UNIVERSITY STUDENT PROJECT TRACKER (USPT)

BY

OGWUCHE SIMON

PG/MSC/13/65346

Being an MSc project report submitted in partial fulfillment of the requirements for the award of a Master of Science's degree in Computer Science of the University of Nigeria, Nsukka.

SUPERVISOR: DR M. C. OKORONKWO

Department of Computer Science,
University of Nigeria, Nsukka.

September, 2015.

CERTIFICATION PAGE

I, Ogwuche Simon, hereby declare that the work presented herein was done by me, and not by a third party. Should I be convicted of having cheated in this work, I shall accept the verdict of the university.

OGWUCHE SIMON, 2015/PG/65346

APPROVAL PAGE

This project report is approved for submission.

DR M. C. OKORONKWO

DEDICATION

This work is dedicated to God Almighty, the omniscience, omnipotent, all-knowing and the one through whom all good things come from.

ACKNOWLEDGEMENTS

The completion of this work is the product of the persistent support of several persons.

First and foremost, I wish to thank my supervisor Dr M. C. Okoronkwo, for his continuous support. Dr M. C. Okoronkwo has always been there to listen and give me advice. He taught me how to express my ideas, showed me different ways to approach a research problem and the need to be persistent to accomplish any goal and arrange work in a meaningful ways.

Second, I also wish to thank my instructor, Engr. Dr J. O. Awulu, Mr. Jiya Eli, Mr. Nanshuwan, Victor Datom and Mr. Onmonya Gabriel who stood by my side as I worked throughout this research project.

Third, I wish to appreciate my sponsors: Dr Edwin Ogwuche and family, Dr Augustine Ogwuche and family, Mr. Isaac Ejembi and family, Mr. John Ogwuche and his special wife, Dr Oche Ogwuche, Mr. Francis Oko Ogwuche, Engr. Ajefu F. A and Dr Alice Ogbole for their financial, moral and material support. Without this, the goal would not have been possible.

Forth, I received tremendous support from friends and colleagues throughout this process. This journey would not have been possible without the sacrifices that were made by those individuals on my behalf. During this research project, I have gained a deeper appreciation of the gift of love and joy.

Fifth, my heart-felt gratitude goes to my family: Mr. Ogwuche Ejembi (late) and his two wifeôs Mrs. Ojo Ogwuche and Mrs. Anne Ogwuche. My brothers and sisters: Mr. Francis, Mrs. Mercy Elaigwu, Mrs. Elizabeth Amali, Mrs. Rose Abalike, Mrs. Comfort Aba, Dr Edwin, Dr Augustine, Mr. David, Mr. Gabriel, Mrs. Alice Clement Aba, and Mrs. Regina Ademu. Each of them have contributed one way or the other by words of advice and encouragement, moral, material and financial support which have help me to pursue my goal.

Last, but not least, is a special thanks to Mrs. Ugwu in PG school which on her part contributed known and unknown to make my dream come true.

Abstract

Over the years, we have had complaints of people copying someone elseŝ project directly as his/her own. Therefore, this project is aimed at solving the problem of copyright as identified during seminar presentations, project defense etc in schools. To solve the above mentioned problem, this application is developed to capture relevant information of all project works undertaken and completed in the department. The application is an off-line tracking system; it will be installed on the user computer for use. The system will track projects by inputting some parameters of a new project to ascertain whether the work has been done by somebody especially from the Computer Science department, University of Nigeria Nsukka (UNN). This system would help supervisor to assign topic to students using recommendations from the previous project. Object Oriented Analysis and Design (OOAD) methodology was used for the analysis and design while Java and MYSQL was used to implement in a window 7.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.0 Introduction

Over the years, there have been several issues of students going to the departmental library to pick a copy of project written by another person, not as a guide but to submit the same topic as a research proposal to his/her supervisor without the knowledge of the teacher or supervisorôs recommendation. When approved, the student goes ahead to copy word after words as his/her own. And these are plagiarism which is against the academic law. Therefore, this project will serve as database for the department of computer science to check mate these abnormality by student. To this end, this project is aim at solving the problem of copying someone else project as his/her own as identified during a seminar presentation in the department of computer science. To solve this problem mentioned above, monitoring software will be designed, to capture relevant information of all project works undertaken and completed in the department. This project will track student by querying the database, using text from the students research work through some parameters of the new project to ascertain whether the work has been done by somebody else. This system would help supervisor to assign topic to students by the recommendation from the previous project. The University Student project Tracker (USPT) will be designed using Java and MYSQL implemented in a Windows 7. This USPT is a standalone application that will help to track the project work from the database through the search engine. In this case, the entire project that undertaken and completed in the department will be stored for records purpose or referential purposes. The USPT is an off-line tracking system that will help to track the project work by comparing the proposed project with the one stored in the database which is off-line.

The USPT will help the department of computer science to track project works, documents, assignments and research papers. They can check the contents of a projectôs originality before accepting or rejecting it. When a project is loaded in the USPT, it goes through its content to ascertain the percentage of text copied or the similarities with existing projects through the search engine.

1.1 Statement of the problem

Over the years, we have had complaints of people copying someone elseôs project directly as his/her own. This has led to lack of originality in several researches, increased level of plagiarism and laziness on the part of researchers.

1.2 Aim/ Objectives of the project

The aim/goal of this work is to design a tracking system for the department of computer science that will serve as a database for storing project work done by students. Specifically the aim/objectives of the project are:

1. To identify the original owner of a project work.
2. To identify the percentage of text copied or similar in a project
3. To help supervisor in assigning topics to students using the recommendations of the tracker.
4. To provide a database of the project works done by student of the department for records purpose or referential purposes.

1.3 Significance of the project

1. This project has a greater potential for the department of computer science as it will help them before the defense to track any students who copied someoneôs work directly as his/her own.
2. It will also help supervisor in assigning topics to students using the recommendations of the tracker.

Chapter 2: Literature Review

2.0 Introduction

This chapter provides an over view of technologies and the tools used in designing and development of this project.

It also reviewed some works on plagiarism which provided a better understanding, problems and solutions which other researchers have attempted to tackle and profer a way forward.

Finally, it gives the summary of the main points which are essentially, the concerns of earlier efforts which helped and to chart the way forward for this work, and to draw from the contributions of earlier works.

2.1 Theoretical Background

The technologies used in this project work are Java and MYSQL. [1] Java is a stand-alone application that was used to develop this University Student Project Tracker (USPT) application. The USPT is developed using frame and panels. A frame is the basic window or container used in graphics user interface (GUI) applications. It has a border and a title. Other components are added to the frame. Components like: Menus, Panels, Label, TextArea, ComboBox, Button, Split Panes, and Scroll Panes etc. A frame is use for the creation of user interface and it holds the user interface components. The javax.swing.JFrame class is used for creating the frame. Figure 2.1 is the home page of this project, developed using frames and Panels.

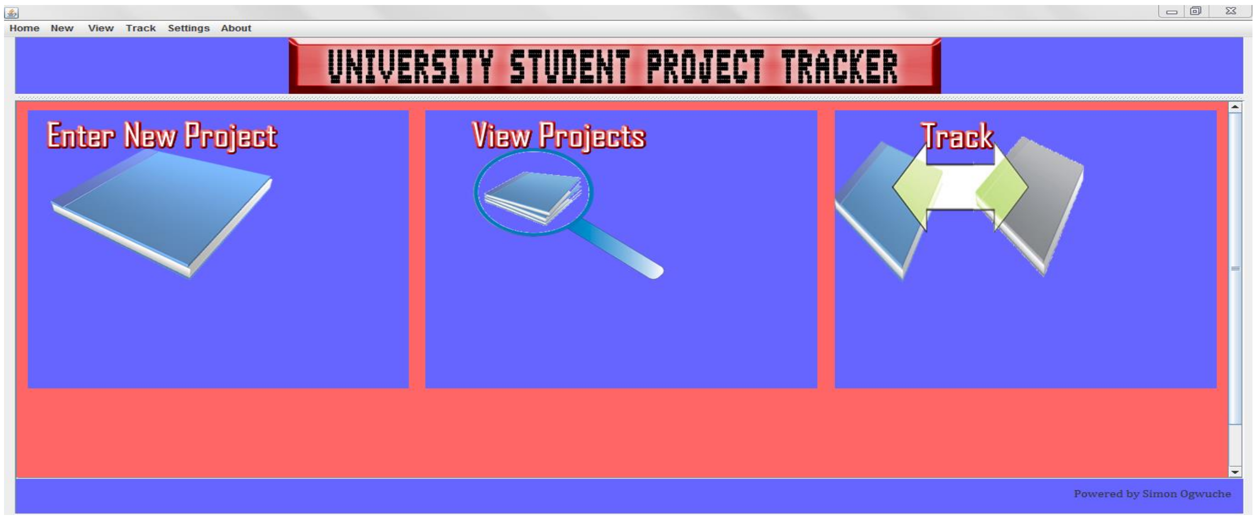


Figure 2.1 Home Page of USPT.

The major Java elements for constructing the GUI are (JMenu, JButton, JRadioButton, JLabel, JList, JTextField, JTextArea, and JScrollPane). From the frame above, we have the main Menu which contains other sub menus where user can navigate within the applications. The following procedure describes how to create a menu:

- Create a JMenuBar object, and set it into a menu container, such as a JFrame.
- Create one or more JMenu objects, and add them to the menu bar object. Object like: Home, New, View, Track, Settings, and About.
- Create one or more JMenuItem objects, and add them to the menu object. E.g Settings in JMenu has a JMenuItems: Update Projects and Delete Projects as there menu items.

The javax.swing.JMenu, javax.swing.JMenuItem classes are used to create menus. [2]

Next on the frame is a container that holds a banner (the title of the project) follow by desktop Panels that holds Labels and icons in the project. These desktop panels can also help user to move to next page within the project when clicked. Below is the codes used in constructing the home page of USPT, figure 2.1.

```
importjavax.swing.*;
public class Homepage extends javax.swing.JFrame {

public static void main(String args[]) {
```

```
            public void run() {

                new Homepage().setVisible(true);

    private javax.swing.JLabel jLabel1;

    private javax.swing.JMenu jMenu1;

private javax.swing.JMenuBar jMenuBar1;

    private javax.swing.JMenuItem jMenuItem1;

 }

       }

    }
```

[3] MySQL Relational Database Management System (RDBMS) was used to create the database. Mysql is also used for accessing, maintaining, and managing the data in the database. The form in figure 2.2 contains a table that is used to display and edit information in the form of a grid. The table was created by java using the syntax: javax.swing.JTable class. JTable does not store the data, but only displays the data from the table model. Mysql stores the data in the database and make it available when needed for user. Below is sample of form of table in the USPT tracker interface, showing how the student project that was stored in database is shown in the interface.

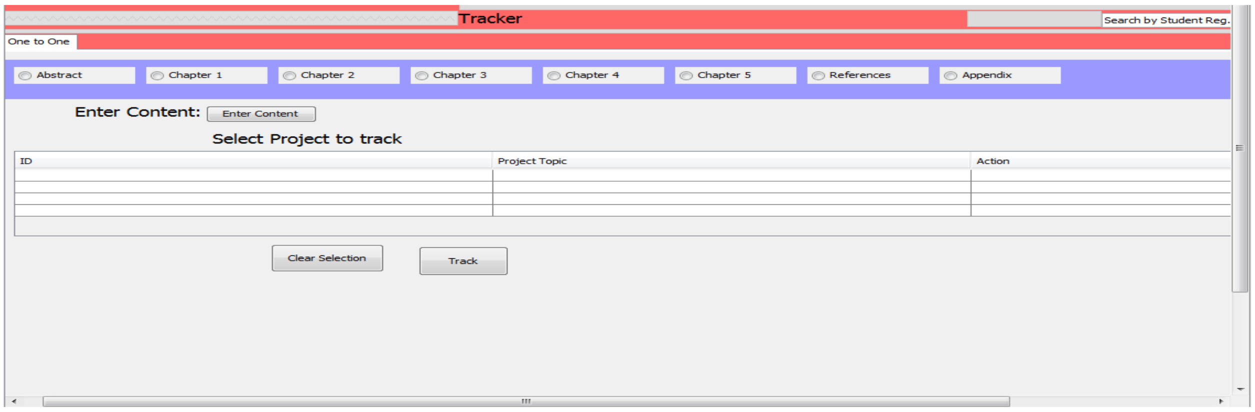


Figure 2.2 An interface showing how a student project is entered into a table for tracking.

Another Java technologyô s used here is Java Database Connectivity Application Programming Interface (JDBC API) which enables the user to interact with the databases.

The JDBC provides a standard interface for accessing a relational database. Modeled after the open database connectivity (ODBC) specification, the JDBC package contains a set of classes and methods for issuing SQL statements, table updates, and calls to stored procedures [2].

