPHRASE AND WITH PREPROCESSING RESULTS HEATMAP

Figure 4.8 displays the similar results of the test documents when compared to itself without paraphrasing but using preprocessing methods. Similar to the results in Figure 4.7, all documents were 100% identical to itself. This means that the preprocessing methods didn't alter the document's content. Preprocessing methods were used to clean and normalize the text data before the similarity test.

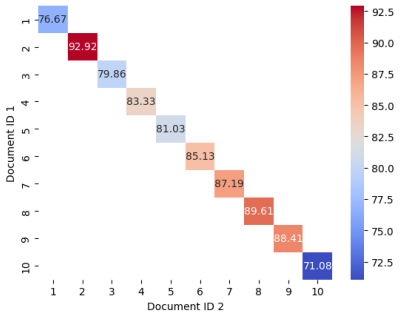


Figure 4. 9 WITH PARAPHRASE AND WITHOUT PREPROCESSING RESULTS HEATMAP

Figure 4.9 shows the results when the test documents were compared to a document with its paraphrased version but without preprocessing methods used. The similarity scores range from 71.09% to 92.92%. Depending on the organization's threshold, the interpretation of the results may be similar or not similar. While a score of 0% means that they have no similarity, a score ranging from 71% to 92% suggests that the paraphrased version of the documents is still similar or contains similar content to the original document.

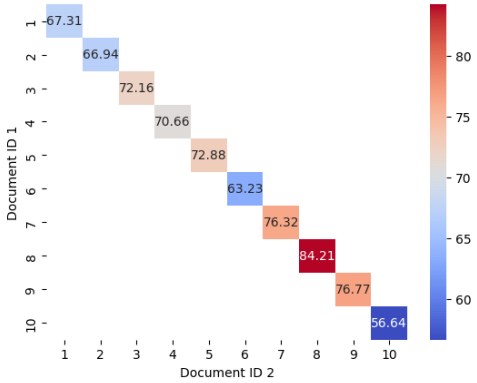


Figure 4. 10 WITH PARAPHRASE AND WITH PREPROCESSING RESULTS HEATMAP

Figure 4.10 presents its findings using the paraphrased version of the original document while using preprocessing techniques. The results display similarity scores ranging from 56.64% to 84.21%. Compared to the results obtained in Figure 4.9 without preprocessing methods, the similarity scores are generally lower, suggesting that preprocessing methods have affected the similarity scores of the documents.

The preprocessing techniques used were:

- Case folding or lowercasing.
- Filtering out words with at most three characters.
- Removing numerals.
- Punctuations.
- Removing dates and stopwords.

While the results in Figure 4.10 are lower than Figure 4.9, without using preprocessing methods, the documents still contained noise and unnecessary words, digits, and punctuations that don't have significance in a document.

Beta Testing and Software Evaluation

The researcher also conducted beta testing to evaluate the developed web application. A total of 15 responders participated in the testing, and everyone was a student from the College of Computing Studies. The respondents were first asked to utilize the application to test its functions. The researcher used a questionnaire with five categories: design, functionality, reliability, usability, and efficiency.

The five-point Likert scale was used to rate the application with a scale ranging from 1 to 5. The highest rating is five (5) with an interpretation of Strongly Agree, followed by four (4) with a variation of Agree. Three (3) have an understanding of Slightly Agree, two (2) Slightly Disagree, and one (1), which was the lowest rating, has an interpretation of Strongly Disagree.

Question	Strongly Disagree (1)	Slightly Disagree (2)	Slightly Agree (3)	Agree (4)	Strongly Agree (5)	Equivalent Total	Mean	Median	Mode	Interpretation
The system's overall layout is user-friendly and visually appealing.	0	1	3	8	3	58	3.86	4	4	Agree
The font is designed in an easy-to-read size and	0	0	4	1	10	66	4.4	5	5	Strongly Agree

style.										
The system's layout makes finding what I'm seeking for simple.	0	0	4	5	6	62	4.13	4	5	Agree
Total	0	1	11	14	19	186	4.13	4	5	Agree

Table 4. 2 Design Evaluation Results

Table 4.2 displays the respondents' evaluation in terms of the Design category of the application. This category contained three statements with a mean score ranging from 3.86 to 4.4 with an equivalent interpretation of Agree and Strongly Agree.

Overall, the equivalent total for all three questions was 186, with a mean score of 4.13, indicating that most respondents agreed with the statements regarding the system's design. These points were calculated by adding the total number of responses and the equivalent total score. These results imply that users valued and thought the system's design was efficient.

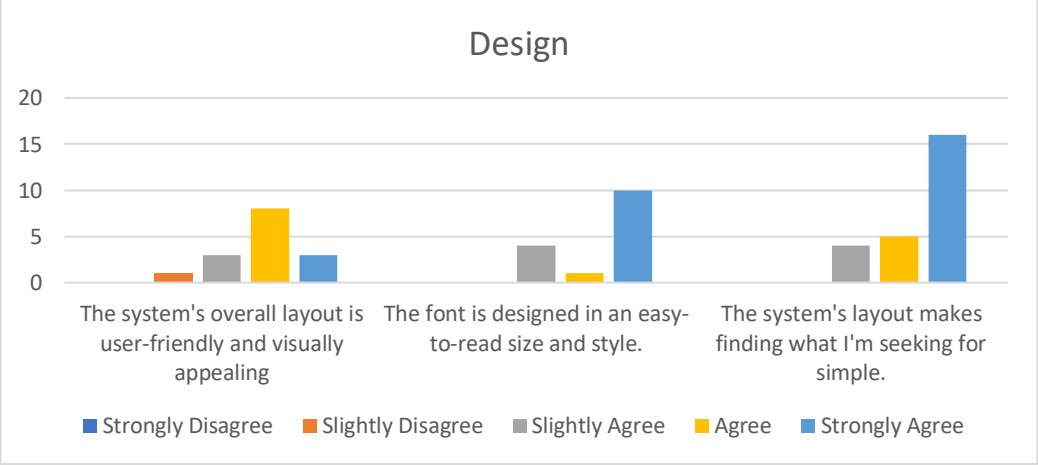


Figure 4. 11 Graphical Representation of Design Category Evaluation Results

Figure 4.11 displays the evaluation results' graphical representation in terms of the design category. Most respondents *Strongly Agree* that the systems' layout is simple, making what they're looking for easy. While one (1) respondent *Slightly Disagree* in terms of the system's overall layout being user-friendly and visually appealing.

Question	Strongly Disagree (1)	Slightly Disagree (2)	Slightly Agree (3)	Agree (4)	Strongly Agree (5)	Equivalent Total	Mean	Median	Mode	Interpretation
Main Function/s										
Digital Repository	0	0	2	6	7	65	4.33	4	5	Strongly Agree
Uploading Document and Perform Similarity Check										
Title Defense Document	0	0	2	8	5	63	4.20	4	4	Strongly Agree
Proposal Defense Document	0	0	2	7	6	64	4.26	4	4	Strongly Agree
Final Defense Document	0	1	3	4	7	62	4.13	4	5	Agree
Minor Function/s										
Retrieval of Document in Digital Repository	0	0	2	5	8	66	4.40	5	5	Strongly Agree
Uploaded Documents History	0	0	4	4	7	63	4.20	4	5	Strongly Agree
The system is able to perform all the functions it claims to do.	0	0	2	6	7	65	4.33	4	5	Strongly Agree
The document similarity feature of the system has acceptable accuracy and is a reliable tool for	0	0	2	7	6	64	4.26	4	4	Strongly Agree

comparing documents.										
The system operates smoothly without any major bugs or errors.	0	0	1	10	4	63	4.20	4	4	Strongly Agree
The system's performance speed is fast and responsive.	0	0	2	5	8	66	4.40	5	5	Strongly Agree
Total	0	1	22	62	65	631	4.20	4	5	Strongly Agree

Table 4. 3 Functionality Evaluation Results

Table 4.3 displays the respondents' evaluation results regarding the application's functionality. The functionality category contained ten statements discussing the application's functionality. The respondents tested the application before rating it. The application's main functions were assessed with a mean score ranging from 4.13 to 4.33, with an equivalent interpretation of *Agree* to *Strongly Agree*. While the minor parts' mean score was rated from 4.20 to 4.40, all with a variation of *Strongly Agree*.

The total mean score of the functionality category was 4.20, indicating that most respondents *Strongly Agree* with most of the statements stated in the group; this means that the system was well accepted and met most of the end users' expectations.

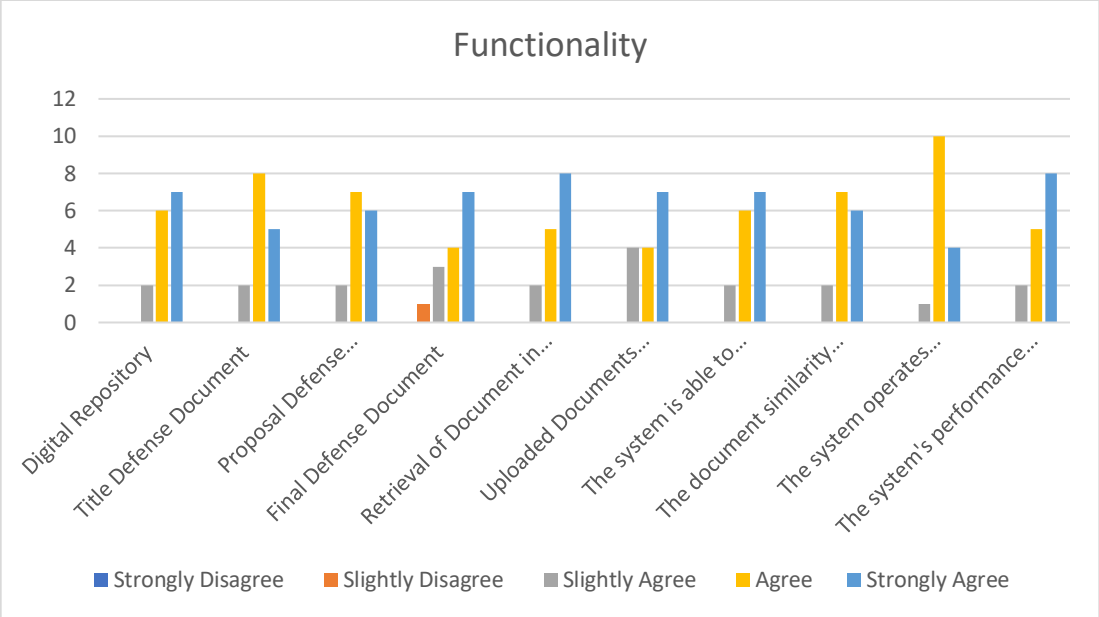


Figure 4. 12 Graphical Representation of Functionality Category Evaluation Results

Figure 4.12 shows the graphical representation of the functionality category regarding its evaluation results. As displayed in the figure above, most respondents either *Strongly Agree* or *Agree* with the statements regarding the application's functionality. While one of the respondents *Slightly Disagrees* with the statement regarding the Final Defense Document, this still means that most respondents met their expectations regarding the application's functionality.

Question	Strongly Disagree (1)	Slightly Disagree (2)	Slightly Agree (3)	Agree (4)	Strongly Agree (5)	Equivalent Total	Mean	Median	Mode	Interpretation
The system is available and accessible when I need it.	0	0	1	7	7	66	4.4	4	5	Strongly Agree
The system does not experience frequent downtime or crashes.	0	0	2	5	8	66	4.4	5	5	Strongly Agree

Total	0	0	3	12	15	132	4.9	4	5	Strongly Agree

Table 4. 4 Reliability Evaluation Results

Table 4.4 presents the evaluation results of the respondents regarding the reliability of the developed web application. This category contained two statements to evaluate if the application was reliable according to the end-user's standards. Each statement received a mean score of 4.4 with an equivalent interpretation of *Strongly Agree*.

A combined mean score of 4.9 indicates that most respondents *Strongly Agreed* that the application was reliable and consistent with its performance.

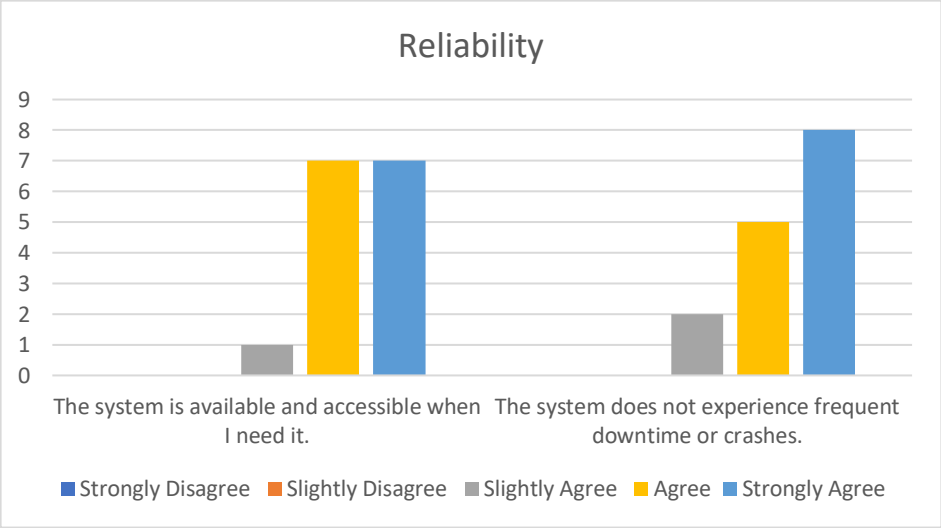


Figure 4. 13 Graphical Representation of Reliability Category Evaluation Results

Figure 4.13 shows the graphical representation of the evaluation results in terms of the reliability of the application. As displayed in the figure above, most responses regarding the system's reliability varied from *Slightly Agree* to *Strongly Agree*. This response means that the system was dependable, according to the respondents.