2.2 Review of Related Literature

2.2.1 What is Plagiarism?

[4] Stated that Plagiarism is the act of taking another person's writing, conversation, song, or even idea and passing it off as oneô s own. This includes information from web pages, books, songs, television shows, email messages, interviews, articles, artworks or any other medium. Whenever you paraphrase, summarize, or take words, phrases, or sentences from another person's work, it is necessary to indicate the source of the information within your paper using proper citation. It is not enough to just list the source in a bibliography at the end of your paper. Failing to properly quote, cite or acknowledge someone else's words or ideas with citation is plagiarism.

2.2.2 Effect of Plagiarism

When a student could not make the necessary efforts and gain an undeserved grade, such a student is not different from an athlete who cheats and takes banned drugs to gain an unfair advantage. Student, who through plagiarism graduates, lacks the important knowledge and skills. Such acts reduce intellectual capability. The person loses the chance to develop skills which make for a productive life [5]. Such a person lacks the Critical, creative, and independent thinking. The low productivity leads to lack of development since new ideas are not found.

2.2.3 Implication of plagiarism on academic

According to [6] ñInstructors typically have little tolerance for plagiarism because they want you to learn and earn your grades fairly and honestly. If you are caught plagiarizing, academic sanctions can include a lower grade, failing the course or dismissal from an academic major. An incident of plagiarism can also diminish oneȏs chances of a good reference from the instructor for a scholarship application, study abroad program, graduate school, internship or graduate assistantship. Your degree can be revoked if it is later discovered that you plagiarized a capstone project, thesis or dissertationȯ.

[7] developed a web-based prototype system for analyzing up to 1 million documents; He uses character strings rather than word strings as the basis for a fingerprint, and quotes effective lengths of 30-45 characters. His work focuses on methods of selection to produce a reduced size fingerprint from the full set. These systems address copy detection on a very large scale, from 1 million to tens of millions of documents. As the numbers of document increases, it becomes difficult to detect plagiarism. For instance, the SCAM system was evaluated just by classifying a sample of 50 texts on a subjective basis.

[8] explored the ranking and fingerprinting approaches for detecting plagiarism of text. Text plagiarism involves copying parts of manuscripts, papers, and documents. These approaches have a common preprocessing stage that includes case folding, Stemming (removing prefix/suffix from words), stopping (removing common words), and term parsing (removing whites pace, punctuation, and control characters from the document). The ranking approach consists of two stages to find documents similar to a query. In the first stage, documents are indexed. In the second stage, terms in the query document are matched against the indexed terms of each collection document, and a similarity score is calculated. Documents are ranked by decreasing similarity score for presentation to the user. The fingerprinting future generate fingerprint of the document. A document fingerprint is a collection of integers that represent some key content of the document. Each of these integers is referred to as a minutia typically; a fingerprint is generated by selecting substrings from the text and applying a mathematical function to each selected substring. This function, similar to a hashing function produces one minutia. The minutiae are then stored in an index for quick access when querying. Using experiments on two collections, they demonstrated that the identity measure and the best fingerprinting technique are both able to accurately identify plagiarized documents.

[9] tried to classify metrics used for plagiarism detection. They proposed two ways of classifying metrics. First classification is based on the number of documents involved in the metrics

calculation process and second one is based on computational complexity of the methods employed to find similarities. In the first classification, metrics can be classified as singular or paired metrics and as corpal or multi-dimensional metrics, depending on how many documents are preceded, and depending on the set of documents involved in proceeding, respectively. A corpal metric operates on an entire corpus of documents. A multidimensional metric operates on a chosen number of documents. In the second classification, metrics can be classified as superficial metrics and structural metrics. A superficial metric is a measure of similarity that can be gauged simply by looking at one or more documents. In this case knowledge of the linguistic features of natural language is not necessary. A structural metric is a measure of similarity that requires knowledge of the structure of one or more documents.

[10] published a work on Plagiarism Detection across Programming Languages. They proposed an approach to detecting plagiarize source code copy from one programming language to another. They proposed an approach called XPlag, to detect inter-lingual plagiarism by comparing the structure of intermediate code produced by a compiler suite. A compiler suite is a compiler that supports more than one language. The XPlag mechanism comprises two stages. In the indexing stage, all programs in the collection are converted into tokens, and token information is stored in an inverted index. In the detection stage, source code is used to query the index and produce a list of programs in the collection, ranked by decreasing similarity. At the end of their work, they tested the approach against three collections using ground truth developed from an existing state-of-the-art plagiarism detection system and through manual comparisons. The results show that their approach detects plagiarism with reasonably good precision for all the test collections. More importantly, it succeeds in detecting plagiarism across languages.

[11] worked on Intrinsic Plagiarism Detection. Their work centered on detecting plagiarized passages within a document or books where the source-document is not available in digital form. The method of identifying potentially plagiarized passages was to analyze a document with respect to variations in writing style. That is, a document is divided into ñnaturalò parts - sentences, paragraphs, or sections, and variation in the style and features of these parts were analyzed. To measure the writing style, their approach was to quantify text statistics, which operate at the character level; syntactic features, which measure writing style at the sentence-

level; part-of-speech features to quantify the use of word classes; closed-class word sets to count special words; structural features, which reflect text organization, and the averaged word frequency class. In their experiment, t he size of a part was chosen between 40 to200 words. More than 450 instance documents were generated each of which contain between 3 and 6 plagiarized passages of different lengths. During the plagiarism analysis these instance documents were decomposed into 50 - 100 passages from which the feature vectors were computed. Their result showed an achievement recall values of 85% with a precision of 75%.

[12] worked on Plagiarism Detection through Multilevel Text Comparison. The work was actually an implementation of plagiarism detection tool provided within Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS). The algorithm leverages the plagiarist behaviour, which is modeled as a combination of 3 basic actions: insertion, deletion and substitution. Their approach was based on the philosophy that the plagiarist may insert, delete or substitute a word, period or a paragraph at any level of the document structure. The detection procedure consist two main steps: document structure extraction and plagiarism function calculation. The pre-processing modules break the document into tokens and identify words, periods and paragraphs called chunks. A document structure tree is created, where each node identifies a chunk, and each child of the node represents a contained chunk. Chunks were then compared from the highest level of the tree to the lowest through the evaluation of the plagiarism function. The plagiarism function yields the minimum distance between two strings defined in terms, insertion, deletion, and substitution. Given two character strings, the idea was to find the smallest set of primitives that applied to one string. Given two chunks they will be considered equivalent for the edit distance calculation if their plagiarism function is greater than a certain threshold. Though they developed the algorithm, it was not tested in the work for its accuracy and strength.

[13] worked on Plagiarism Detection without Reference Collections. The work was an approach to detect plagiarized texts within a document of which no reference collection is given against which the suspicious texts can be matched. Intrinsic Plagiarism detection is related to the identification of an author's writing style, for which various measures have been developed in the past.   They used vocabulary richness to detect plagiarism in a document. To analyze the phenomena of intrinsic plagiarism detection, they constructed a base corpus from which various

application corpora can be compiled each of which model plagiarism of different dialects. If it is discover that there are inconsistences in the vocabulary across different paragraphs and vocabulary richness seems to be higher than the average strength of the author. There experiments revealed that the introduced averaged Word frequency class of their work outperforms other well-known measures in this respect.

[14] worked on Computer Based Plagiarism Detection Methods and Tools. There work was based on two principle methods for plagiarism prevention, and methods for plagiarism detection. Some examples of methods in each class are as follows: plagiarism prevention ï honesty policies and/or punishment systems, and plagiarism detection ï software tools to reveal plagiarism automatically.

The prevention and detection methods are used to achieve one common goal ï to fight against plagiarism. To make this fight efficient, system approach to plagiarism problem solving is needed, i.e. it is needed to combine plagiarism prevention and detection methods. To achieve momentary, short ï term positive results plagiarism detection methods must be applied at problemôs initial stages, but to achieve positive results in long ï time period, plagiarism prevention methods must be put into action. Plagiarism detection methods can only minimize plagiarism, but plagiarism prevention methods can fully eliminate plagiarism phenomena or at least to a great extent decrease it. Unfortunately, plagiarism prevention is a problem for society as a whole, i.e., it is at least national wide problem which cannot be solved by efforts of one university or its department. Plagiarism detection usually is based on comparison of two or more documents. In order to compare two or more documents and to reason about degree of similarity between them, it is needed to assign numeric value, so called, similarity score to each document.

[15] worked on Automatic Plagiarism Detection based on n-Grams comparison. They splited a suspicious document into sentences and the sentences were further splited into n-grams. A reference document corpus was also splited into n-grams words, and each suspicious document sentence was searched in singleton over the reference documents. In other to determine whether a sentence from a suspicious document was plagiarized from a sentence from the reference document, the set of n-gram words or both suspicious document and the reference document were compared.

The aim of their experiments is to define the best n-gram level to detect plagiarism cases. To achieve this, they used a corpus composed around 750 notes as reference corpus. 444 from the 942 news paper notes compose the suspicious documents set. a 5-fold cross validation process was carried out and Precision, Recall were calculated. The result shows that n=1 (unigram), a good Recall was obtained; for n=2, (Bigram) the search was more flexible, allowing better Recall, while for n=3 (Trigram), based search was more rigid, resulting in a better Precision. For n=4and above, the result produced a rigid search strategy with lowest Recall values. This means that minor changes in a plagiarized sentence avoids its detection, resulting in the lowest Recall values.

[16] worked on Intrinsic Plagiarism Detection Using Character n-gram Profiles. The task of intrinsic plagiarism detection deals with cases where no reference corpus is available and it is exclusively based on stylistic changes or inconsistencies within a given document. With this method, a document is automatically segmented according to stylistic inconsistencies and decides whether or not a document is plagiarism-free. A set of heuristic rules was used that attempt to detect plagiarism on either the document level or the text passage level which also reduce the effect of irrelevant stylistic changes within a document. The proposed approach was evaluated on the corpus of the first international Competition on Plagiarism Detection with over 70% plagiarized-free document correctly classified at the document level.

A Hashing and Merging Heuristics for Text Reuse Detection was developed by [17]. In this study, three steps of seeding, extension and filtering for text alignment were used. For seeding, they used *n*-grams character with a variant of the Rabin-Karp Algorithm for multiple pattern searches. Subsequently, an elaborate merging mechanism was incorporated with several cases to align with the individually found seeds. A short filtering step is then used to remove extraneous passages. The test results from Plagdet score revealed 0.65954 and 0.73416 respectively for corpora 2 and 3 in the final run.

In a related study, [18] experimented the Short Stories Corpus which comprises of four different texts reuse scenarios namely: no-plagiarism, story-retelling, synonym-replacement and character-substitution. Among these scenarios the most interesting one is story retelling - through it, they find patterns of textual similarity between story retellings. They use Grimm brotherôs fairy tales as described in the Project Gutenberg as the source of their documents. The corpus consists of 200 pairs of documents, with 50 document pairs for each type of text reuse. The empirical observation shows interesting patterns of textual identity within the corpus. Furthermore, plagiarism detection using text reuse approaches demonstrates how tasking detection of various groups within the corpus are.

2.2.4 Lessons

Several literatures that focus on detecting plagiarism were reviewed. The works generally are of two types: detection of plagiarism in documents with known or available reference source and detection of plagiarism in document whose reference sources are not available in digital form. Some of the commonly used methods in many literatures to detect are: writing style of a particular author and n-gram approach which segment and match the suspicious text to a reference document.

Though several algorithms were used with N-gram approach, but none of the works used sequential search algorithm. USPT was developed using sequential search algorithm with 3-grams. This method worked by comparing the proposed project with the existing one through a text matching formula or rules which then later determine the percentage (%) of text copied.

Chapter 3: SYSTEM ANALYSIS AND DESIGN

3.0 Introductions

The methodology used for this work is; Object Oriented Analysis and Design (OOAD). This design applies object-oriented concepts to develop and communicate the architecture and details of how to meet the system requirements. Also, the methodology uses a string of text for the comparison.

The UML technique used in this study includes the following: use case diagram, class diagram, System Architecture, flow diagram, and database designed tools.

3.1 Description of the Existing System

Over the years, the department of computer science in the Faculty of Physical Sciences University of Nigeria Nsukka (UNN) has been accessing and storing all projects work undertaken and completed in the department as hard copies in the departmental library. Due to this system of storing hard copies of the research work and projects in library, several students who carry out a final year project or postgraduate research exploit the opportunity by going to the library to pick a copy of already written project. The projects the students pick are not use as a guide, but rather as a topic for research proposal.