Question	Strongly Disagree (1)	Slightly Disagree (2)	Slightly Agree (3)	Agree (4)	Strongly Agree (5)	Equivalent Total	Mean	Median	Mode	Interpretation
The system's user interface is clear and simple to use.	0	0	3	4	8	65	4.33	5	5	Strongly Agree

Task completion and system navigation are simple.	0	0	2	5	8	66	4.4	5	5	Strongly Agree
The system's labels and instructions are brief and easy to understand.	0	0	1	5	9	68	4.53	5	5	Strongly Agree
Users receive beneficial feedback from the system.	0	0	1	10	4	61	4.06	4	4	Agree
Total	0	0	7	24	29	260	4.33	4	5	Strongly Agree

Table 4. 5 Usability Evaluation Results

Table 4.5 discussed the usability of the application, which contained a total of four statements. This category received a mean score ranging from 4.06 to 4.53, indicating that most respondents *Strongly Agreed* or *Agreed* with the systems' usability in terms of the academic setting.

The Usability category also received a total mean score of 4.33; this result shows that most respondents who participated in the beta testing *Strongly Agreed* with the system being simple, easy to use, and navigate.

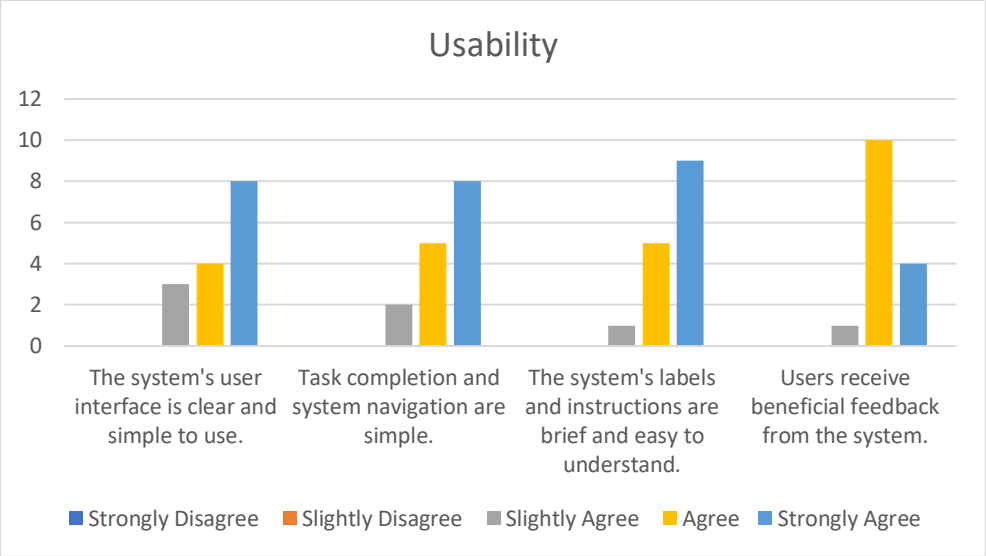


Figure 4. 14 Graphical Representation of Usability Category Evaluation Results

Figure 4.14 displays that the statement that got the most response regarding the rating *Agree* was about the users receiving valuable feedback from the system. And the rest of the statements received either *Agree* or *Strongly Agree* rating from the respondents.

Question	Strongly Disagree (1)	Slightly Disagree (2)	Slightly Agree (3)	Agree (4)	Strongly Agree (5)	Equivalent Total	Mean	Median	Mode	Interpretation
The system allows me to complete tasks quickly and efficiently.	0	0	1	6	8	67	4.46	5	5	Strongly Agree
The document similarity feature of the system saves me time in detecting unintentional plagiarism.	0	0	1	6	8	67	4.46	5	5	Strongly Agree
The system's search	0	0	3	4	8	65	4.33	5	5	Strongly Agree

function quickly finds relevant information .										
The system's performance speed is fast enough to meet my needs.	0	0	2	5	8	66	4.4	5	5	Strongly Agree
Total	0	0	7	21	32	265	5.41	5	5	Strongly Agree

Table 4. 6 Efficiency Evaluation Results

Table 4.6 shows the evaluation results in terms of the efficiency of the application. With a mean score ranging from 4.33 to 4.46, all statements regarding the efficiency of the application received a *Strongly Agree* interpretation based on its mean scores.

The total mean score received by this category was 5.41; this indicates that the respondents *Strongly Agreed* that the system helped them perform work swiftly and efficiently.

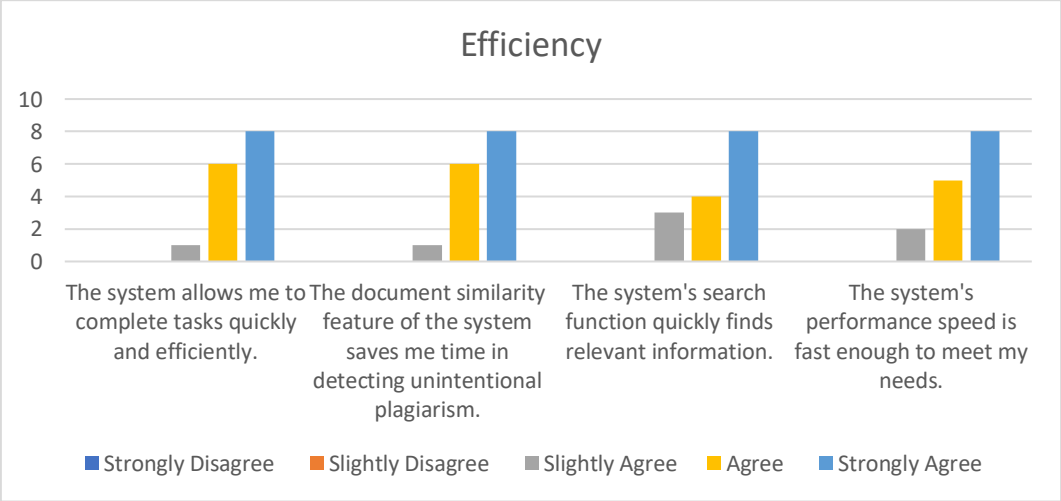


Figure 4. 15 Graphical Representation of Efficiency Category Evaluation Results

Figure 4.15 displays the efficiency category's graphical representation according to its evaluation results. The response shows that the statements received a rating varied from *Slightly Agree* to *Strongly Agree.*

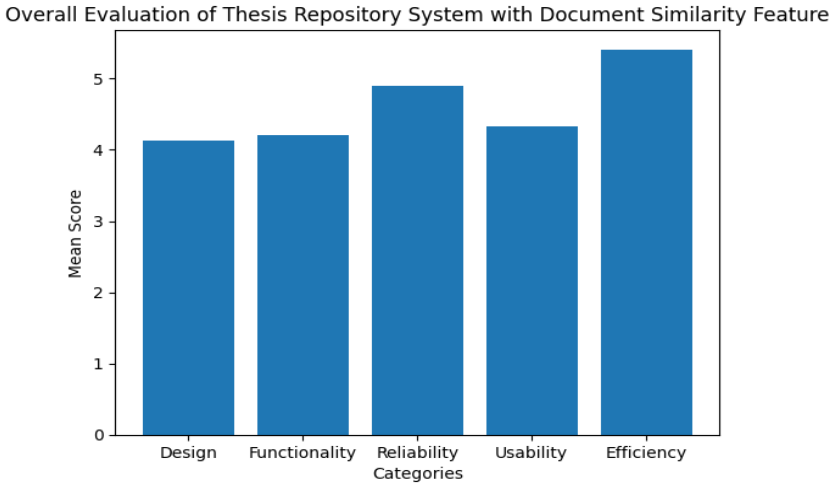


Figure 4. 16 Overall Evaluation of the System based on Mean Score

Figure 4.16 shows the overall evaluation of the system in terms of the mean score. Based on the mean score received from the respondents, the application was evaluated positively, with scores varying from 4.13 to 5.41. the interpretation of these scores indicates that the respondents mostly Agreed or Strongly Agreed with the application's design, functionality, reliability, usability, and efficiency.

Effectiveness of the techniques

The researcher assessed the effectiveness of the techniques and methods used to implement the system using two web applications with plagiarism-checking features. Using the same five (5) documents with different versions, the researcher compared the results of the similarity checking.

A web-based application for detecting plagiarism called Turnitin is widely utilized in educational institutions worldwide. Prepostseo is a web-based plagiarism detection tool frequently used by writers and educational organizations to ensure original written content.

The researcher utilized Turnitin and Prepostseo to contrast some test documents the system's similarity feature had also examined. The effectiveness of the system's algorithms in detecting plagiarism was assessed by comparing the findings from Turnitin and Prepostseo against the developed system. This made it possible to evaluate the system's performance more thoroughly and gave valuable insights into its advantages and disadvantages.

Document Name	ABOUT
DOC_ORIGINAL	The original and raw document.
DOC_FORMATTED	The formatted document.
DOC_100_EDITED	Paraphrased version of the original document.

DOC_75_SAME	Paraphrased the first paragraph of the original document.
DOC_50_SAME	Paraphrased the first and second paragraph of the original document.
DOC_25_SAME	Paraphrased the first, second, and third paragraph of the original document.

Table 4. 7 Test Documents Used in Turnitin, Prepostseo, and the Developed System

Table 4.7 displays the test documents used to test the plagiarism feature of the three applications, and it also shows what the documents were about. DOC_FORMATTED, DOC_100_EDITED, DOC_75_SAME, DOC_50_SAME, and DOC_25_SAME was compared to DOC_ORIGINAL.

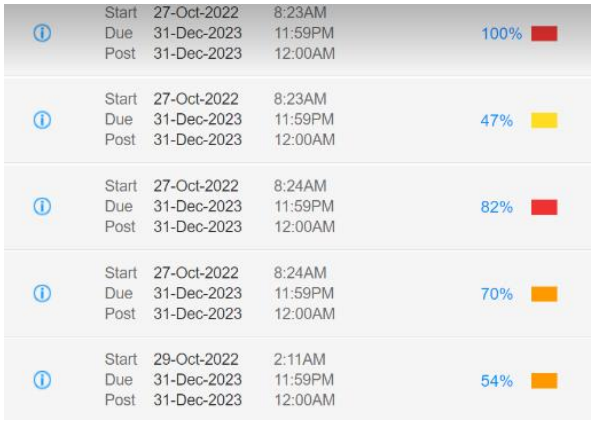


Figure 4. 17 Turnitin Similarity Results

Figure 4.17 shows the Turnitin similarity results of the test documents. The results vary from 47% to 100%. The document that received a 47% similarity result was the DOC_100_EDITED; this shows that even if it was the complete paraphrased version of the original document, it's still considered similar. DOC_50_SAME and DOC_25_SAME received 70% and 54% similarity results, respectively. The results indicate that the content of the two documents was still noticeably identical to the original document, considering that two or three paragraphs were paraphrased. The paper with an 82% similarity score was DOC_75_SAME, indicating that it's similar to the original document, especially since only the first paragraph of this document was a paraphrase. The DOC_FORMATTED document, compared to the original record, got a similarity score of 100% since this document was only formatted with its font size and style changed.

Apart from Turnitin, the researcher also utilized Prepostseo to compare the documents. Prepostseo is a plagiarism checker tool that helps determine the content's originality. Using two different instruments, the researcher compared the comparison results and evaluated the effectiveness of the developed system and its document similarity feature.

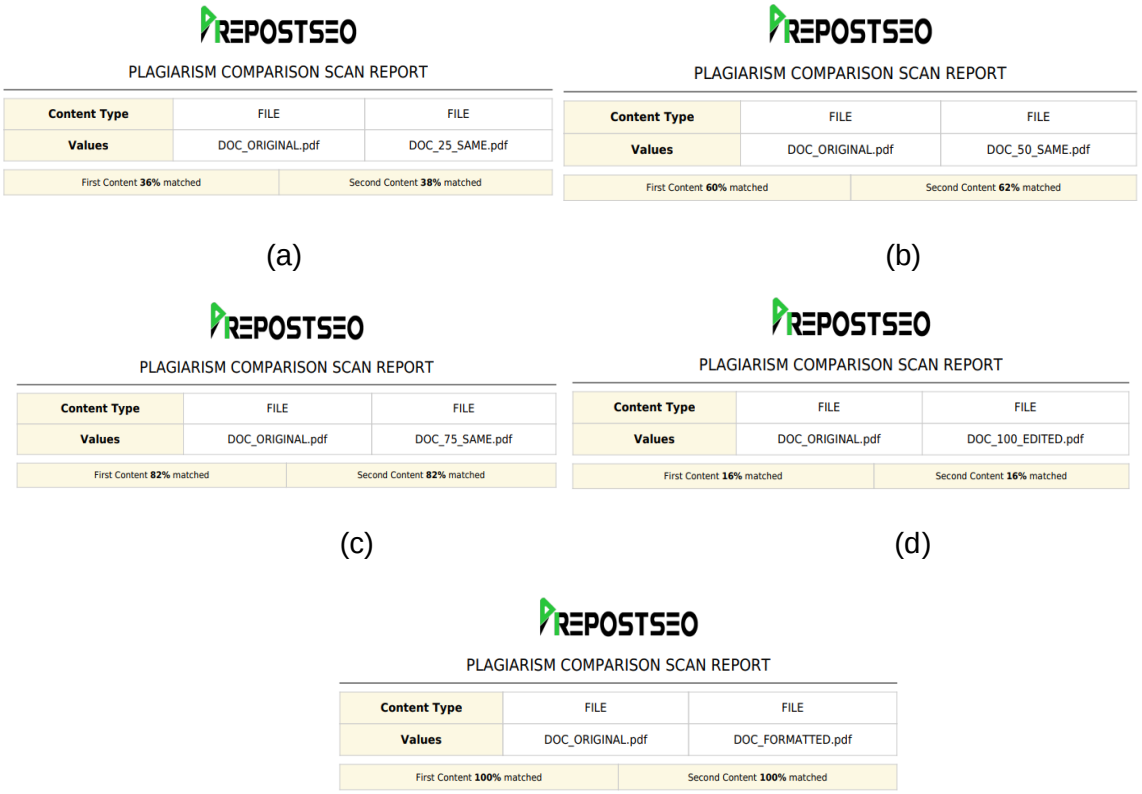


Figure 4. 18 Prepostseo Similarity Results

Figure 4.18 shows the content similarity result using Prepostseo, a plagiarism detection tool. It visualized the percentage similarity between the original document and the different versions of the same document.

Figure 4.18a shows the content similarity of the original document compared to the paper with its first, second, and third paragraphs paraphrased, which has a 38% similarity.

Figure 4.18b shows the content similarity of the original document compared to the record with its first and second paragraphs paraphrased, which has a 62% similarity.

Figure 4.18c shows the content similarity of the original document compared to the paper with its first paragraph paraphrased, which has an 82% similarity.

Figure 4.18d and Figure 4.18e show the original document's results compared to the translated and formatted version of itself; it generated results of 16% and 100%, respectively.

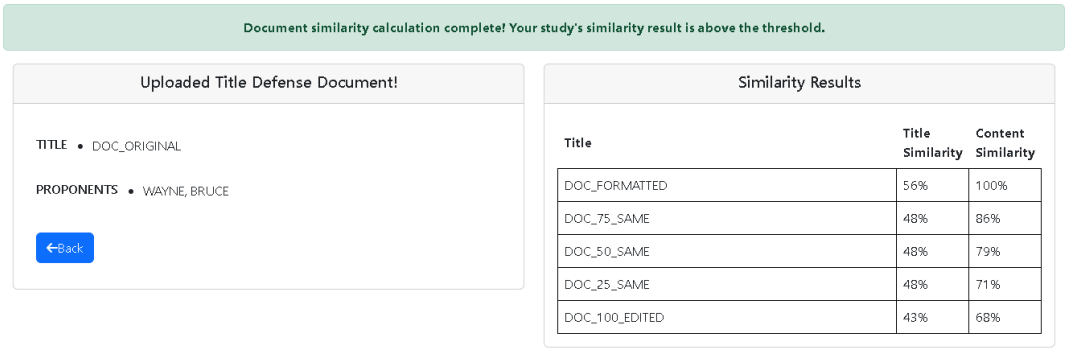


Figure 4. 19 Developed Systems' Similarity Results

Figure 4.19 display the similarity results of the developed web application using the same test documents as Turnitin and Prepostseo. The systems' document similarity results using the original document as a query are the following:

The formatted version of the original document, DOC_FORMATTED, has the highest similarity score of 100%. This was expected because the record remains unchanged except for minor formatting adjustments.

DOC_75_SAME, the paraphrased version of the first paragraph of the original text, has the next-highest similarity score of 86%. Due to the consistency of the paragraph's overall structure and content, this document has a high degree of similarity.

Less similar results are found for the remaining documents, which range from 79% to 68%. Given that these texts contained more than two or three paraphrased content paragraphs.

Using the similarity scores from Turnitin and Prepostseo compared to the developed system, if arranged in descending order, it shows the same sequence with the developed systems' similarity results.

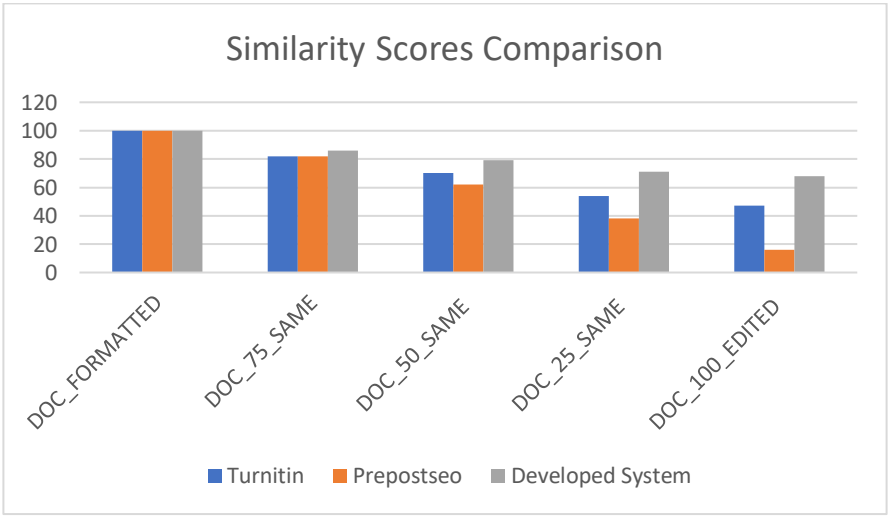


Figure 4. 20 Turnitin, Prepostseo, Developed System Similarity Scores Comparison

Figure 20 shows the similarity scores for each document compared version compared to the original document across the three web applications.

For the DOC_FORMATTED version, all three web applications show a 100% similarity score, indicating that the document's formatting does not impact the similarity percentage.

For the DOC_75_SAME version, both Turnitin and Prepostseo show an 82% similarity score, while the Developed System offers a slightly higher 86% similarity score.

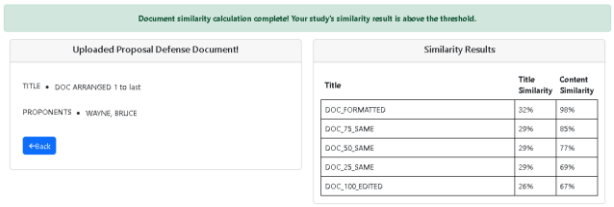
For the DOC_50_SAME version, Turnitin shows a 70% similarity score, Prepostseo shows a lower 62% similarity score, and the Developed System offers a higher 79% similarity score.

For the DOC_25_SAME version, Turnitin shows a 54% similarity score, Prepostseo shows a lower 38% similarity score, and the Developed System offers a significantly higher 71% similarity score.

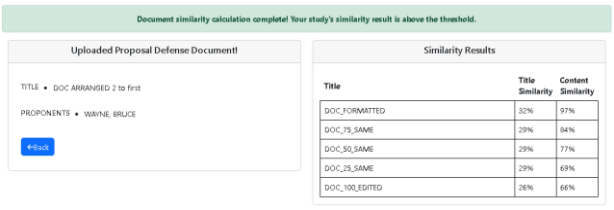
Finally, for the DOC_100_EDITED version, Turnitin shows a 47% similarity score, Prepostseo shows a lower 16% similarity score, and the Developed System offers a higher 68% similarity score.

The result shows the capability of the Developed System to compare documents regardless if the paper was a paraphrase. The Developed System also assessed the similarity of each title of the documents. This could help identify instances where researchers or students used the same title with other existing records. The similarity results of each web application vary depending on the algorithms used. Therefore, even if the three systems take slightly different approaches to detecting similarity, they can be valuable tools for spotting plagiarism and guaranteeing originality in academic and professional work.

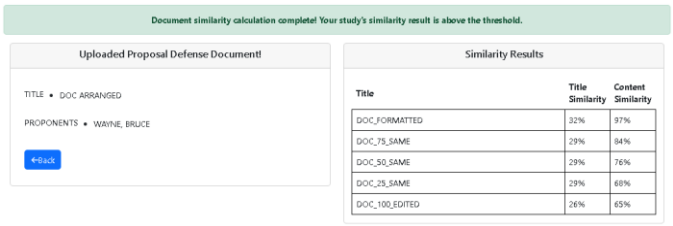
To further test the effectiveness of the developed system, the researcher tested a factor related to the arrangement of the paragraphs. Using the original document above, the researcher arranged each paragraph from first to last, second to first, and randomly set each section in any order.



(a)



(b)



(c)

Figure 4. 21 Similarity Results using the Developed System with Arranged Content

Figure 4.21a presents the similarity results to the original document. At the same time, the first paragraph of the content was rearranged and placed in the last part of the document and was compared to the repository containing five documents as a corpus. The documents were different versions of the original document. The results show that the arranged content was 98% similar to the formatted copy of the original document.

Figure 4.21b shows the results of the second document that was arranged. The results present that the content similarity was 97% which has a 1% difference from Figure 4.21a. The second paragraph of the content of the second document was arranged and placed as the first paragraph.

Figure 4.21c visualizes the results of the randomly arranged document version of the original document. The results show that content similarity was 97%, with mostly the same results against Figure 4.21b.

Figure 4.21 results, compared to Figure 4.19, shows that even if the sequence or the paragraphs of documents were arranged, it still has the exact arrangement of the similarity results as the original document found in Figure 4.19. with slightly varying similarity results, the developed system could detect similarity even if the documents' content was arranged randomly, intentionally, or unintentionally.

CHAPTER V
CONCLUSION AND RECOMMENDATIONS

Conclusion

The digital repository is vital in this growing organization. Similarly, document similarity checking is crucial in promoting integrity and honesty in the organization.

Based on the results of the tests conducted by the researcher, it was deduced that the algorithms: TF-IDF, Cosine Similarity, and K-NN, along with NLP techniques, were influential in creating the document similarity feature of the system. It was also proven from the beta testing results that the system's design, functionality, reliability, usability, and efficiency were aesthetically pleasing, user-friendly, and well-received. The system's main features were also judged to be working and trustworthy tools.

In conclusion, the researcher achieved its overall goal of creating an online repository of thesis documents with similarity checking to identify accidental plagiarism and promote originality in thesis ideas. The successful deployment and evaluation of the system also contributed to achieving the defined goals.

Recommendations

The web application may have been proven effective in the digital world, especially with its main features: digital repository and document similarity checking. However, the researcher still suggests further improvement to polish the application in its process. The following are the recommendations to be considered:

- The researcher suggests adding another algorithm to the document similarity checking to get its semantic content.
- The researcher suggests adding a new feature wherein the user can see which document parts are similar.
- To enhance the system, integrate more NLP techniques.
- Use cloud storage to store the documents to add security and backups.
- It is also recommended that the panels can give comments per page.

References:

[1] Anders Björklund (2017). What is Online vs Digital?. Retrieved from https://zooma.agency/en/learn/digitalisation/online-vs-digital#:~:text=Digital%2C%20meanwhile%2C%20is%20a%20format,a%20disc%20or%20hard%20drive.

[2] Pinfield S, Cox AM, Smith J (2014). Research Data Management and Libraries: Relationships, Activities, Drivers and Influences. PLoS ONE 9(12): e114734. Retrieved from https://doi.org/10.1371/journal.pone.0114734

[3] Singh, P. (2016). Open access repositories in India: Characteristics and future potential. IFLA Journal, 42(1), 16–24. Retrieved from https://doi.org/10.1177/0340035215610131.

[4] LISBDNETWORK (2018). Advantages and Disadvantages of the Digital Library. Retrieved from https://www.lisedunetwork.com/advantages-and-disadvantages-of-the-digital-library/

[5] Dmitriy Selivanor (2018). Document Similarity. Retrieved from https://text2vec.org/similarity.html#documents_similarity

[6] Baeldung (2023). Semantic Similarity of Two Phrases. Retrieved from https://www.baeldung.com/cs/semantic-similarity-of-two-phrases#:~:text=2.-,Text%20Similarity,be%20lexical%20or%20in%20meaning.