In many cases, the supervisor is not aware that such topics have already been researched, and therefore approves the topic. When this happens the student goes ahead to plagiarizes the project word for word. Because projects are stored in hard copies, it is very difficult to know the areas they have copied since it is in hard copies. Also, it is difficult to search since the projects are many.

3.1.1 Problem facing the System

The existing system possesses the following shortcomings:

1) It is difficult to search very large project write ups.
2) Looking for a particle author, supervisor or topics takes a longer time.
3) It required a lot of space for storing the project because they are in hard copies.
4) Over time some projects are lost when some students borrow from the library without proper documentation by the librarian.
5) Recommendations by the student are difficult to access and implement.

3.2 Analysis of the Proposed System

In this proposed system sequential search algorithm is used for this work.

The method worked by comparing the proposed project with the existing ones in electronic format through a text matching formula or rules, the system then determines the percentage (%) of text copied and indicates whether the compared project is duplicated or unique. Below is details description of how the system (University Student Project Tracker (USPT)) works.

3.2.1 Description of the USPT System

By default, the system user interface has the following: home page menu include: home, new, view, track, setting, and about,

From the user interface, new projects can be entered into the system by clicking ónewô button in a menu bar or óenter new projectô icon. When you click óNewô, another user interface will appear which the user uses to complete the fields that are required for comparison then, submit the project into the system. For easy entering of the project into the system, all projects written and completed by students in the department are to be submitted electronically to the project coordinator of the department, where it can be stored and used to browse the contents of the files.

All projects saved in the system can be viewed through óviewô in the menus bar of the home page of the USPT. óAboutô in the menus bar is used to show the abstract of the project.

Finally, project can trace by clicking ótrackô in the USPT home page. Tracking is done on one-to-one matching of projects; matching is done by comparing a string (text) of the proposed project with string of existing project that is saved in the system. To do this, first the user selects the radio button of the particular aspect she/he wants (abstract, chapters, reference, and appendix), then, click on enter content and a menus appear, browse for the files then click submit. After that, select the project to be track then click track.

When the tracking finishes, the system displays the result in percentage (%) of duplicated and percentage (%) of uniqueness of the project. Also, there is an easy way to search for project to be track when one have over a thousand of projects in the database and want one particular project; type student registration number or topic name, and the project will be displayed.

The proposed system has the following functionalities:

1) Ability to Search the contents of very large projects.
2) Find and return particular project from the database by simply entering student registration number or topic in a search engine.
3) Compare the proposed project with existing projects in the database.
4) Serves as database of student project for referential purposes.

3.2.2 Use Case Diagram

The use case diagram of the USPT figure 3.1 contains two actors which are User and System (USPT) and the roles played by each of them.

The roles played by User are:

- ➢ Upload Project
- ➢ Compare Project for plagiarize text
- ➢ View Project

The roles played by system (USPT) are:

- ➢ Upload Project
- ➢ Compare Project for plagiarize text
- ➢ View Project

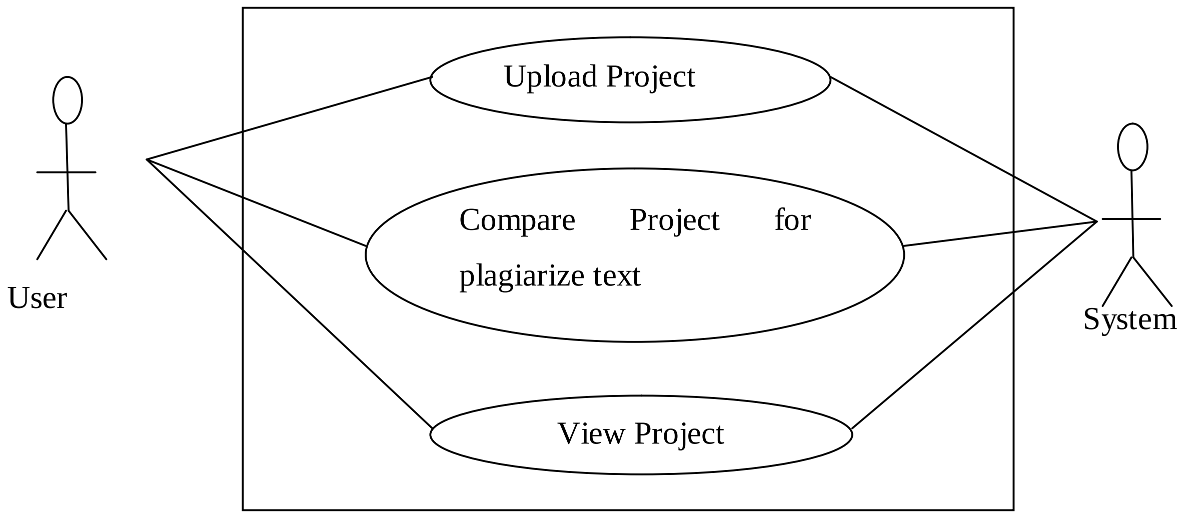


Figure 3.1: Use Case Diagram of USPT

3.2.3 Class Diagram

The class diagrams in figure 3.2 shows a set of classes in the system, and the associations and relationships between the classes. Class nodes contain a listing of attributes and operations. The class diagram was used for showing the structure of the system and what needs to be programmed. Most UML case tools can generate code based on the class diagram [2].

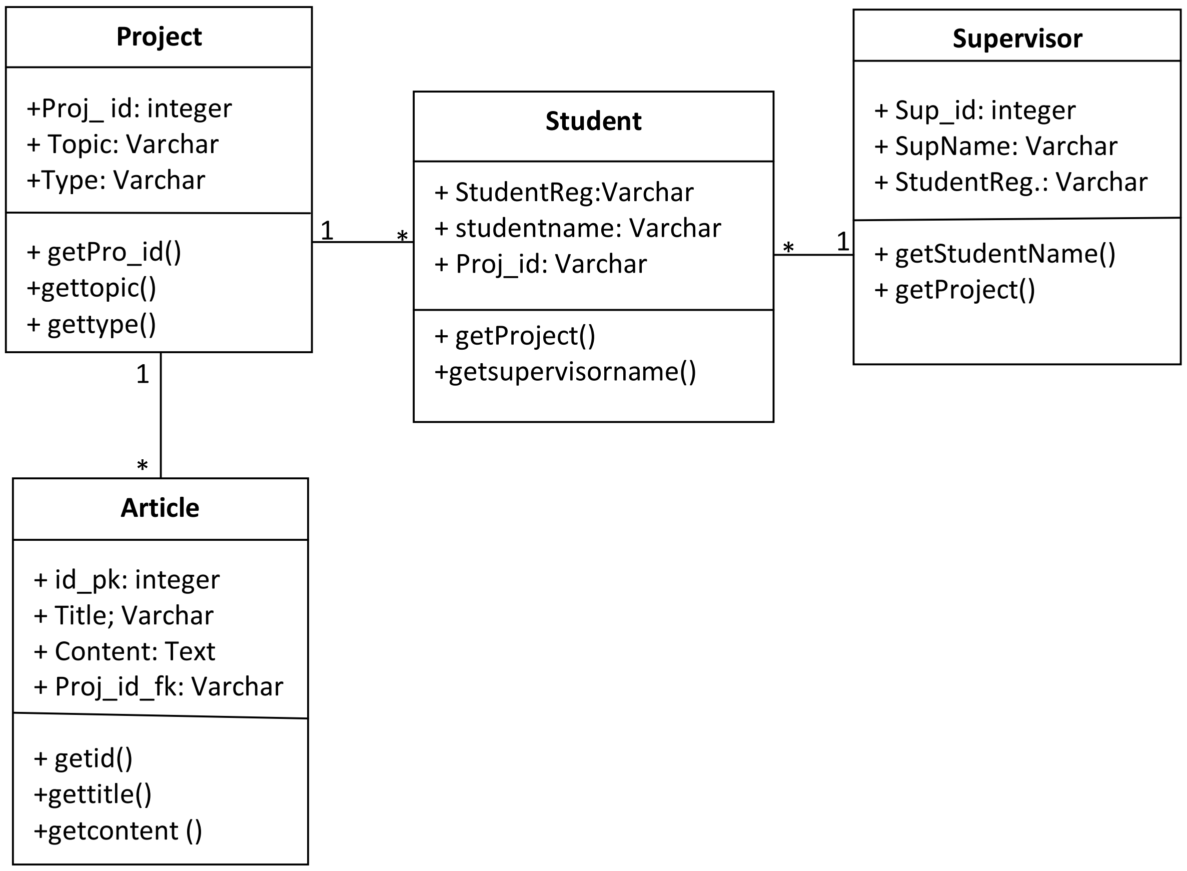


Figure 3.2: Class Diagram of USPT

3.3 System Architecture

The architecture of the USPT is 2-tiers. The first tier is the user interface and it is designed using java swing. The second tier is application program where processing and logic operation take place. Java code is used for comparing project against plagiarism and matching text alongside with the database for storing data in the system and it is implemented using MySQL database. Figure3.3 shows the system architecture for the USPT.

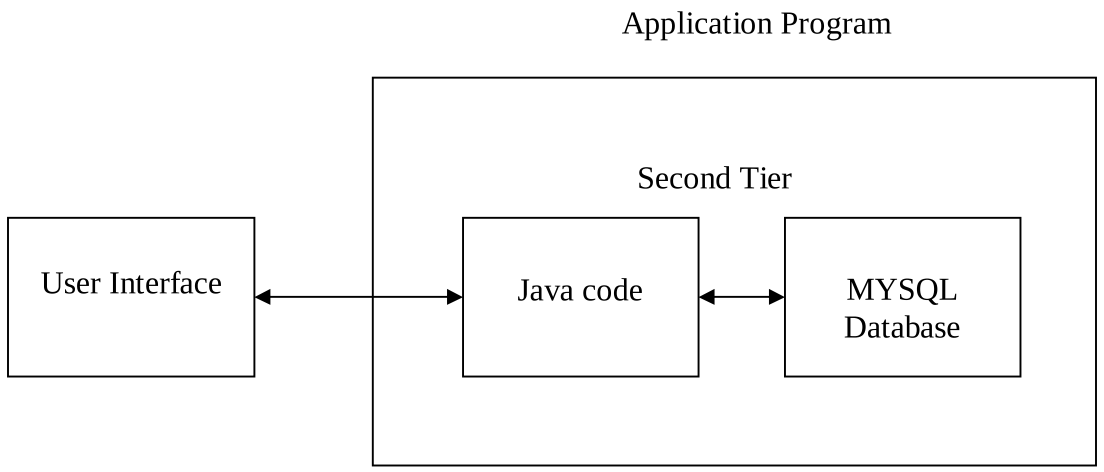


Figure 3.3: System architecture for the USPT.

3.3.1 Database Design

The DBMS used in this USPT is MySQL. The system has four basic entities: Articles, Projects, Students and Supervisors. The tables for storing records about each entity are describes below.

Table 3.1: Article

Field Name	Data Type	Size	NULL	Description	Action	Extra
Id_pk	Int	5	No	Serial number	Pk	Auto increment
Title	Varchar	45	No	Title of project	Required	
Content	Text		No	Content of the project e.g abstract, chapters etc.	Required	
Proj_id	Int	5	No	project id	Fk	Referencing project table

Table 3.2: Project

Field Name	Data Type	Size	NULL	Description	Action	Extra
Proj_id	Int	5	No	project id	Pk	Auto increment
Topic	Varchar	45	No	Project topic	Required	
Type	Varchar	6	No	Type of project e.g B.Sc /M.Sc/PHd	Required	

Table 3.3: Student

Field Name	Data Type	Size	NULL	Description	Action	Extra
StudentRegNum	Varchar	15	No	Student registration number	Pk	
StudentName	Varchar	45	No	Student name	Required	
Proj_id	Int	5	No	project id	Fk	Referencing project table

Table 3.4: Supervisor

Field Name	Data Type	Size	NULL	Description	Action	Extra
Sup_id	Varchar	8	No	Supervisor identity	Pk	Auto increment
SupName	Varchar	45	No	Supervisor name	Required	
StudentRegNum	Varchar	15	No	Student registration number	Fk	Referencing students table

3.3.2 Input Design

Input design has to do with technique or tools use to design input data into the system for processing. This section describes the way data about students projects in the department are entered into the system. Below Figure 3.4 is input for entering new projects into the system.