[7] Stecanella, B. (2019). Understanding TF-ID: A simple Introduction. Retrieved from https://monkeylearn.com/blog/what-is-tf-idf/

[8] Scott, W. (2019). TF- IDF from scratch in python on a real-world dataset. Retrieved from https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

[9] Ganesan, K. (2019). What is Term Frequency? Retrieved from https://www.opinosis-analytics.com/knowledge-base/term-frequency-explained/

[10] Inverse Document Frequency. (n.d). Retrieved from https://blog.marketmuse.com/glossary/inverse-document-frequency-idf-definition/

[11] Heres, D. (2017). Source code plagiarism detection using machine learning (Master's thesis). Retrieved from https://studenttheses.uu.nl/handle/20.500.12932/27904

[12] Qaiser, S. & Ali R. (2018) Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. Retrieved from https://www.researchgate.net/publication/326425709_Text_Mining_Use_of_TF-IDF_to_Examine_the_Relevance_of_Words_to_Documents?enrichId=rgreq-70c92401e64c214d416f8493f2bd8bb6-XXX&enrichSource=Y292ZXJQYWdlOzMyNjQyNTcwOTtBUzo2NDkwNzEyMjQxNjQzNTNAMTUzMTc2MjA0NzQ1Nw%3D%3D&el=1_x_3&_esc=publicationCoverPdf

[13] Kim, S.W. & Gil, J.M. (2019) Research paper classification systems based on TF-IDF and LDA schemes. Retrieved from https://doi.org/10.1186/s13673-019-0192-7

[13] Faith Karabiber (n.d). Retrieved from https://www.learndatasci.com/glossary/cosine-similarity/

[14] Jones M. (2009) EVALUATION OF THE EFFECTIVENESS OF COSINE SIMILARITY IN PREDICTING RELEVANCE BETWEEN PAIRED CITING AND CITED SENTENCES. Retrieved from https://cdr.lib.unc.edu/concern/masters_papers/73666817r?locale=en

[15] Javatpoint. (n.d.). K-nearest neighbor algorithm for machine learning. Retrieved from https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[16] Ali, N., Neagu, D. & Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. SN Appl. Sci. 1, 1559 (2019). https://doi.org/10.1007/s42452-019-1356-9

[17] References: Levy, F., Pyles, R., Szarejko, C.M., & Wyatt, L.G. (2012). Developing an Electronic Repository for Undergraduate Theses. Retrieved from https://core.ac.uk/download/pdf/17269166.pdf

[18] Kania, R., Solihati, T., & Arzaqi, F. (2022). THESIS SIMILARITY DETECTION APPLICATION AT BANTEN JAYA UNIVERSITY. Jurnal Sistem Informasi Dan Informatika (Simika), 5(1), 78-89. Retrieve from https://doi.org/10.47080/simika.v5i1.1682

[19] Oppi Anda Resta, Aditya, A., & Febry Eka Purwiantono. (2021). Plagiarism Detection in Students' Theses Using The Cosine Similarity Method. Sinkron : Jurnal Dan Penelitian Teknik Informatika, 5(2), 305-313.Retrieved from https://doi.org/10.33395/sinkron.v5i2.10909

[20] DEL ROSARIO, M. J. ., & SARENO, J. . (2020). Theses and Capstone Projects Plagiarism Checker using Kolmogorov Complexity Algorithm. Walailak Journal of Science and Technology (WJST). Retrieved from https://doi.org/10.48048/wjst.2020.6498

[21] Dinauanao, A. M.. (2013). Development of the Online Repository of Theses and Dissertations of the University of Cebu – Graduate School Library (ucGSLIB). IAMURE International Journal of Multidisciplinary Research, 7(1). Retrieved from http://ejournals.ph/form/cite.php?id=2599

[22] Mesa, A.R. (2017). Design and Development of an Online Repository System for Thesis and Special Problem Manuscripts. Retrieved from https://www.researchgate.net/publication/353706725_Design_and_Development_of_an_Online_Repository_System_for_Thesis_and_Special_Problem_Manuscripts

Appendix A **Beta testing form**

Republic of the Philippines
Western Mindanao State University
College of Computing Studies
Department of Computer Science

Dear Ma'am /Sir,

I am writing to request your participation in a survey questionnaire that forms part of my BS in Computer Science thesis project entitled "**Development of Thesis Repository System with Document Similarity Feature for the College of Computing Studies**" at Western Mindanao State University.

The purpose of this questionnaire is to evaluate whether the objectives of my research have been achieved, and your valuable feedback will greatly contribute to the success of my study. Therefore, I would like to request that you kindly take the time to complete the questionnaire as accurately and completely as possible.

All responses will be kept confidential and participation in the survey is entirely voluntary. Your cooperation in this study would be deeply appreciated. Thank you for considering my request.

RESPONDENT'S CONSENT TO PARTICIPATE IN THE SURVEY

I understand that the purpose of this study is to develop an online system for storing theses with a document similarity feature that aims to improve the effectiveness and efficiency of information retrieval while minimizing the occurrence of unintentional plagiarism. By participating in this survey, I acknowledge that my responses will be utilized to evaluate the design, functionality, reliability, usability, and efficiency of the proposed system.

I understand that my participation in this study is voluntary and that I may choose to withdraw from the survey at any time without any negative consequences. I further understand that all data collected from me during this survey will be treated with the utmost confidentiality. The results and interpretations of this study will be utilized solely for the purpose of this research.

The specific objectives of this study include utilizing the term frequency-inverse document frequency (TF-IDF) method to vectorize the documents in the repository, using the cosine similarity technique to compare the vectorized documents, and utilizing the k-nearest neighbor (k-NN) technique to present the five documents that are the most comparable to a query. Additionally, natural language processing techniques will be utilized to enhance the system's performance. A user-friendly interface will be developed to manage and access the stored theses in the repository.

Thank you for your participation, and your valuable insights will be greatly appreciated in advancing the development of the proposed system.

Signature of the Respondents: _____

Date: _____

PART I: RESPONDENT PROFILE

Direction: Please provide the following information about yourself:

Name (Optional): _____
College: _____
Course: _____
Year: _____

PART II. SOFTWARE EVALUATION

Direction: After utilizing the "**Thesis Repository System with Document Similarity Feature**", please check the box next to the statement that best represents your response to the following questions

Legend:
5 – Strongly Agree **2** – Slightly Disagree
4 – Agree **1** – Strongly Disagree
3 – Slightly Agree

Design					
	5	**4**	**3**	**2**	**1**
The system's overall layout is user-friendly and visually appealing.					
The font is designed in an easy-to-read size and style.					
The system's layout makes finding what I'm seeking for simple.					

Functionality					
	5	**4**	**3**	**2**	**1**
The system is able to perform all the functions it claims to do.					
Main Function/s					
• Digital Repository					

Uploading Document and Perform Similarity Check					
➢ Title Defense Document					
➢ Proposal Defense Document					
➢ Final Defense Document					
Minor Function/s:					
● Retrieval of Document in Digital Repository					
● Uploaded Documents History					
The document similarity feature of the system has acceptable accuracy and is a reliable tool for comparing documents.					
The system operates smoothly without any major bugs or errors.					
The system's performance speed is fast and responsive.					

Reliability					
	5	**4**	**3**	**2**	**1**
The system is available and accessible when I need it.					
The system does not experience frequent downtime or crashes.					

Usability					
	5	**4**	**3**	**2**	**1**
The system's user interface is clear and simple to use.					
Task completion and system navigation are simple.					
The system's labels and instructions are brief and easy to understand.					
Users receive beneficial feedback from the system.					

Efficiency					
	5	**4**	**3**	**2**	**1**
The system allows me to complete tasks quickly and efficiently.					

The document similarity feature of the system saves me time in detecting unintentional plagiarism.					
The system's search function quickly finds relevant information.					
The system's performance speed is fast enough to meet my needs.					

Are there any additional features or improvements you would like to see in the system?

_____

_____

_____

Thank you very much!

Mark Anthony Tubat
Researcher

Appendix B **Picture of the system**

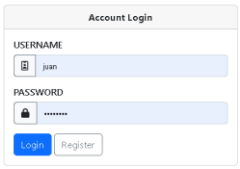


Figure Appendix B.1. Login page

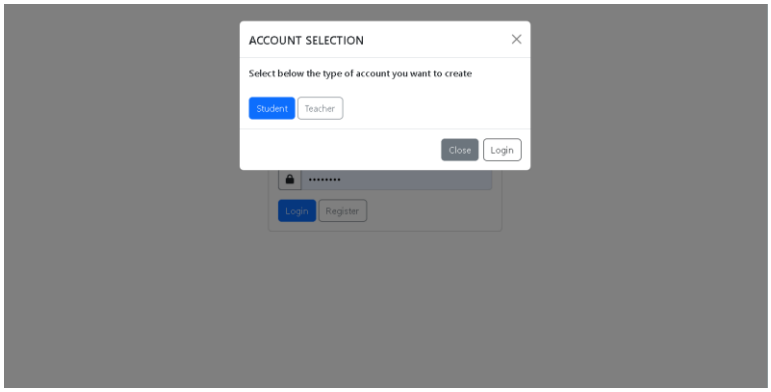


Figure Appendix B.2. Account selection

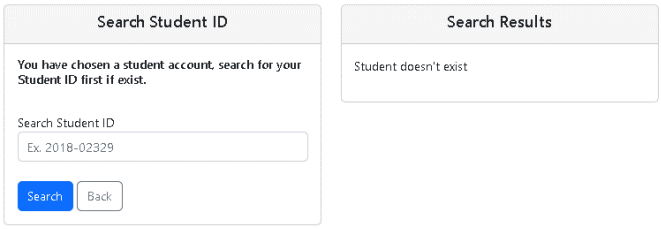


Figure Appendix B.3. Search student ID

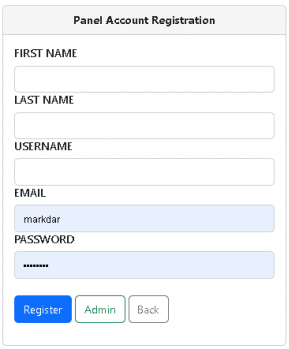


Figure Appendix B.4. Panel Account Registration

Figure Appendix B.5. Admin Account Registration

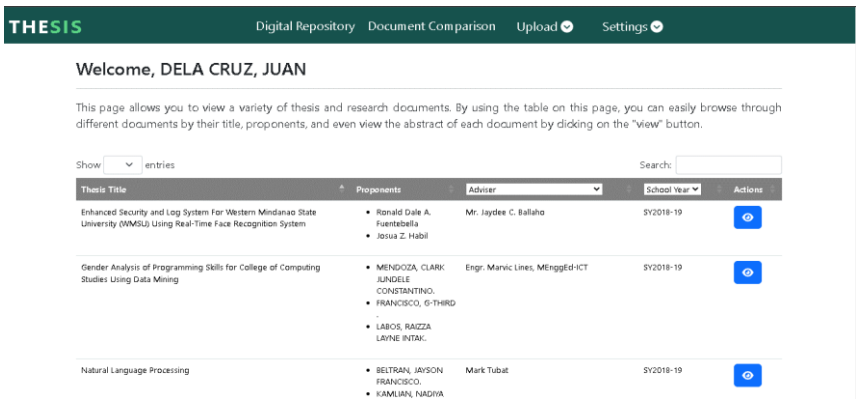


Figure Appendix B.6. Student Dashboard (Digital Repository)

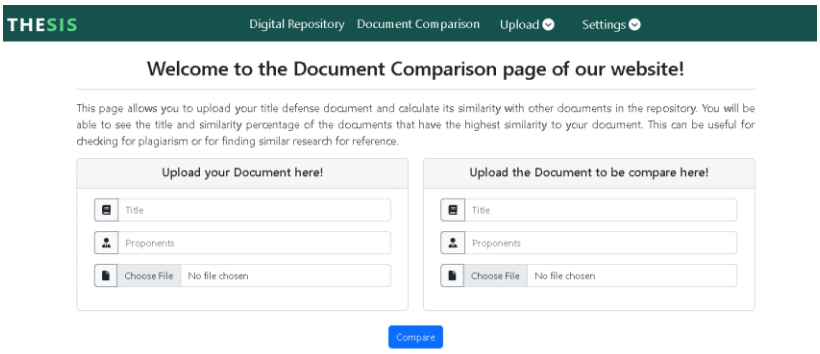


Figure Appendix B.7. Document Comparison

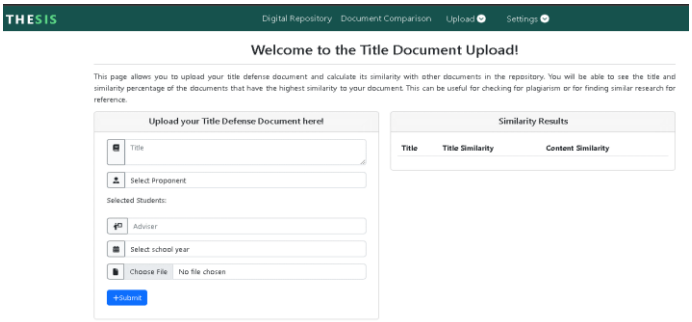


Figure Appendix B.8. Upload Title Defense Document

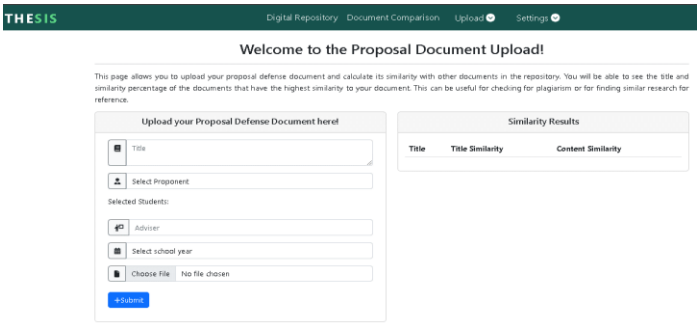


Figure Appendix B.9. Upload Proposal Defense Document

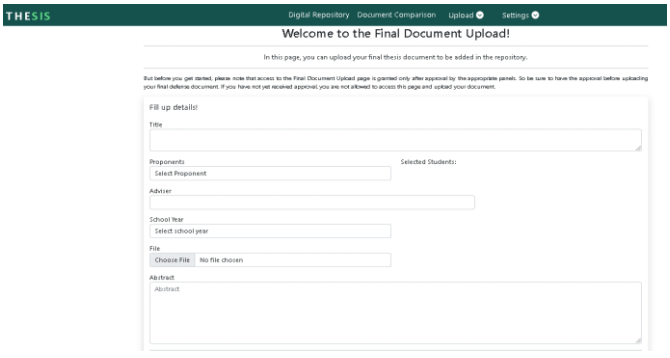


Figure Appendix B.10. Upload Final Defense Document

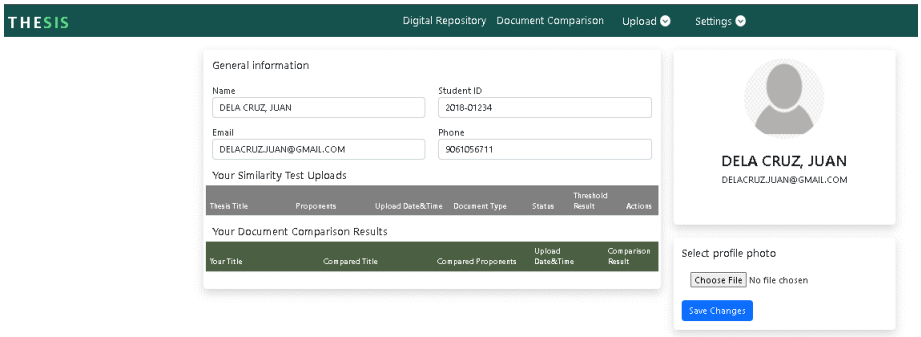


Figure Appendix B.11. Student Account Settings

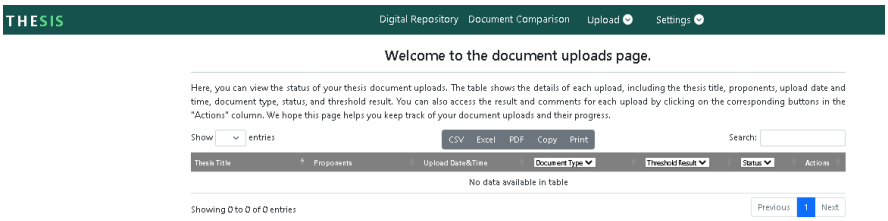


Figure Appendix B.11. Uploaded Document Results

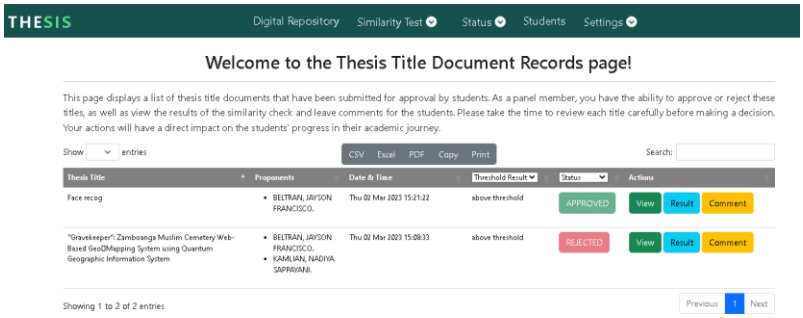


Figure Appendix B.12. Uploaded Thesis Title Defense Document

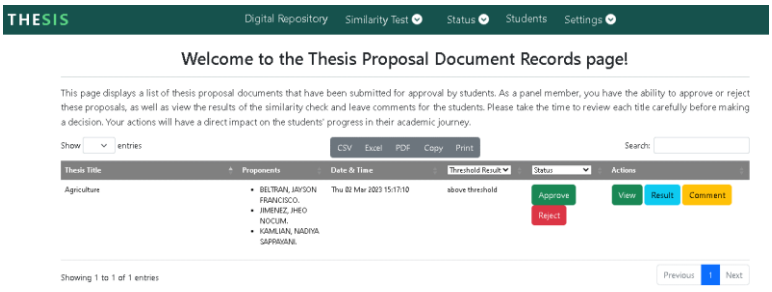


Figure Appendix B.12. Uploaded Thesis ProposalDefense Document

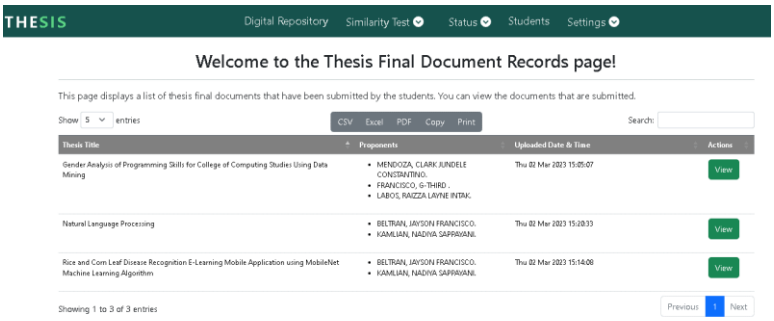


Figure Appendix B.13. Uploaded Thesis Final Defense Document

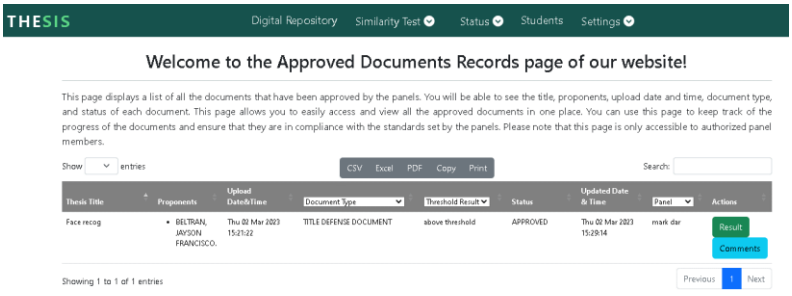


Figure Appendix B.14. Approved Documents

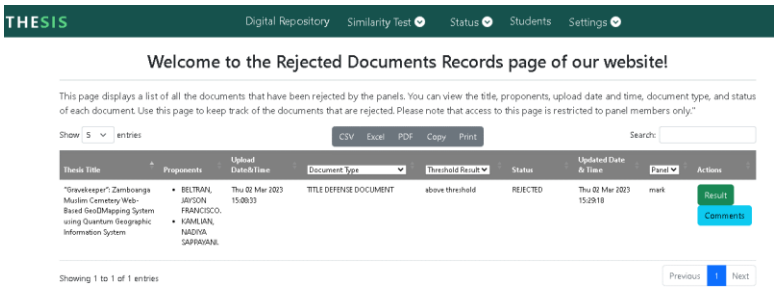


Figure Appendix B.15. Rejected Documents

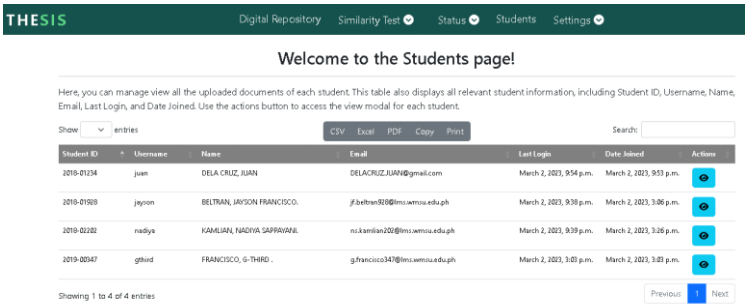


Figure Appendix B.16. Registered Students with their uploads

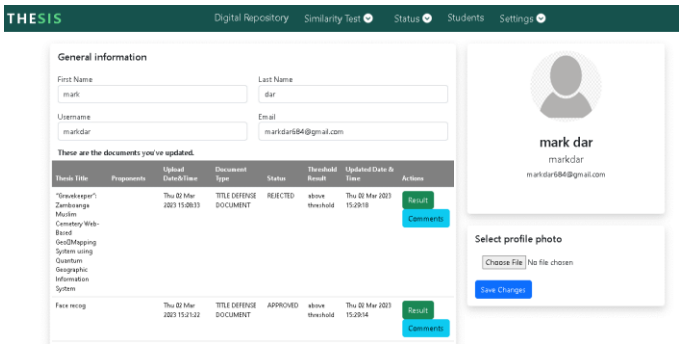


Figure Appendix B.17. Panel Account Settings

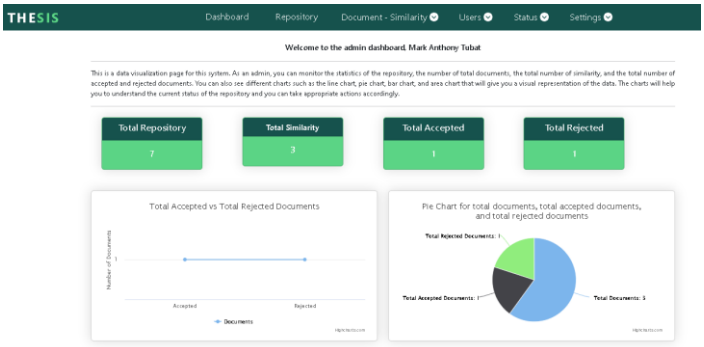


Figure Appendix B.18. Admin Dashboard

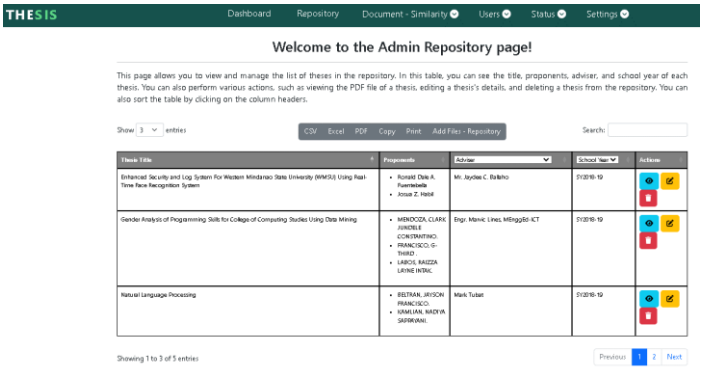


Figure Appendix B.19. Admin Repository

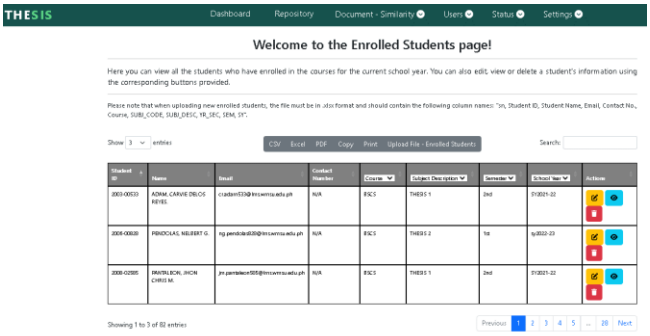


Figure Appendix B.20. Enrolled Students

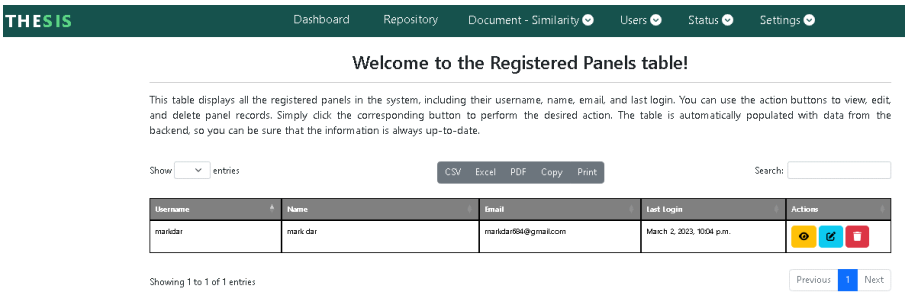


Figure Appendix B.21. Registered Panels

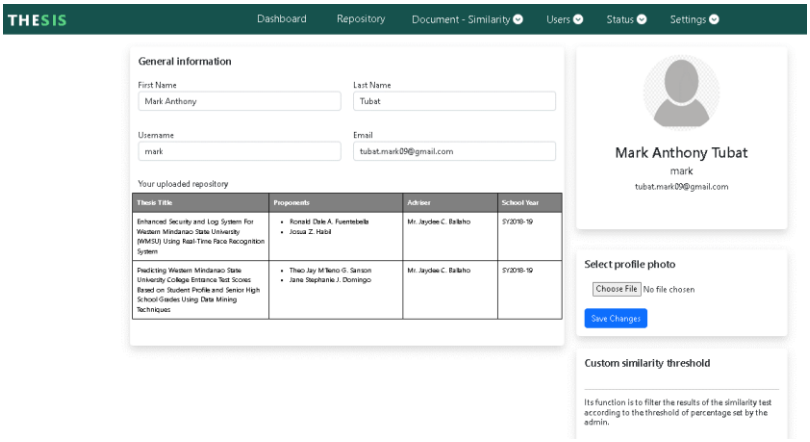


Figure Appendix B.22. Admin Account Settings

	A	B	C	D	E	F	G	H	I	J	K
1	sn	Student ID	Student Name	Email	Contact No.	Course	SUBJ_CODE	SUBJ_DESC	YR_SEC	SEM	SY