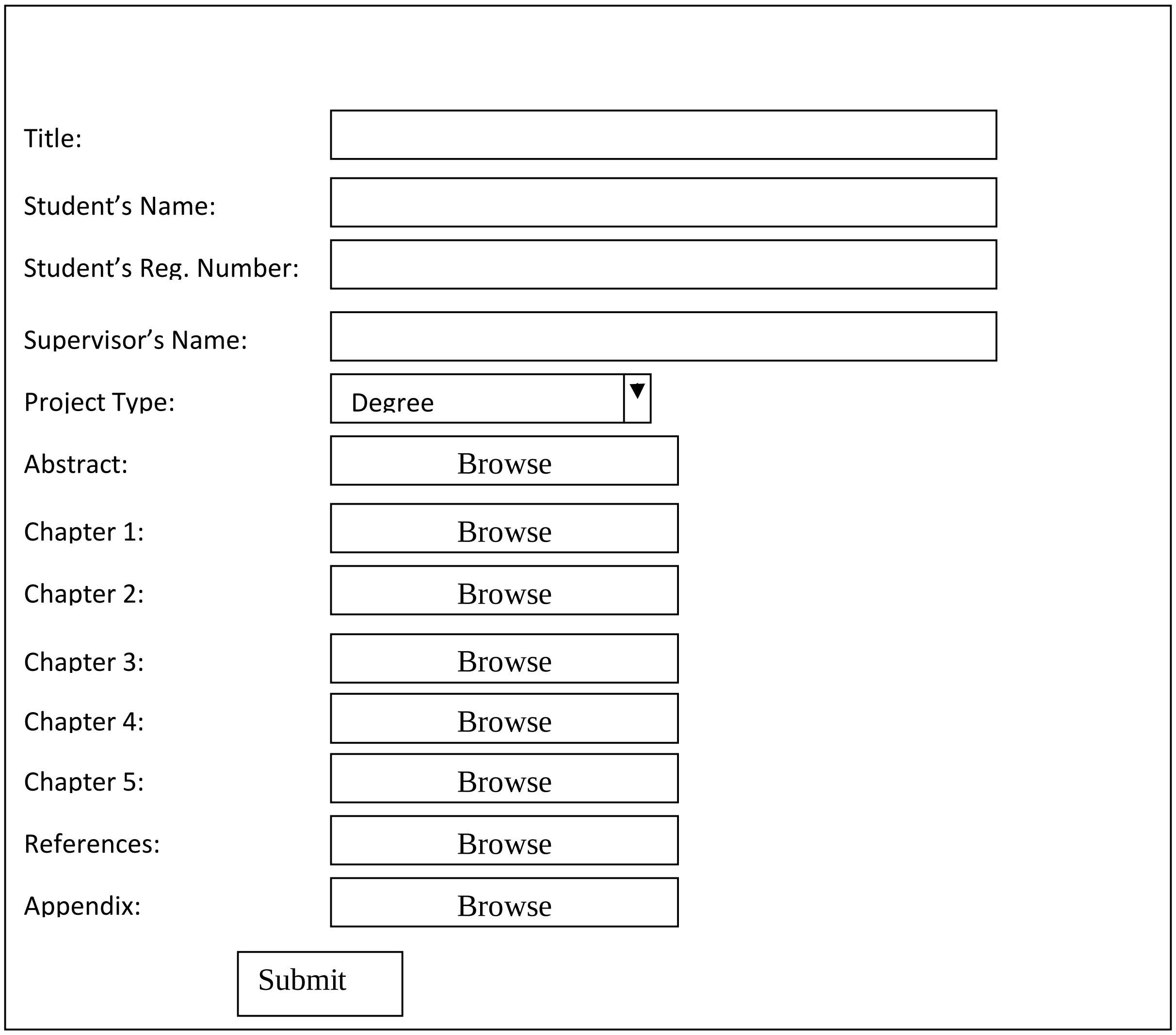


Figure 3.4: Input form of USPT.

3.3.3 Output Design

This section shows the result of processing from the input. Below Figure 3.5 to 3.7 is various output design as viewed in the USPT application:

- Title
- Abstract or Chapters
- Tracker.

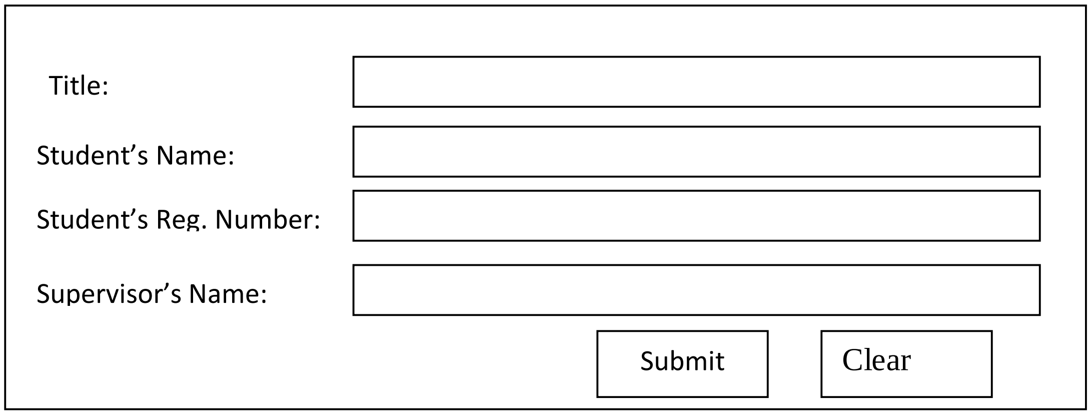


Figure 3.5: Form for Project Title.

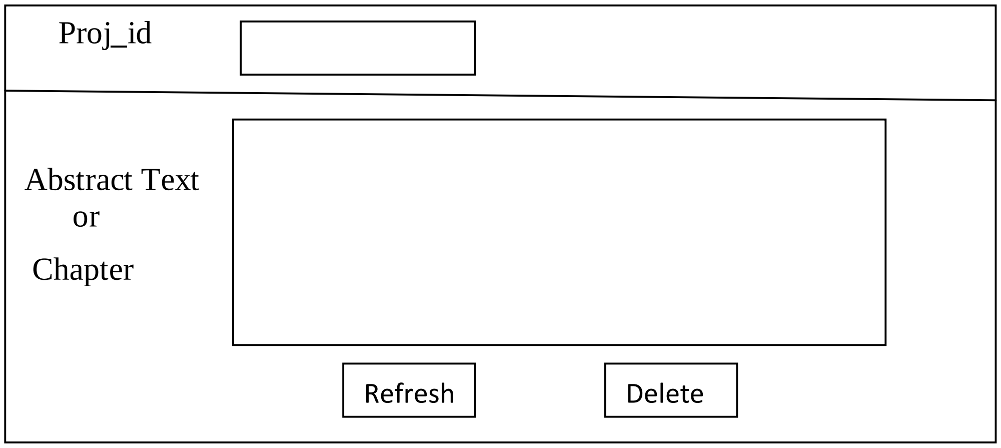


Figure 3.6: Output form for Abstract and Chapter.

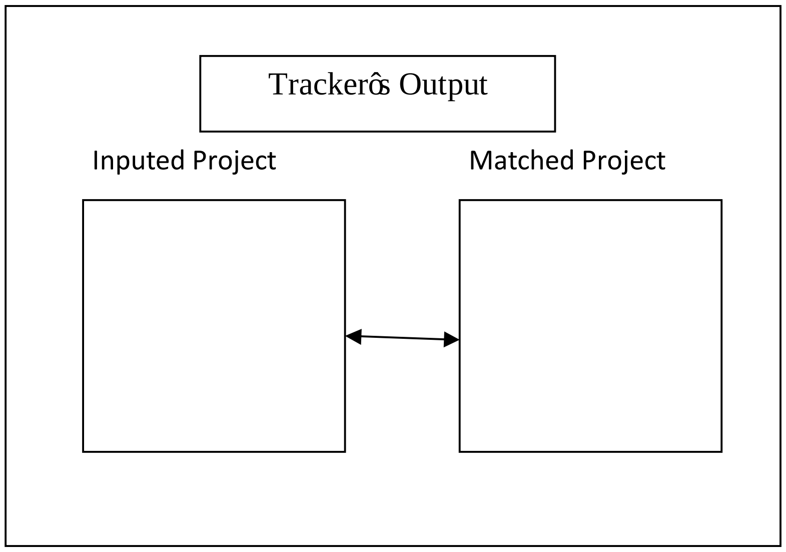


Figure 3.7: Trackerôs Output form

3.3.4 Algorithm Design

The algorithm used in this work (USPT) is Sequential Search Algorithm (SSA). SSA is used in this work because it searches text in a liner form ñstarting at the beginningò and till the end; then stops. Considering two conditions which terminate a search in the algorithm:

i. The document is found.
ii. The entire array has been scanned, to the end.

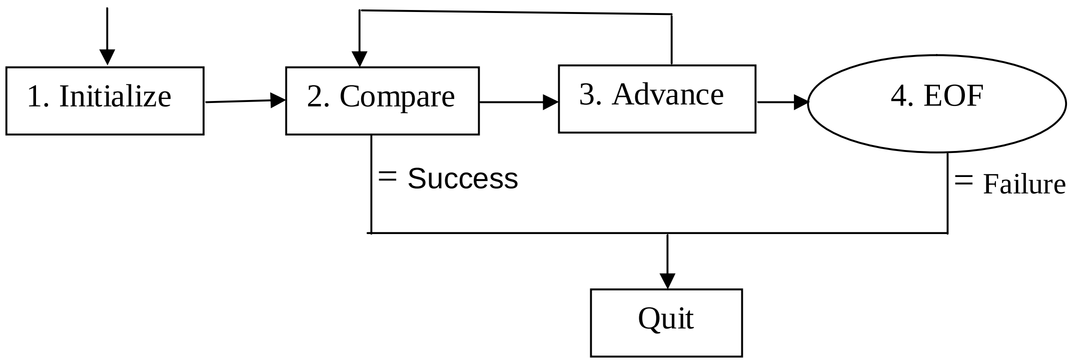


Figure 3.8: A flow diagram of the algorithm for USPT

The Sequential Search Algorithm for USPT

// Match the selected string of Proposed Project (PP) to string of Stored Project (SP) to get the selected strings

for (int i = 0; i < arr.size(); i++) {   // int i is a start value of loop and arr.size is end value of loop.

matchText(Area2, word2, arr.get(i).toString()); // search text and match the 3-grams proposed string with the project in the database, convert it to string.

}   //end loop.

arr as used in the algorithm (arr = array) used to hold set of items like a list of text. When the program run, the array stores result of 3-grams in a temporal folder after which the memory is lost.

Details description of this algorithm for USPT

<u>Track Operation</u>

i. Check that proposed project is not empty.
ii. Check that a matched project is selected.
iii. Check that a radio button is selected.
iv. Collect the matched project.
v. Apply a 3-grams approach.

<u>Database Operation</u>

i. Collect the matched and proposed project.
ii. Check if they are equal report else.
iii. Remove all new line spaces by converting them to single space.
iv. Apply the 3-grams approach.
v. Match the 3-grams proposed string to match project in the db and collect the selected string.

Here, n-gram represents the number of words to be matched, in this case, since n = 3, 3-grams means 3- words to be matched at a time.

Furthermore, from the software, it takes 3- words of the PP and match it with 3- words of SP this process continue until all the words in PP are matched and the software release result of comparison. Below is an example showing how 3- grams techniques compare project.

Rules for the Sequential Search Algorithm

- To determine the percentage (%) of duplicate, the number of matched words is divided by the total number of words matched from the main text.

  $\frac{x}{y} \text{X} \frac{100}{1}$, where x = number of matched words in project and y = total number of words in project.

- To determine the percentage (%) of uniqueness, the number of words unmatched is divided by the total number of words matched from the main text used for the match multiplied by hundred. i.e.

  $\frac{x}{y} \text{X} \frac{100}{1}$, where x = number of words unmatched and y = total number of words in main text used for the match.

If the numbers of matched words is 50% and above, then the project is considered plagiarized, else, project is unique.

Example of how 3- grams matched two projects:

A = The man is going to the church.
B = The man is going to the market.

Where A is the propose project while B is the project in the database or project stored in the system.

Iterations of Algorithm:

1 = The man is ✓

2	= going to the	✓
3	= man is going	✓
4	= to the church	X
5	= is going to	✓

Here, the sign good (✓) signify success while the symbol (X) signify failure.

From the formula above, to determine the percentage (%) of duplicate is:

$\frac{x}{y} \text{X} \frac{100}{1}$, in this example, some words are repeated e.g, man is in line 1 & 3, so, the system sort the word into a single sentence and determine the percentage duplicate.

Therefore: x = 6 (the man is going to the) and y = 7 (total number of words in A).

Percentage (%) of duplicates = 6/7 x $\frac{100}{1}$ = 86%

In analyzing the above iteration, sequential search algorithm with 3-grams approached was used. By this instance, sampled sentences, paragraphs and or the documents are grouped in three (3) words after which comparison can be made to ascertain the percentage of duplicated words and their uniqueness.