Figure Appendix B.23. Excel format of Officially Enrolled Students

Appendix C **Beta Test Results Summary**

Design Category

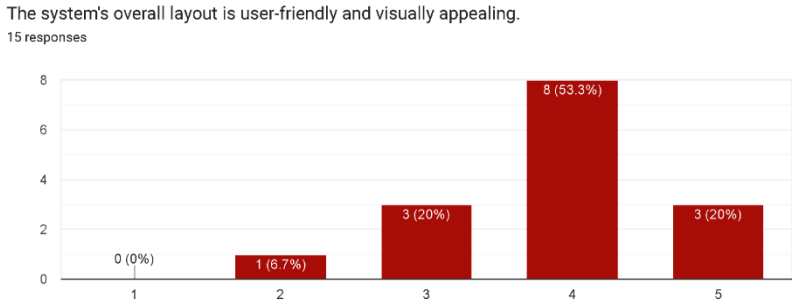


Figure Appendix C.1. Design Question 1

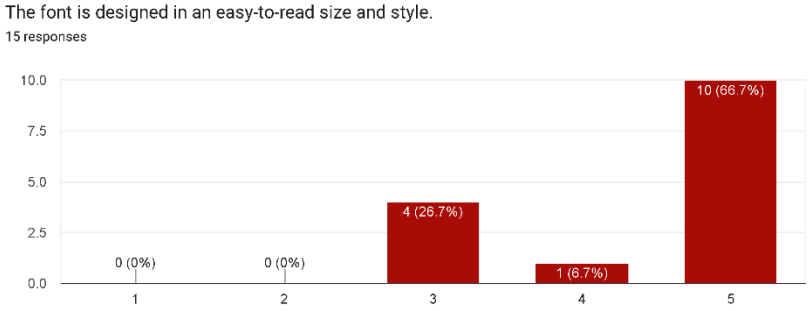


Figure Appendix C.2. Design Question 2

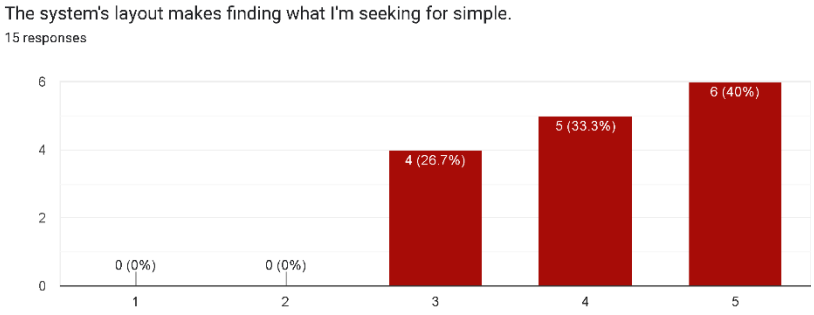


Figure Appendix C.3. Design Question 3

Functionality Category

Main Functions

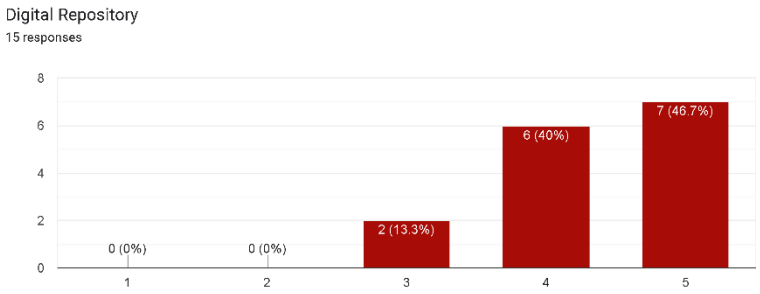


Figure Appendix C.4. Functionality Question 1

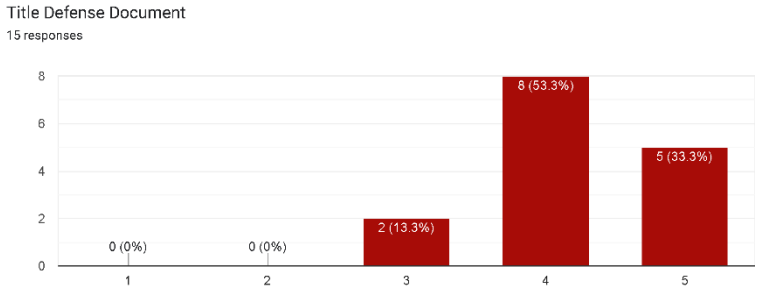


Figure Appendix C.5. Functionality Question 2

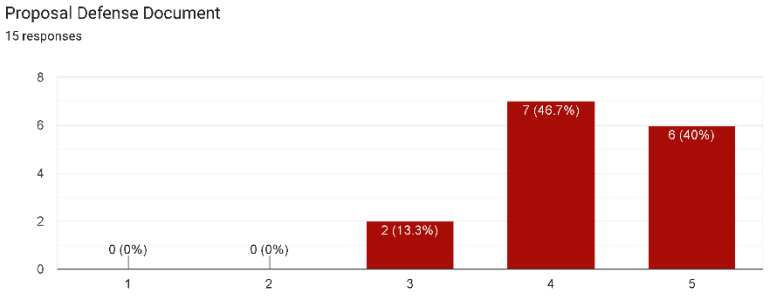


Figure Appendix C.6. Functionality Question 3

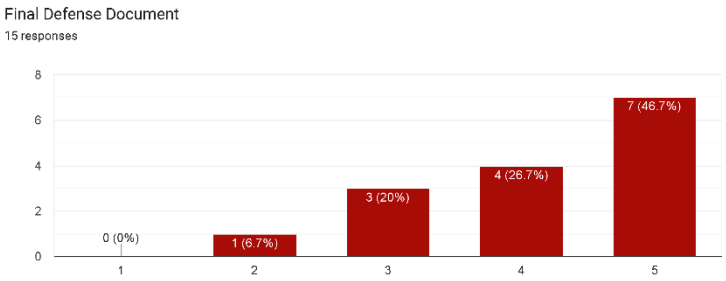


Figure Appendix C.7. Functionality Question 4

Minor Functions

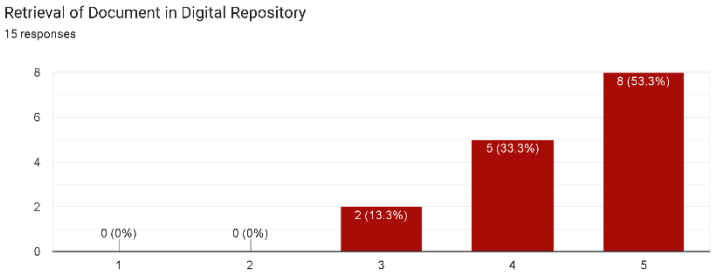


Figure Appendix C.8. Functionality Question 5

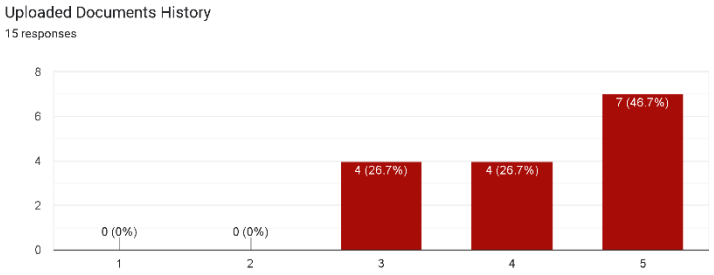


Figure Appendix C.9. Functionality Question 6

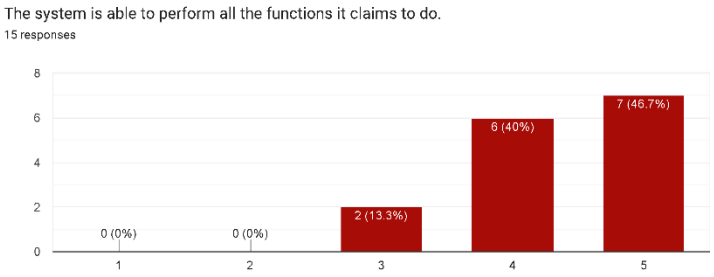


Figure Appendix C.10. Functionality Question 7

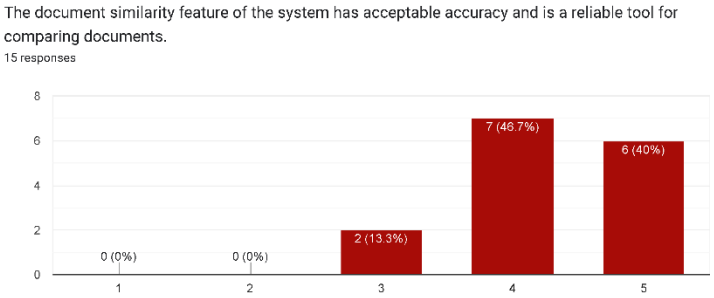


Figure Appendix C.11. Functionality Question 8

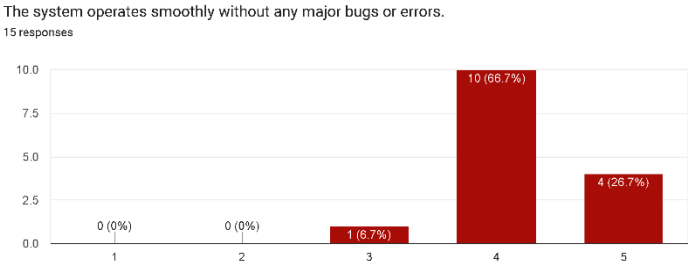


Figure Appendix C.12. Functionality Question 9

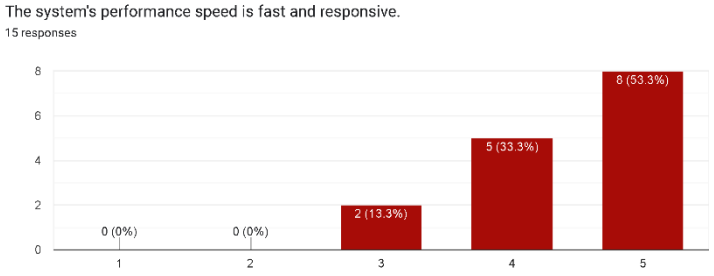


Figure Appendix C.13. Functionality Question 10

Reliability Category

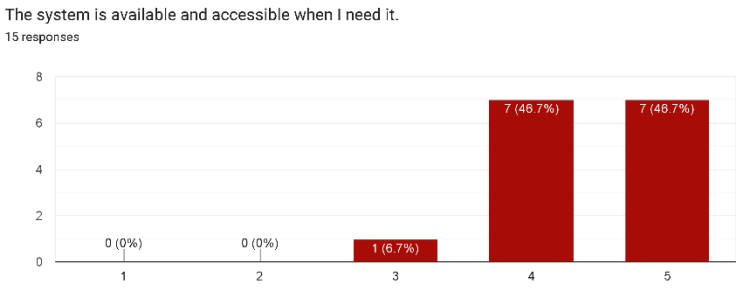


Figure Appendix C.14. Reliability Question 1

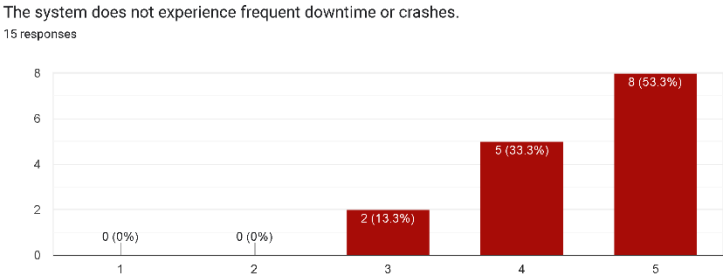


Figure Appendix C.15. Reliability Question 2

Usability Category

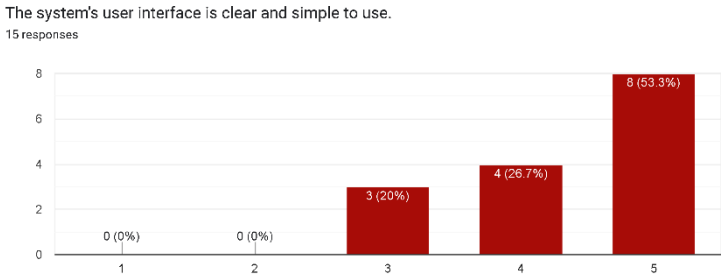


Figure Appendix C.16. Usability Question 1

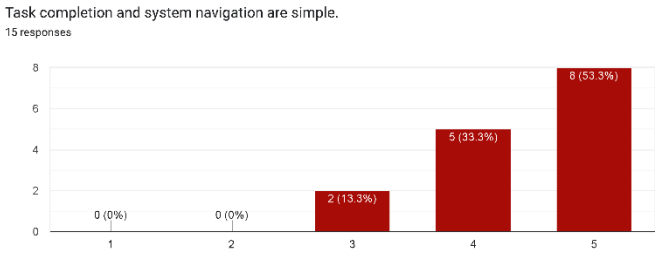


Figure Appendix C.17. Usability Question 2

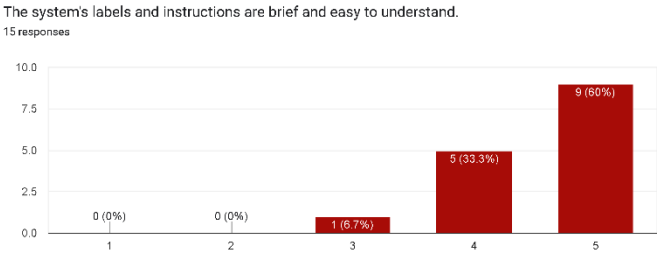


Figure Appendix C.18. Usability Question 3

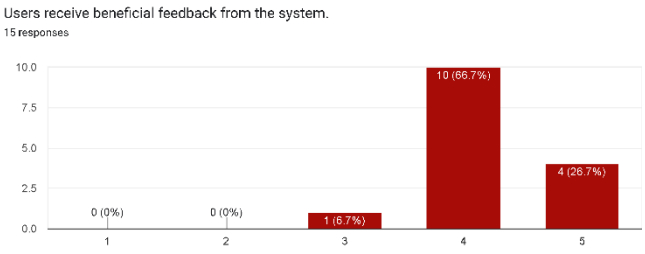


Figure Appendix C.19. Usability Question 4

Efficiency Category

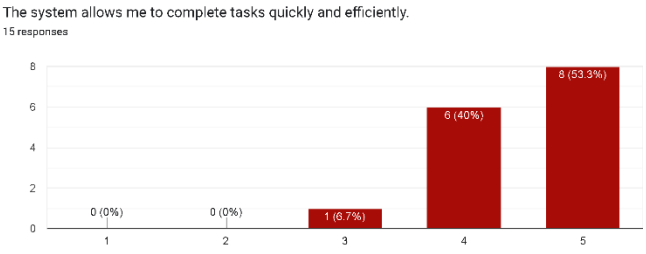


Figure Appendix C.20. Efficiency Question 1

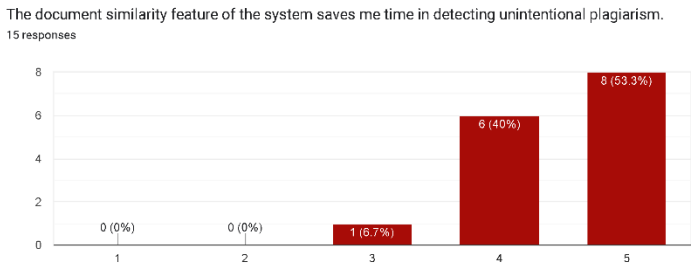


Figure Appendix C.21. Efficiency Question 2

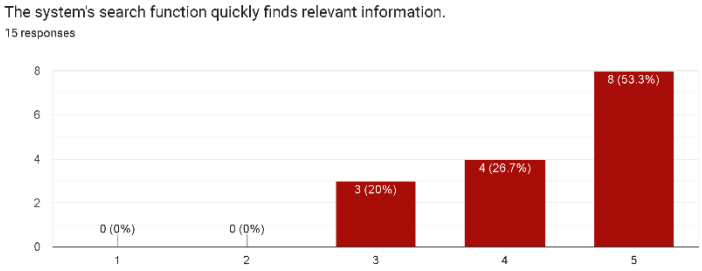


Figure Appendix C.22. Efficiency Question 3

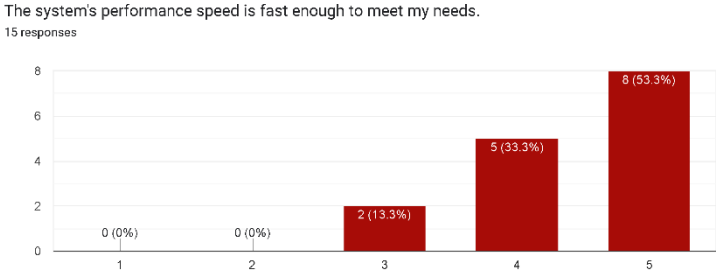


Figure Appendix C.23. Efficiency Question 4

Appendix D **Flowcharts**

Figure Appendix D.1. ***Student site***

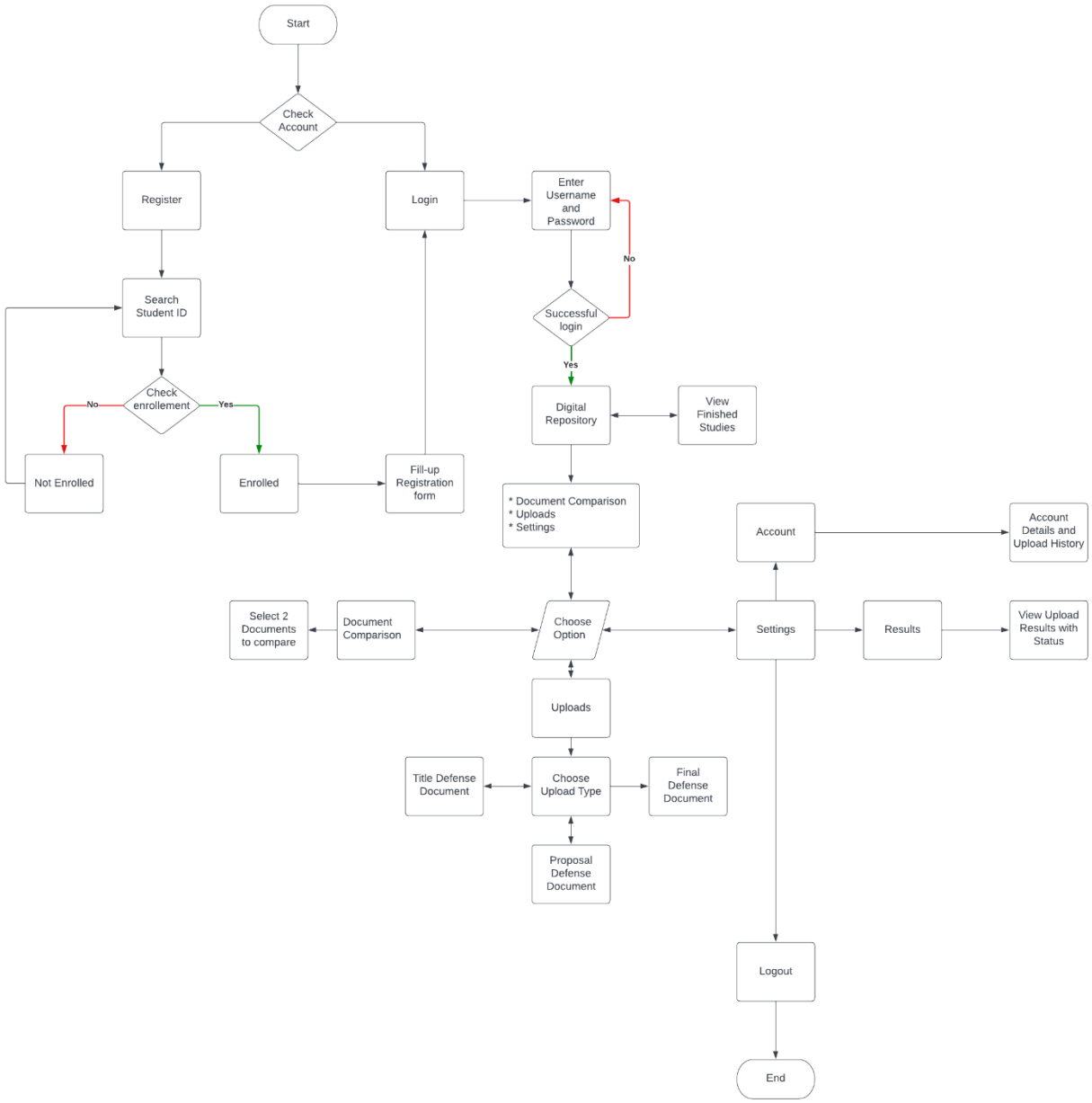


Figure Appendix D.2. ***Panel Site***

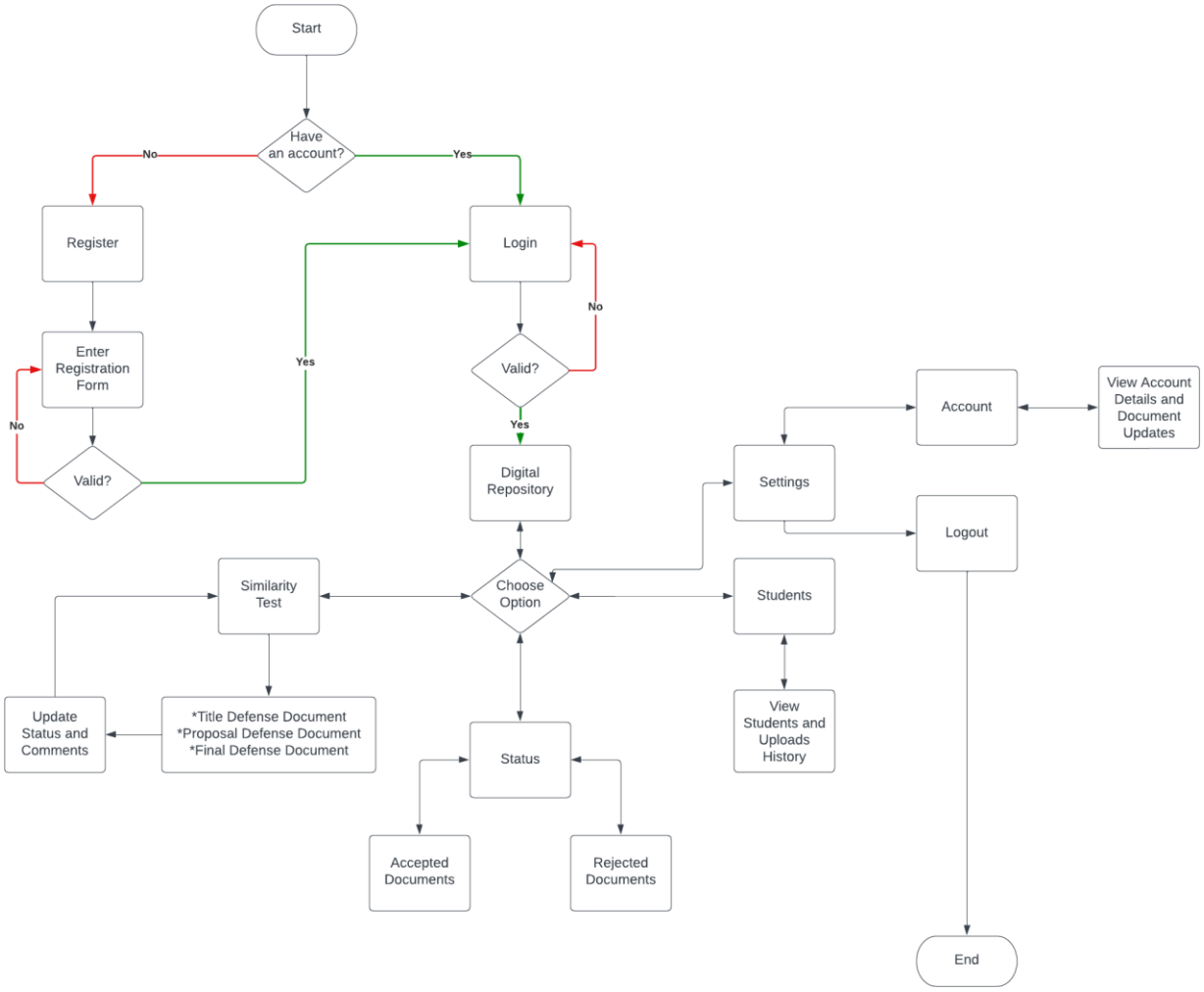
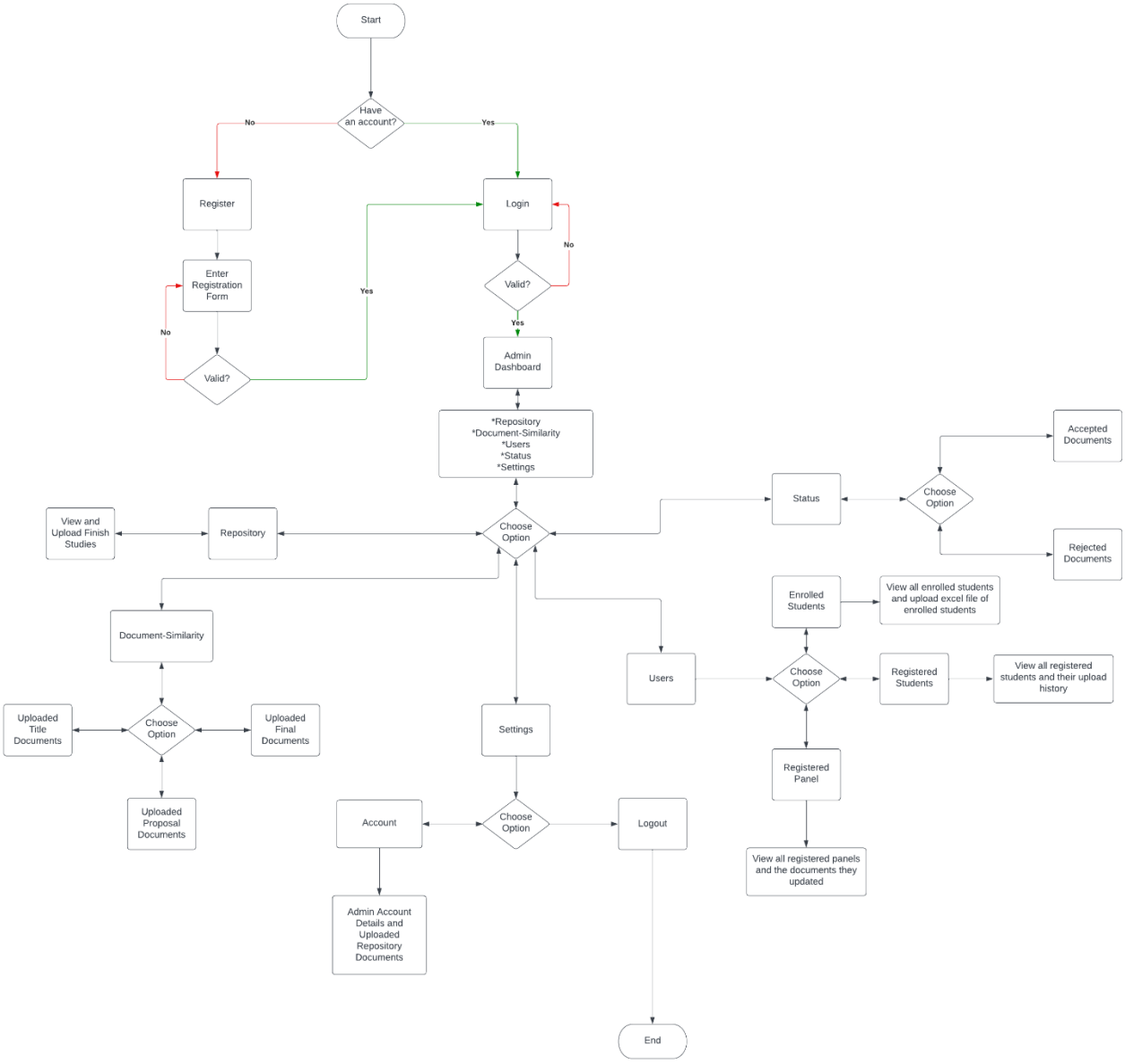
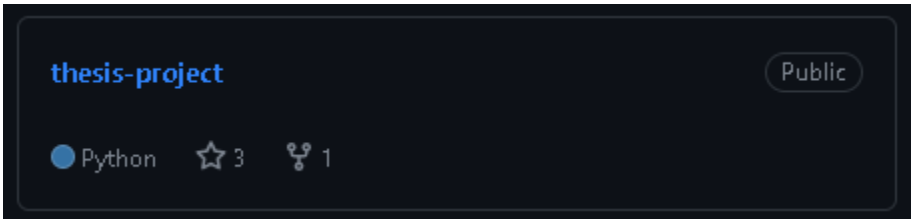


Figure Appendix D.3. ***Admin Site***



Appendix E **Source Code**



Github - [TubatMark/thesis-project (github.com)]

The link above contains the source code for the web-based application of this project.

Function to Extract/Convert PDF to .txt file

The pdf was first converted into text files and saved in the database or in a folder. The function “extract_pdf_text” was for the admin side uploading data into the repository. The function “student_pdf_text” was for the student side uploading thesis documents for similarity checking.

```
def extract_pdf_text(pdf_file, repository_file):
    # open the PDF file
    pdf = PyPDF2.PdfReader(pdf_file)
    # extract the text from each page and save it in a list
    text_list = [pdf.pages[page].extract_text() for page in range(len(pdf.pages))]
    # join all the texts from the list and save it as a single string
    text = "\n".join(text_list)
    # construct the file path using MEDIA_ROOT and MEDIA_URL
    file_path = os.path.join(settings.MEDIA_ROOT, "ExtractedFiles")
    if not os.path.exists(file_path):
        os.makedirs(file_path)
    text_file_name = pdf_file.name.replace('.pdf', '.txt')
    text_file = os.path.join(file_path, text_file_name)
    with open(text_file, 'w', encoding='utf-8') as f:
        f.write(text)
    # Save the file path to the database
    repository_file.text_file = text_file
    repository_file.save()

def student_pdf_text(pdf_file, vectorizer):
    # open the PDF file
    pdf = PyPDF2.PdfReader(pdf_file)
    # extract the text from each page and save it in a list
    text_list = [pdf.pages[page].extract_text()
                 for page in range(len(pdf.pages))]
    # join all the texts from the list and save it as a single string
    text = "\n".join(text_list)
    # preprocess the text using NLTK
    query_text = preprocess(text)
    query_matrix = vectorizer.transform([query_text])
    return query_matrix
```

Function to Preprocess Text Documents

To preprocess the documents, it was first open a .txt file containing all the stopwords and saved on a variable; after that, the data was lowercase and started removing punctuations and digits, stopwords and characters less than two and date.

```
def preprocess(data):
    # open and read stopwords.txt
    with codecs.open('stopwords/stopwords.txt', 'r', encoding='utf-8',
errors='ignore') as f:
        stopwords = f.read().splitlines()

    # Convert data to lowercase
    data = data.lower()

    # Remove punctuation and digits
    data = re.sub(r'[^\w\s]', ' ', data)
    data = re.sub(r'\d+', '', data)

    # Remove stop words
    words = word_tokenize(data)
    filtered_words = [word for word in words if word not in stopwords and
len(word) > 2 and not is_date(word)]
    return " ".join(filtered_words)

def is_date(string):
    try:
        dparser.parse(string, fuzzy=True)
        return True
    except ValueError:
        return False
```

Function to Generate Document Similarity Result

The function “vectorize” takes several parameters, including the query matrix, a vectorizer, a k value, a thesis title, and a list of selected proponents. The function begins with retrieving the threshold value set by the admin, and if one exists, it puts it as the threshold to identify if the query document is above/below the threshold. Then retrieves the latest IDs for each title in the RepositoryFiles object; these filters out files with the same proponents as the selected ones. After filtering out and getting the matrices of query and corpus documents, it calculates the cosine similarity between them. It also calculates the similarity between each thesis title of the query document and the titles found in the repository using the fuzzywuzzy library. The function then creates a list of dictionaries containing information about each file, including its title, title similarity score, adviser, school year, document matrix, content similarity score, and list of proponents. It sorts this list by content similarity score in descending order and selects the first k documents. Finally, it adds a flag to indicate if the content similarity score is

below/above the threshold and returns the list of k nearest neighbours. This function is used in a web application to handle document similarity issues for a college.

```
def vectorize(query_matrix, vectorizer, k, student_title, selected_proponents):
    threshold = 1
    last_threshold = SimilarityThreshold.objects.all().last() #admin set threshold
    if last_threshold:
        threshold = last_threshold.threshold
    matrices = []

    # Get the IDs of the selected proponents
    selected_proponents_ids = [proponent.id for proponent in selected_proponents]

    # Get the latest ID for each title
    latest_ids = RepositoryFiles.objects \
        .exclude(proponents__id__in=selected_proponents_ids) \
        .values('title') \
        .annotate(latest_id=Max('id')) \
        .values_list('latest_id', flat=True)

    # Get the RepositoryFiles that don't have the same proponents as the selected
ones and have the latest ID for their title
    filtered_files = RepositoryFiles.objects \
        .exclude(proponents__id__in=selected_proponents_ids) \
        .filter(id__in=Subquery(latest_ids)) \
        .values('text_file', 'title', 'adviser', 'school_year')