In this case, since the percentage (%) is above the limit for rules of this sequential search algorithm, this project is considered to be plagiarized.

Chapter 4: System Implementation

4.0 Introduction

This chapter contains details of the choice of development environment, the implementation architecture of University Student Project Tracker (USPT), software testing, system specifications, and documentations including the user manual and source code listing.

4.1 Choice of development Environment

JAVA language and Netbeans were used for the development of this USPT, while MySQL database management system was used for the implementation of the database.
Java was chosen for the implementation of this USPT because, Java is an object oriented programming language that can be used to develop stand-alone application [1].

4.2 Implementation Architecture

Figure 4.1 describes the implementation architecture of the designed system and the various components showing the modules/submodules, their linkages and their output data.

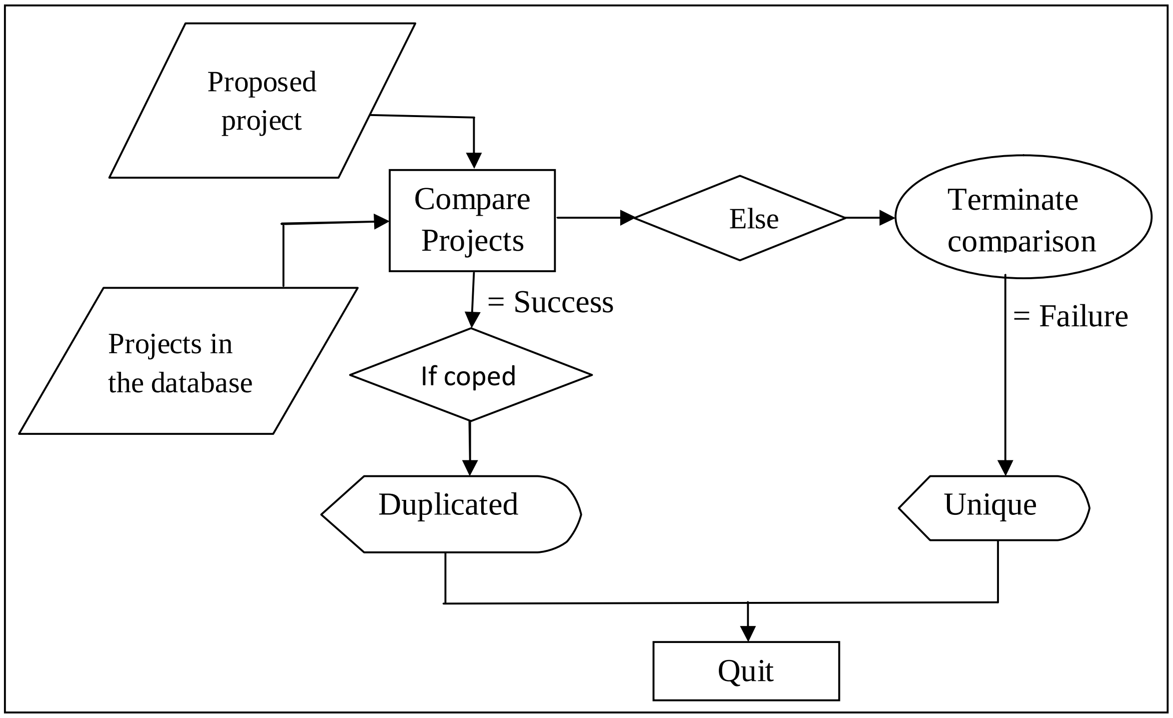


Figure 4.1: Implementation Architecture

4.2.1 System Specifications

This section provides the detailed documentation of the proposed system. It is designed for proper understanding and effective implementation of the system. It is divided into two parts:

1. Software specification and
2. Hardware specification

4.2.1.1 Software Specification

The proposed system requires the following components (and the description of their functions) at minimal level for effective implementation:

- **Java**. This is use for the compilation of the program.
- **Database Server**. This is the server-side component deployed using MYSQL for data storage.
- **Windows operating System**. Any version of windows operating system is compactable.

4.2.1.2 Hardware Specification

The proposed system is executable via the following minimal hardware specification:

- Pentium IV Processors.
- A 128MB RAM Size.
- A 40GB hard disk capacity.
- A standard keyboard and mouse.

4.3 Software Testing

The logic behind testing is to find errors. The system is tested at every stage of its development in other to be able to detect errors and remove them immediately. The software testing is in two phases: Firstly, testing during development phases by removing bugs. Secondly, testing done by running the application on the developerȏs computer in other to know if the system meets the desired requirements

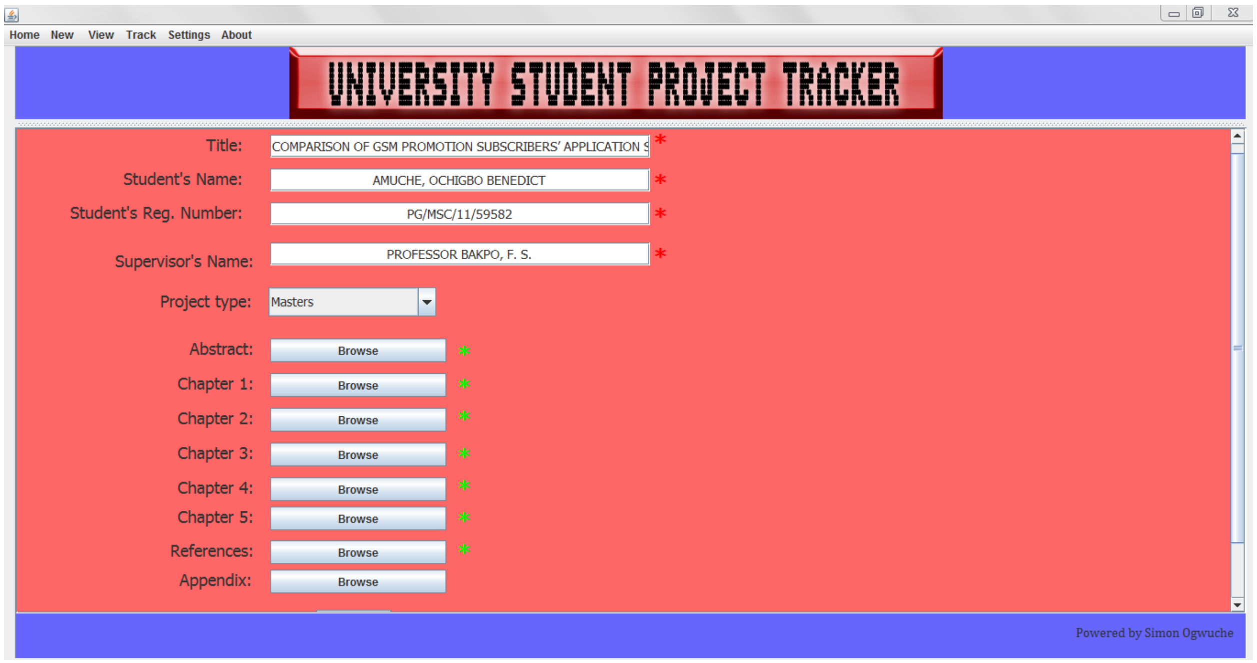
4.3.1 Testing during Development Phases by removing Bugs

Code-testing strategy examines the logic of the program. To follow this testing method, test cases were developed that resulted in executing every instruction from the program (Every path through the program was tested). A path is a specific combination of conditions that is handled by the program. For example, viewing a project from the user interface, this is not possible except the application is connected to the database before the project can be viewed.

4.3.2 Testing done by running the Application on the Developer's Computer

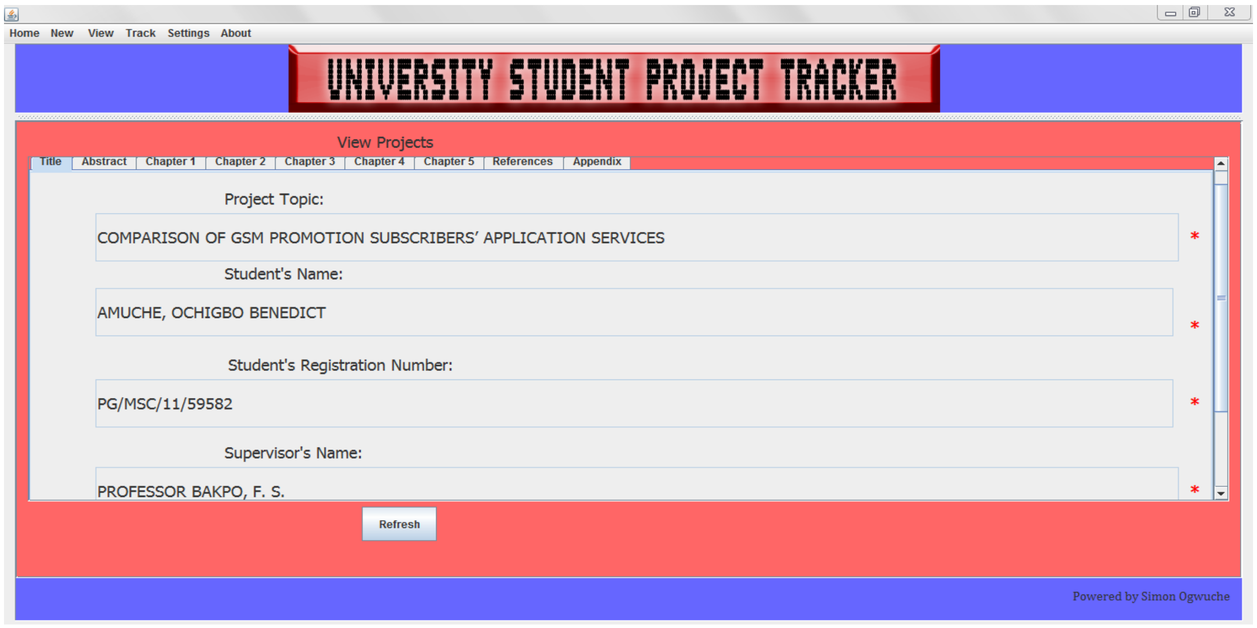
In testing the Application on the Developerôs Computer, features were designed that detect errors during the running of the program and make necessary corrections while the program is in use. This was done to track the errors that the user may encounter while using program and avoid occurrence of failure subsequently.

Figures 4.2 and 4.3 are input and output screen shots of the system. The text field has an asterisk (*), indicating that it is compulsory while the other asterisk attach to the text field (browse) change from color red to green if the content is successfully uploaded otherwise, the asterisk color remain same as default red.


Home New View Track Settings About
UNIVERSITY STUDENT PROJECT TRACKER
Browse
Powered by Simon Ogwuche

Title: COMPARISON OF GSM PROMOTION SUBSCRIBERS' APPLICATION S *
Student's Name: AMUCHE, OCHIGBO BENEDICT *
Student's Reg. Number: PG/MSC/11/59582 *
Supervisor's Name: PROFESSOR BAKPO, F. S. *
Project type: Masters
Abstract: Browse *
Chapter 1: Browse *
Chapter 2: Browse *
Chapter 3: Browse *
Chapter 4: Browse *
Chapter 5: Browse *
References: Browse *
Appendix: Browse

Figure 4.2: Screen shot showing how input is entered.


Home New View Track Settings About
UNIVERSITY STUDENT PROJECT TRACKER
View Projects
Title Abstract Chapter 1 Chapter 2 Chapter 3 Chapter 4 Chapter 5 References Appendix

Project Topic:
COMPARISON OF GSM PROMOTION SUBSCRIBERS' APPLICATION SERVICES *
Student's Name:
AMUCHE, OCHIGBO BENEDICT *
Student's Registration Number:
PG/MSC/11/59582 *
Supervisor's Name:
PROFESSOR BAKPO, F. S. *

Refresh
Powered by Simon Ogwuche

Figure 4.3: Screen shot of output showing the input entered.

4.4 Documentation

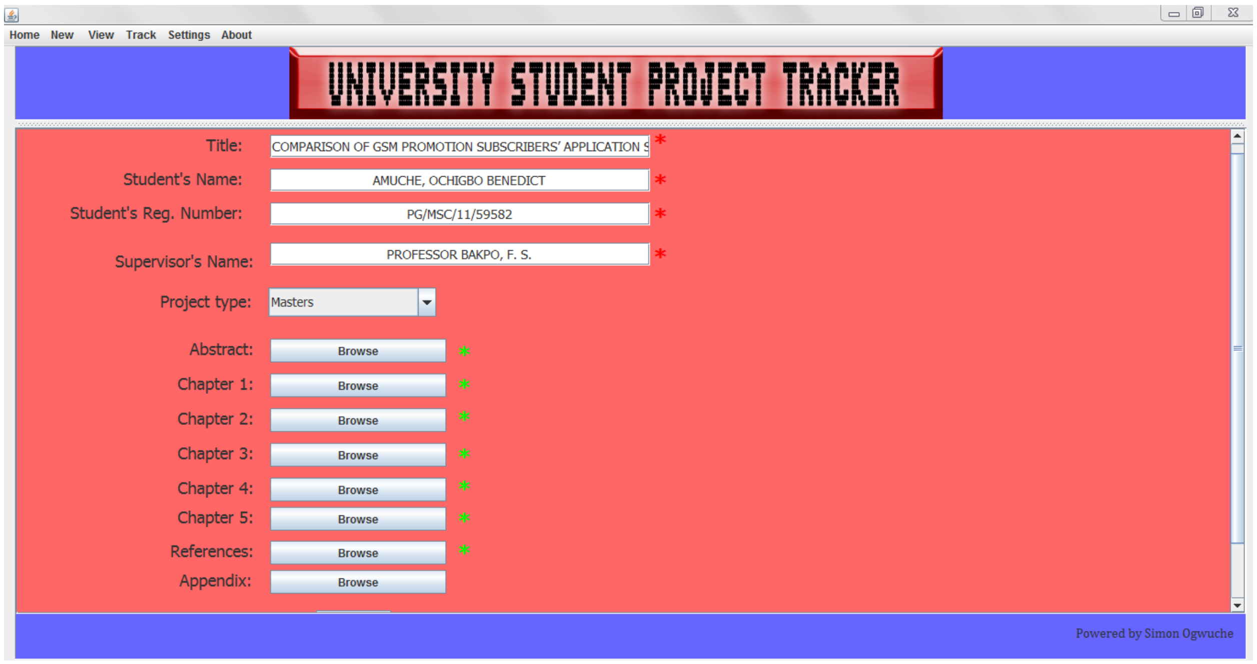
Documentation instructs the user on how to perform certain function on application. This is quite easy by asking the user to follow them exactly through a series of events. The detailed descriptions of the technical user manual and source code are shown in figures 4.4 ï 4.9

4.4.1 User Manual

From the use case diagram of USPT figure 3.1, the activities of the user are: Upload Project, Compare Project and View Project. The systems can also ócompareô Projects with database information to track cases plagiarism.

4.4.1.1 Upload Project:

Figures 4.4 shows how project is uploaded into the system. Click on ónewô button in a menu bar or óenter new projectô icon. When you click óNewô, another user interface will appears, then, fill it before clicking submit. The text field has an asterisk (*), indicating that it is compulsory while the other asterisk attach to the text field (browse) change from color red to green if the content is successfully uploaded otherwise, the asterisk color remain same as default red.

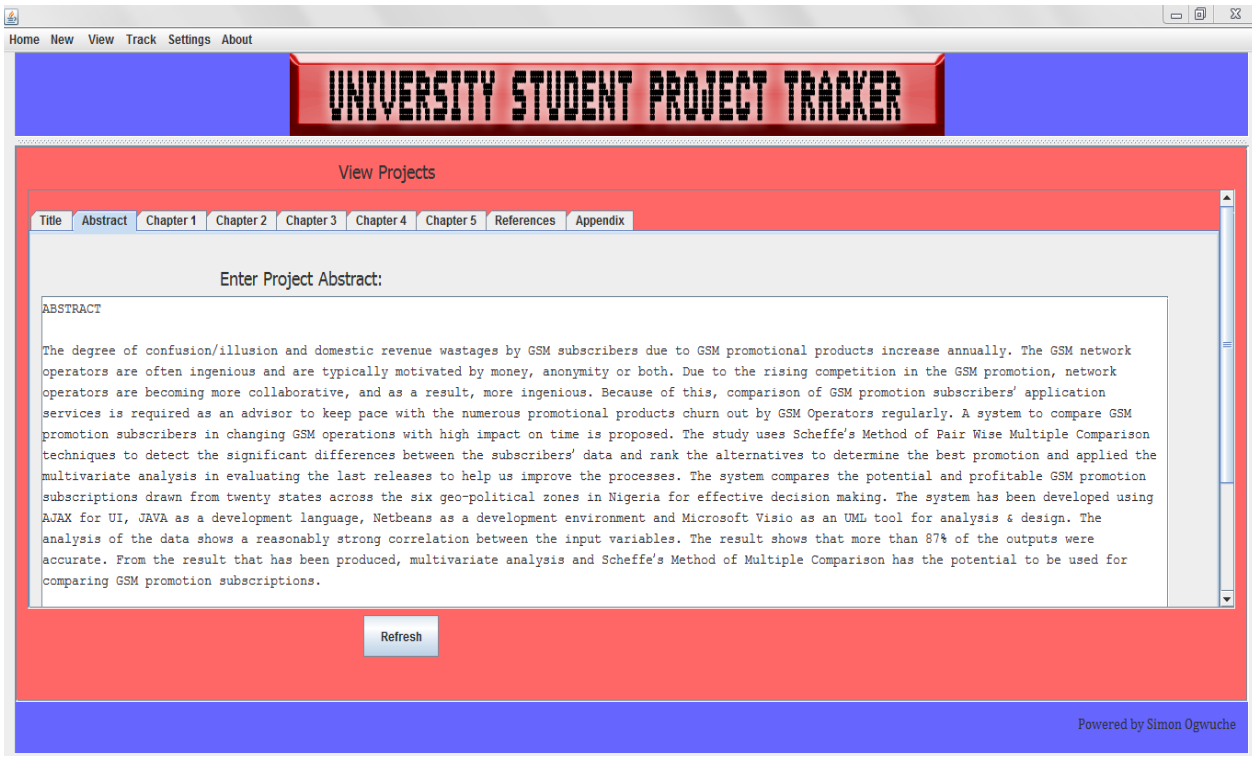

Home New View Track Settings About
UNIVERSITY STUDENT PROJECT TRACKER
Browse
Powered by Simon Ogwuche

Title:	COMPARISON OF GSM PROMOTION SUBSCRIBERS' APPLICATION S *
Student's Name:	AMUCHE, OCHIGBO BENEDICT *
Student's Reg. Number:	PG/MSC/11/59582 *
Supervisor's Name:	PROFESSOR BAKPO, F. S. *
Project type:	Masters
Abstract:	Browse *
Chapter 1:	Browse *
Chapter 2:	Browse *
Chapter 3:	Browse *
Chapter 4:	Browse *
Chapter 5:	Browse *
References:	Browse *
Appendix:	Browse

Figure 4.4: Uploading project into the system.

4.4.1.2 View Project:

To view, click on view from home page, and then click on view projects, a list of projects will appear, select the project to be viewed by clicking on it then right click on the project and select view full project from the pop-up menu and the view project opens, from where, title, abstract, chapter 1 to chapter 5, references and appendix of the project can be viewed. Figure 4.3 is an example of view project showing abstract of the project.


Home New View Track Settings About
UNIVERSITY STUDENT PROJECT TRACKER
View Projects
Title Abstract Chapter 1 Chapter 2 Chapter 3 Chapter 4 Chapter 5 References Appendix
Enter Project Abstract:
Refresh
Powered by Simon Ogwuche

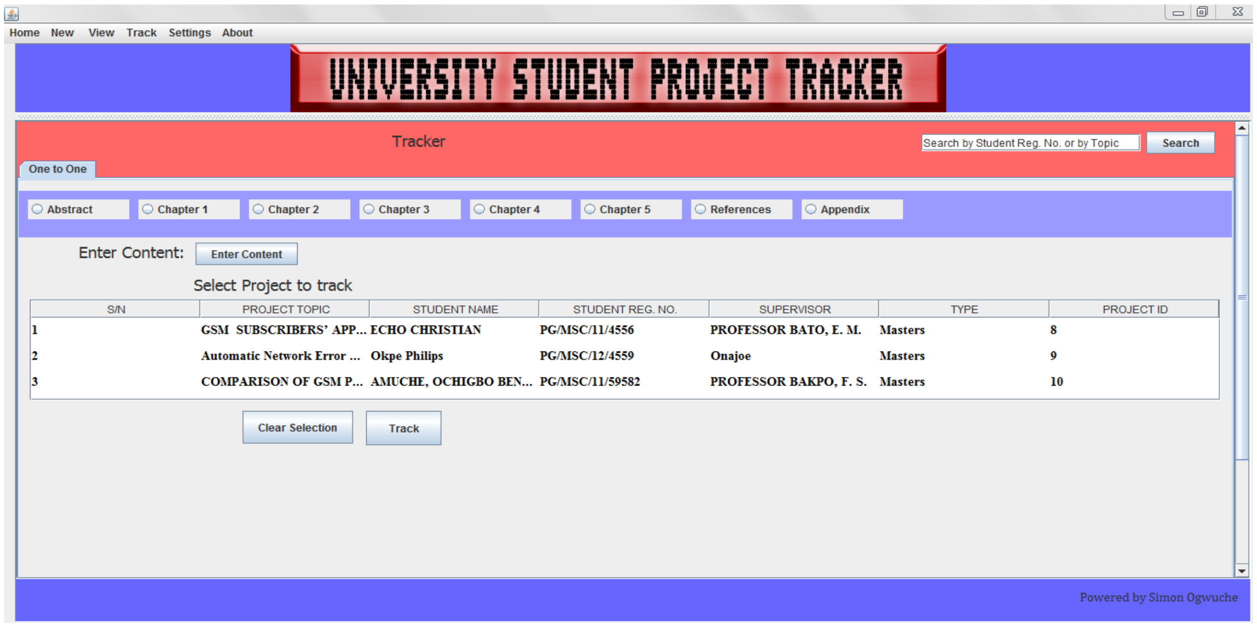
```
ABSTRACT

The degree of confusion/illusion and domestic revenue wastages by GSM subscribers due to GSM promotional products increase annually. The GSM network
operators are often ingenious and are typically motivated by money, anonymity or both. Due to the rising competition in the GSM promotion, network
operators are becoming more collaborative, and as a result, more ingenious. Because of this, comparison of GSM promotion subscribers' application
services is required as an advisor to keep pace with the numerous promotional products churn out by GSM Operators regularly. A system to compare GSM
promotion subscribers in changing GSM operations with high impact on time is proposed. The study uses Scheffe's Method of Pair Wise Multiple Comparison
techniques to detect the significant differences between the subscribers' data and rank the alternatives to determine the best promotion and applied the
multivariate analysis in evaluating the last releases to help us improve the processes. The system compares the potential and profitable GSM promotion
subscriptions drawn from twenty states across the six geo-political zones in Nigeria for effective decision making. The system has been developed using
AJAX for UI, JAVA as a development language, Netbeans as a development environment and Microsoft Visio as an UML tool for analysis & design. The
analysis of the data shows a reasonably strong correlation between the input variables. The result shows that more than 87% of the outputs were
accurate. From the result that has been produced, multivariate analysis and Scheffe's Method of Multiple Comparison has the potential to be used for
comparing GSM promotion subscriptions.
```

Figure 4.5: Screenshot of a project showing the abstract

4.4.1.3 Compare Project for plagiarize text:

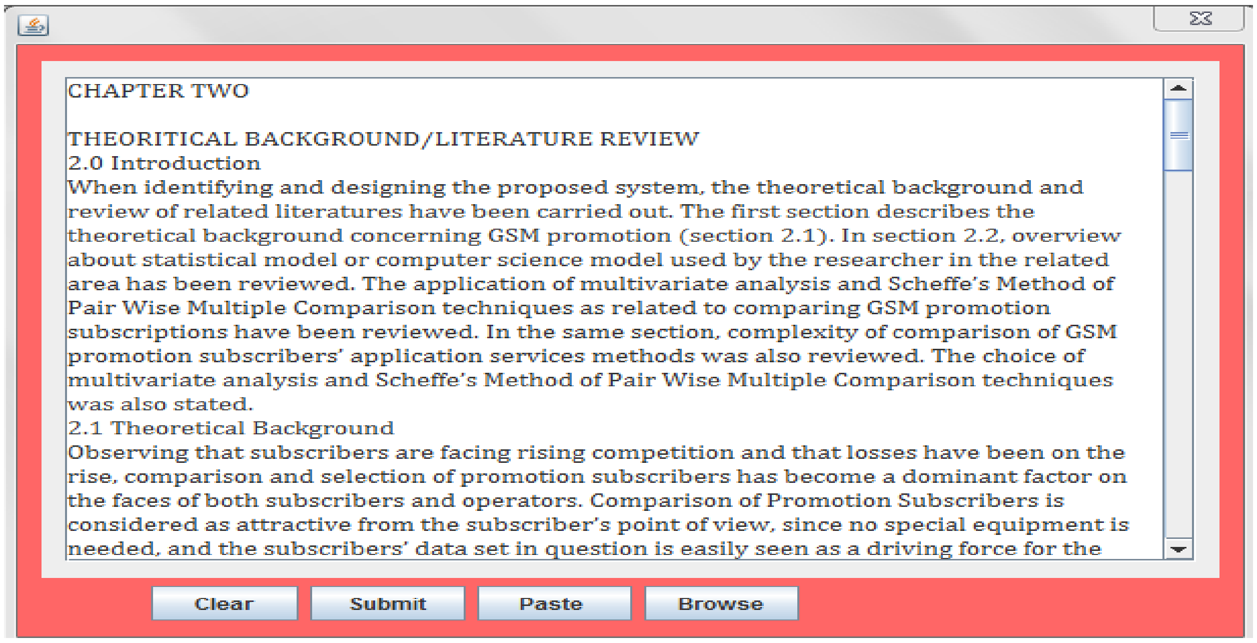
Figures 4.6 to 4.8 showed how text document can be compared to match plagiarized work. Figure 4.6 is the first interface for the tracker, to do this, click on track then click on tracker to open the interface.