    for file_info in filtered_files:
        file_path = file_info['text_file']
        with open(file_path, 'r', encoding='utf-8', errors='ignore') as f:
            text = f.read()
            text = preprocess(text)
            doc_matrix = vectorizer.transform([text])
            similarity = cosine_similarity(query_matrix, doc_matrix)[0][0]
            similarity = round(similarity, 2)
            content_similarity = similarity*100

            #preprocess title
            preprocess_student_title = preprocess(student_title)
            preprocess_corpus_title = preprocess(file_info['title'])

            # calculate the similarity between the titles
            title_similarity = fuzz.token_set_ratio(preprocess_corpus_title,
preprocess_student_title)
```

```
# get the RepositoryFiles object(s) for the current file
            repo_files = RepositoryFiles.objects.filter(text_file=file_path)
# filter out any objects that have the same proponents as the selected ones
            repo_files =
repo_files.exclude(proponents__id__in=selected_proponents_ids)
# skip the file if there are no RepositoryFiles objects left after
filtering
             if not repo_files.exists():
                 continue
# get the proponents for the remaining RepositoryFiles object(s)
             proponents = [proponent.name for proponent in
repo_files[0].proponents.all()]

             matrices.append({
                 'title': file_info['title'],
                 'title_similarity': title_similarity,
                 'adviser': file_info['adviser'],
                 'school_year': file_info['school_year'],
                 'matrix': doc_matrix,
                 'content_similarity': content_similarity,
                 'proponents': proponents
             })
    # Sort the list of documents by similarity in descending order
    matrices.sort(key=lambda x: x['content_similarity'], reverse=True)
    # Select the first k documents
    nearest_neighbors = matrices[:k]

    # Add a flag to indicate if the similarity is below the threshold
    for neighbor in nearest_neighbors:
        if neighbor['content_similarity'] < threshold:
            neighbor['below_threshold'] = True
        else:
            neighbor['below_threshold'] = False
    return nearest_neighbors
```

Appendix F **Gantt Chart**

Phases	September	October	November	December	January	February	March
Phase #1: Requirements Gathering							
Phase #2: Design and Planning							
Phase #3: Implementation							
Phase #4: Testing							
Phase #5: Deployment							
Phase #6: Maintenance							

Appendix G **Curriculum Vitae**

MARK ANTHONY D. TUBAT
Purok Fortune, Poblacion(Titay), Titay, Zamboanga Sibugay
Email: http.mark09@gmail.com
Contact Number: +639750353422

I am a hardworking and willing to learn. Adaptable personality that can go with the team. Passionate about programming, new technology and a self-motivated person.

PERSONAL DETAILS

Gender:Male
Age: 23 years old
Date of Birth: March 09, 2000
Nationality: Filipino
Marital Status: Single

EDUCATIONAL BACKGROUND

Bachelor of Science in Computer Science
Western Mindanao State University
2018 – 2023
Administration, Business, and Management
Titay National High School - Senior High School
2015 – 2017

CHARACTER REFERENCES

Salimar B. Tahil
Faculty, CCS
tahil.salimar@wmsu.edu.ph
+63917-177-1654

Jaydee Ballaho
Faculty, CCS
Jaydee.ballaho@wmsu.edu.ph

TRAININGS AND SEMINAR

Participant,
Master Python and Data Libraries Certification Bootcamp
https://www.udemy.com
2021

Participant,
Front End Web Development Ultimate Course
https://www.udemy.com

SKILLS

- Proficient in Web Development using Django Framework, Python, HTML, CSS, Javascript, and JQuery.
- Proficient with MS Word, Excel, and PPT
- Proficient with Video Editing using Filmora
- PC Troubleshooting
- Good at communicating, analyzation, and problem-solving necessary for the team.