S/N	PROJECT TOPIC	STUDENT NAME	STUDENT REG. NO.	SUPERVISOR	TYPE	PROJECT ID
1	GSM SUBSCRIBERS' APP...	ECHO CHRISTIAN	PG/MSC/11/4556	PROFESSOR BATO, E. M.	Masters	8
2	Automatic Network Error ...	Okpe Philips	PG/MSC/12/4559	Onajoe	Masters	9
3	COMPARISON OF GSM P...	AMUCHE, OCHIGBO BEN...	PG/MSC/11/59582	PROFESSOR BAKPO, F. S.	Masters	10

Figure 4.6: Tracker Interface.

After the operation in figure 4.6 above, click on ćenter contentô for figure 4.7 to appear. Copy and paste the content of the proposed project in the text area or browse for files click to submit.

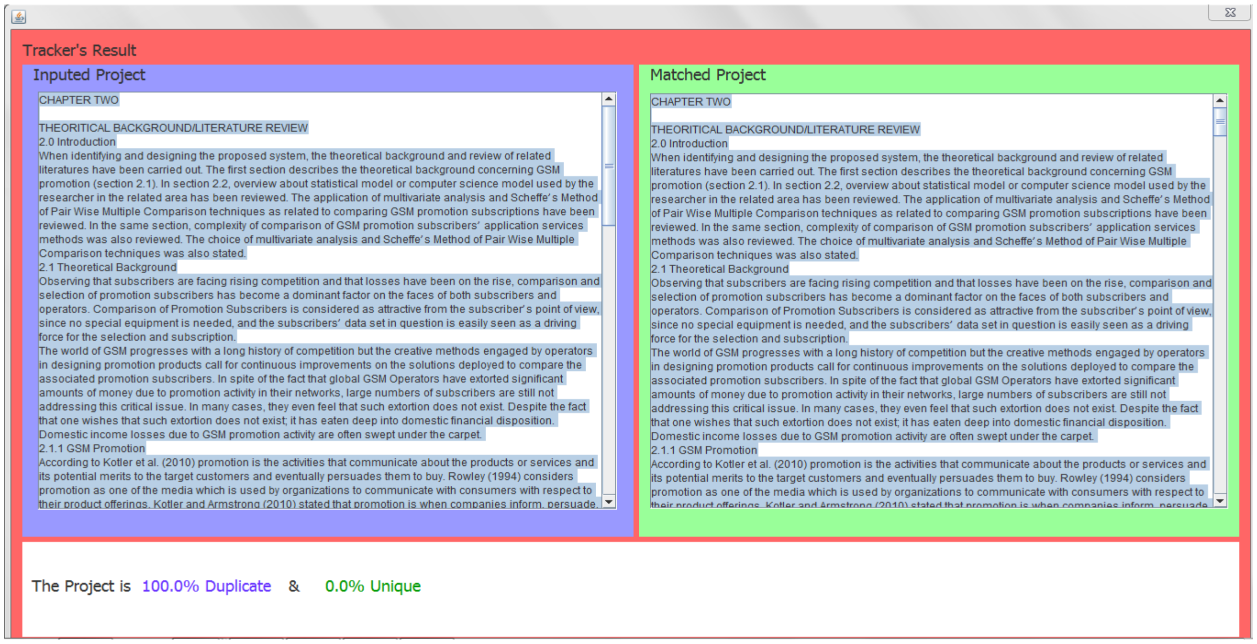


CHAPTER TWO

THEORITICAL BACKGROUND/LITERATURE REVIEW
2.0 Introduction
When identifying and designing the proposed system, the theoretical background and review of related literatures have been carried out. The first section describes the theoretical background concerning GSM promotion (section 2.1). In section 2.2, overview about statistical model or computer science model used by the researcher in the related area has been reviewed. The application of multivariate analysis and Scheffe's Method of Pair Wise Multiple Comparison techniques as related to comparing GSM promotion subscriptions have been reviewed. In the same section, complexity of comparison of GSM promotion subscribers' application services methods was also reviewed. The choice of multivariate analysis and Scheffe's Method of Pair Wise Multiple Comparison techniques was also stated.
2.1 Theoretical Background
Observing that subscribers are facing rising competition and that losses have been on the rise, comparison and selection of promotion subscribers has become a dominant factor on the faces of both subscribers and operators. Comparison of Promotion Subscribers is considered as attractive from the subscriber's point of view, since no special equipment is needed, and the subscribers' data set in question is easily seen as a driving force for the

Figure 4.7: Interface for entering content for the proposed project

After the operation in figure 4.7 proceed to select a radio button of a desired item (abstract, chapters, reference and appendix) then select the project to be tracked by clicking on track. When the tracking is completed the system displays the result in percentage (%) of duplicated and percentage (%) of uniqueness of the project. the left hand side is the proposed project while the right side is the project stored in the system. Figure 4.8 is the result of comparison from two projects (proposed and existing one stored in the system).



Tracker's Result

Inputed Project

CHAPTER TWO

THEORITICAL BACKGROUND/LITERATURE REVIEW
2.0 Introduction
When identifying and designing the proposed system, the theoretical background and review of related literatures have been carried out. The first section describes the theoretical background concerning GSM promotion (section 2.1). In section 2.2, overview about statistical model or computer science model used by the researcher in the related area has been reviewed. The application of multivariate analysis and Scheffe's Method of Pair Wise Multiple Comparison techniques as related to comparing GSM promotion subscriptions have been reviewed. In the same section, complexity of comparison of GSM promotion subscribers' application services methods was also reviewed. The choice of multivariate analysis and Scheffe's Method of Pair Wise Multiple Comparison techniques was also stated.
2.1 Theoretical Background
Observing that subscribers are facing rising competition and that losses have been on the rise, comparison and selection of promotion subscribers has become a dominant factor on the faces of both subscribers and operators. Comparison of Promotion Subscribers is considered as attractive from the subscriber's point of view, since no special equipment is needed, and the subscribers' data set in question is easily seen as a driving force for the selection and subscription.
The world of GSM progresses with a long history of competition but the creative methods engaged by operators in designing promotion products call for continuous improvements on the solutions deployed to compare the associated promotion subscribers. In spite of the fact that global GSM Operators have extorted significant amounts of money due to promotion activity in their networks, large numbers of subscribers are still not addressing this critical issue. In many cases, they even feel that such extortion does not exist. Despite the fact that one wishes that such extortion does not exist; it has eaten deep into domestic financial disposition. Domestic income losses due to GSM promotion activity are often swept under the carpet.
2.1.1 GSM Promotion
According to Kotler et al. (2010) promotion is the activities that communicate about the products or services and its potential merits to the target customers and eventually persuades them to buy. Rowley (1994) considers promotion as one of the media which is used by organizations to communicate with consumers with respect to their product offerings. Kotler and Armstrong (2010) stated that promotion is when companies inform, persuade

Matched Project

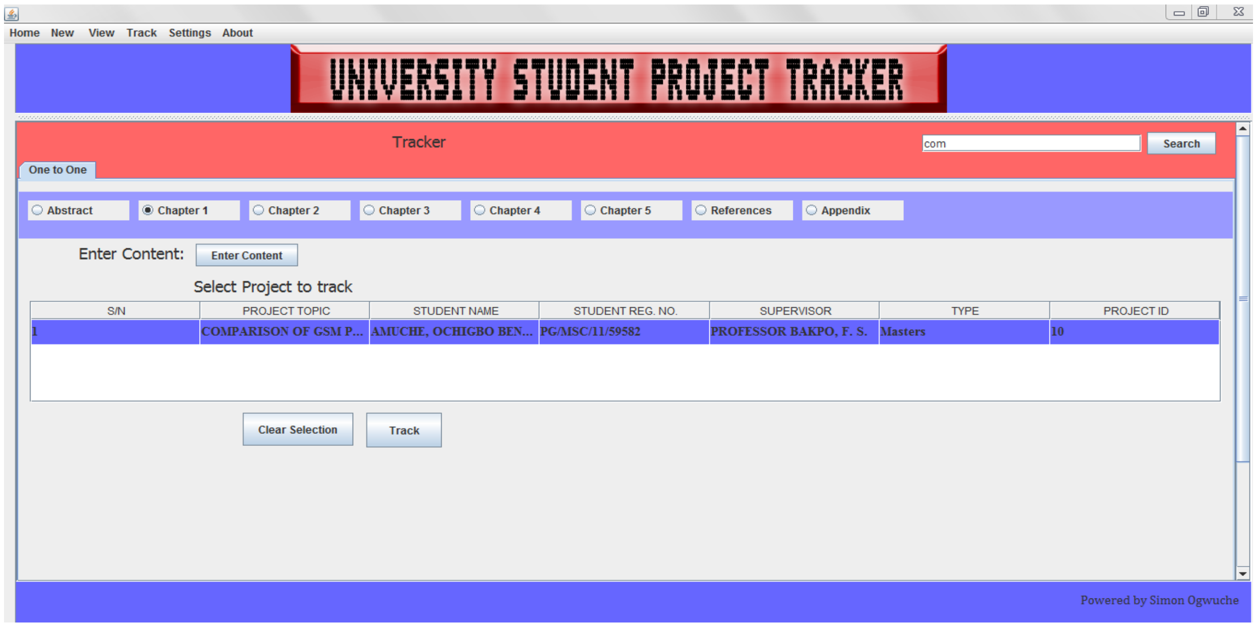
CHAPTER TWO

THEORITICAL BACKGROUND/LITERATURE REVIEW
2.0 Introduction
When identifying and designing the proposed system, the theoretical background and review of related literatures have been carried out. The first section describes the theoretical background concerning GSM promotion (section 2.1). In section 2.2, overview about statistical model or computer science model used by the researcher in the related area has been reviewed. The application of multivariate analysis and Scheffe's Method of Pair Wise Multiple Comparison techniques as related to comparing GSM promotion subscriptions have been reviewed. In the same section, complexity of comparison of GSM promotion subscribers' application services methods was also reviewed. The choice of multivariate analysis and Scheffe's Method of Pair Wise Multiple Comparison techniques was also stated.
2.1 Theoretical Background
Observing that subscribers are facing rising competition and that losses have been on the rise, comparison and selection of promotion subscribers has become a dominant factor on the faces of both subscribers and operators. Comparison of Promotion Subscribers is considered as attractive from the subscriber's point of view, since no special equipment is needed, and the subscribers' data set in question is easily seen as a driving force for the selection and subscription.
The world of GSM progresses with a long history of competition but the creative methods engaged by operators in designing promotion products call for continuous improvements on the solutions deployed to compare the associated promotion subscribers. In spite of the fact that global GSM Operators have extorted significant amounts of money due to promotion activity in their networks, large numbers of subscribers are still not addressing this critical issue. In many cases, they even feel that such extortion does not exist. Despite the fact that one wishes that such extortion does not exist; it has eaten deep into domestic financial disposition. Domestic income losses due to GSM promotion activity are often swept under the carpet.
2.1.1 GSM Promotion
According to Kotler et al. (2010) promotion is the activities that communicate about the products or services and its potential merits to the target customers and eventually persuades them to buy. Rowley (1994) considers promotion as one of the media which is used by organizations to communicate with consumers with respect to their product offerings. Kotler and Armstrong (2010) stated that promotion is when companies inform, persuade

The Project is 100.0% Duplicate & 0.0% Unique

Figure 4.8: Result of comparison

Figure 4.9 showed an easy way to search for project to be tracked. When there is over a thousand projects in the database and a particular project is desired to be compared; type in the studentôs registration number or title name the project will be displayed. Figure 4.9 is an example when ócomô was typed in the search box.



S/N	PROJECT TOPIC	STUDENT NAME	STUDENT REG. NO.	SUPERVISOR	TYPE	PROJECT ID
1	COMPARISON OF GSM P...	AMUCHE, OCHIGBO BEN...	PG/MSC/11/59582	PROFESSOR BAKPO, F. S.	Masters	10

Figure 4.9: Result of searching one project from multiple projects.

See Appendix D for the detailed list of figures.

4.4.2 Source code listing

See Appendix A for the detailed Source Codes Listing.

Chapter 5: Summary and Conclusion

5.0 Summary

This project was aimed at solving the problems of rampant copying of someone elseĝs project as their own. USPT was developed using sequential search algorithm with 3-grams, the USPT was implemented as software with the aid of Java programming language. The USPT was evaluated and found capable of handling ten thousand words length with high efficiency in University Student Project Tracker (USPT), store projects in soft copies in the database and the software can detect plagiarized text with the corresponding percentage text copied from the original text. Database aspect of the USPT was implemented using MYSQL Database Management System (DBMS) which act as the repository of the projects stored in the USPT. The application is an off-line tracking system; it will be installed on the user computer for use.

Studentsô projects tracker was successfully developed to check the rampant cases of plagiarism by students in the university using Java and MYSQL database management system (DBMS) which is capable of handling ten thousand words length and was evaluated to have high efficiency in University Student Project Tracker (USPT), store projects in soft copy in the database, fast in operation, easy application and avoidable.

5.1 Conclusion

The project embarked upon is the development of University Student project tracking system. The project was successfully designed, developed, implemented and tested to have fulfilled the stated objectives. It will help the user to track cases of plagiarism which is rampart among students. It is aimed at providing an easy to use tool to especially supervisors to enable them ascertain the originality or otherwise of a proposed project before granting approval.

5.2 Recommendations

This study recommends that:

1. The work should be put to use to replace the manual ways of storing studentsô projects.
2. Project content should not have more than ten thousand (10, 000) words.
3. Only one project should be compared at a time.

5.3 Suggested areas for further works

The following areas are suggested for further study.

1. Further works should be experimented based on one-to-many matching.
2. Comparison should go beyond words to involve structures as well as symbols.
3. Provision should be made for online Google to facilitate comparison of proposed project.

Reference

[1] Daniel Y. L. (2000): Introduction to Java Programming, Third Edition, Prentice-Hall, Inc. Upper Saddle River, New Jersey, USA. Pp.5

[2] Sun Microsystems, ñJava Programming Languageò, Student Guide, Sun Microsystems, Inc. 2008. Pp A6

[3] Davies, P.B (2004): Database Systems, Third Edition, Palgrave Macmillan, UK. Pp. 33

[4] Craig A., 2004. Notes on Plagiarism. Available online at http://www.lib.usm.edu/legacy/plag/whatisplag.php

[5] Tom Pace.2002, The Effect of Plagiarism _ First Year Composition. Available online at http://sites.jcu.edu/fycomp/pages/plagiarism/the-effects-of-plagiarism/

[6] *Mary D.*, Demand M., The Effect of Plagiarism on Students. Hearst Seattle Media, LLC. 2014. Available online at http://education.seattlepi.com/effect-plagiarism-students-1195.html

[7] Heintze, Nevin. "Scalable document fingerprinting." 1996 USENIX workshop on electronic commerce. Vol. 3. No. 1. 1996.

[8] T. Hoad, and J. Zobel, ñMethods for identifying versioned and plagiarised documentsò, Journal of the American Society of Information Science and Technology. Vol 54, num 3, pp 203ï 215, 2003.

[9] Lancaster, Thomas, and Fintan Culwin. "Classifications of plagiarism detection engines." Innovation in Teaching and Learning in Information and Computer Sciences ITALICS Vol. 4(2) (2005).

[10] Arwin, Christian, and Seyed MM Tahaghoghi. "Plagiarism detection across programming languages." Proceedings of the 29th Australasian Computer Science Conference-Volume 48, 2006. Australian Computer Society, Inc., (pp 277-286).

[11] Sven Meyer Zu Eissen and Stein. B 2006. Intrinsic Plagiarism Detection. LNCS 3936, pp. 565-569, Springer-Verlag.

[12] Zini M., Fabbri M., Moneglia M., Panunzi A., 2006. Plagiarism Detection through Multilevel Text Comparison. Proceedings of the Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS'06), pp

[13] Meyer zu Eissen, S., B. Stein, and M. Kulig. 2007. Plagiarism Detection without Reference Collections. Decker and Lenz (Eds.): Advances in Data Analysis Selected Papers from the 30th Annual Conference of the German Classification Society (GfKl) Berlin, ISBN 978-3-540-70980-0, pp. 359-366, c Springer 2007.

[14] Lukashenko R., Graudina V., and Grundspenkis J. 2007 Computer Based Plagiarism Detection Methods and Tools. International Conference on Computer Systems and Technologies.

[15] Barron-Cede, A., and Rosso, P., 2009 on Automatic Plagiarism Detection based on n-Grams comparison. M. Boughanem et al. (Eds.): European Conference on Information Retrieval (ECIR 2009), Lecture Note in Computer Science (LNCS 5478), pp. 696ï 700, c_Springer-Verlag Berlin Heidelberg 2009

[16] Stamatatos, E., 2009 Intrinsic Plagiarism Detection Using Character n-gram Profiles. Stein, Rosso, Stamatatos, Koppel, Agirre (Eds.): Pan ï PC ï 09 Plagiarism Corpus (PAN'09), pp. 38-46, 2009.

[17] Alvi, F., Stevenson, M., Clough, P.D.: Hashing and Merging Heuristics for Text Reuse Detection. In: Working Notes for PAN at CLEF-2014 Conference, Sheffield, UK, 2014. Available online at http://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-AlviEt2014.pdf

[18] Alvi, F., Stevenson, M., Clough, P.D.: The Short Stories Corpus. In: Notebook for PAN at CLEF 2015 Conference, Sheffield, UK, 2014. Available online at http://ceur-ws.org/Vol-1391/90-CR.pdf

Appendices

Appendix A: Code Listing

```
package projecttracker;



import java.util.ArrayList;

import javax.swing.JOptionPane;

import javax.swing.text.BadLocationException;

import javax.swing.text.DefaultHighlighter;

import javax.swing.text.Highlighter;

import javax.swing.text.JTextComponent;

/**

 * @author Simon

 */

public class Result extends javax.swing.JPanel {

    //variables

     ArrayList holdString = new ArrayList();

    ArrayList holdChar = new ArrayList();

    ArrayList holdS = new ArrayList();//array to hold index of selected.

    ArrayList holdS1 = new ArrayList();//array to hold index of selected1.

    ArrayList holdEnd= new ArrayList();//array to hold end index of selected
```

```
/** Creates new form Result */

    public Result() {

        initComponents();

        format();

    }

    // <editor-fold defaultstate="collapsed" desc="Generated Code">

    private void initComponents() {

        jLabel11 = new javax.swing.JLabel();

        Result = new javax.swing.JDesktopPane();

        unq1 = new javax.swing.JLabel();

        result2 = new javax.swing.JLabel();

        dup = new javax.swing.JLabel();

        result4 = new javax.swing.JLabel();

        jPanel1 = new javax.swing.JPanel();

        jLabel12 = new javax.swing.JLabel();

        jScrollPane1 = new javax.swing.JScrollPane();

        Area1 = new javax.swing.JTextArea();

        jPanel2 = new javax.swing.JPanel();

        jLabel13 = new javax.swing.JLabel();

        jScrollPane2 = new javax.swing.JScrollPane();

        Area2 = new javax.swing.JTextArea();
```

```
setBackground(new java.awt.Color(255, 102, 102));

jLabel11.setFont(new java.awt.Font("Tahoma", 0, 18));

jLabel11.setText("Tracker's Result");

Result.setBackground(new java.awt.Color(255, 255, 255));

unq1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

unq1.setForeground(new java.awt.Color(0, 153, 0));

unq1.setText("100% unique");

unq1.setBounds(330, 10, 140, 80);

Result.add(unq1, javax.swing.JLayeredPane.DEFAULT_LAYER);

result2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

result2.setText("The Project is ");

result2.setBounds(10, 10, 114, 80);

Result.add(result2, javax.swing.JLayeredPane.DEFAULT_LAYER);

dup.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

dup.setForeground(new java.awt.Color(102, 51, 255));

dup.setText("100% duplicate");

dup.setBounds(130, 10, 160, 80);

Result.add(dup, javax.swing.JLayeredPane.DEFAULT_LAYER);

result4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

result4.setText("&");

result4.setBounds(290, 10, 20, 80);
```

```
        Result.add(result4, javax.swing.JLayeredPane.DEFAULT_LAYER);

        jPanel1.setBackground(new java.awt.Color(153, 153, 255));

        jLabel12.setFont(new java.awt.Font("Tahoma", 0, 18));

        jLabel12.setText("Inputed Project");

        Area1.setColumns(20);

        Area1.setRows(5);

        jScrollPane1.setViewportView(Area1);

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

        jPanel1.setLayout(jPanel1Layout);

        jPanel1Layout.setHorizontalGroup(

            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel1Layout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addContainerGap(205, Short.MAX_VALUE))


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(jPanel1Layout.createSequentialGroup()

                    .addGap(17, 17, 17)

                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 340,
Short.MAX_VALUE)
```

```
                .addGap(18, 18, 18)))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabel12)
                .addContainerGap(311, Short.MAX_VALUE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(31, 31, 31)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 271,
Short.MAX_VALUE)
                    .addGap(31, 31, 31)))
        );


        jPanel2.setBackground(new java.awt.Color(153, 255, 153));
        jLabel13.setFont(new java.awt.Font("Tahoma", 0, 18));
        jLabel13.setText("Matched Project");
        Area2.setColumns(20);
        Area2.setRows(5);
        jScrollPane2.setViewportView(Area2);
        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(190, Short.MAX_VALUE))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 340,
Short.MAX_VALUE)
                    .addContainerGap()))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel13)
            .addContainerGap(311, Short.MAX_VALUE))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(33, 33, 33)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 268,
Short.MAX_VALUE)
                .addGap(32, 32, 32)))
      );

      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
      this.setLayout(layout);
      layout.setHorizontalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(Result, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 741, Short.MAX_VALUE)
                .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
```

```
                        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addComponent(jLabel11, javax.swing.GroupLayout.Alignment.LEADING))
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel11)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jPanel2, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(Result, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addGap(33, 33, 33))
        );
    }// </editor-fold>

 // Variables declaration - do not modify
    private javax.swing.JTextArea Area1;
    private javax.swing.JTextArea Area2;
    private javax.swing.JDesktopPane Result;
    private javax.swing.JLabel dup;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JLabel result2;
    private javax.swing.JLabel result4;
    private javax.swing.JLabel unq1;
    // End of variables declaration
//method to add text to interface
```

```
public void add1(String A1,String A2/*, String R1*/){

    Area1.setText(A1);

    Area2.setText(A2);

   //result1.setText(R1);

}//end method

//method to format component

    public void format(){

    Area1.setLineWrap(true);

    Area1.setWrapStyleWord(true);

    Area2.setLineWrap(true);

    Area2.setWrapStyleWord(true);

    Area1.setEditable(false);

    Area2.setEditable(false);

}//end method

//method to track

public void track(){

 //empty global vars if filled

    holdString.clear();

    holdS1.clear();

    holdS.clear();

    holdEnd.clear();
    //first collect the two strings
        String col1 = collectString(Area1);
        String col2 = collectString(Area2);
    //check if they are equal
        if (col1.compareToIgnoreCase(col2)==0){
            int pDup = 100;
            int pUnique = 0;
               //Setting Status Report
               dup.setText(""+pDup+"% Duplicate");
               unq1.setText(""+pUnique+"% Unique");
               //String resultReport = "The Project is "+pDup+"% Duplicate  and \n"+pUnique+"%
Unique ";
               //unq.setText(resultReport);
        }
      else{
        String word1 = col1.replaceAll("\n", " ");
        String word2 = col2.replaceAll("\n", " ");//makesure that newline does not exit
    //next arrange the first string in a 3 by 3 arrangement
        ArrayList arr = arrange1(word1,3);//do a 3 by 3 arrangement of col1 string


 //3. match the selected string to string B to get the selected strings
         for (int i=0;i<arr.size();i++){
          matchText(Area2,word2,arr.get(i).toString());
       }//end loop
//4. get selected words for calculation
    /////////////
     String t1 = "";//to hold mix string
    String t2 = "";//to hold matched String
    String t3 = "";//to hold rearranged String
    ///////////////
    String [] array1 = new String[holdS.size()];
    int [] array2 = new int[holdS.size()];//start index
    int [] array3 = new int[holdEnd.size()];//end index
    for (int y=0;y<holdS1.size();y++){
        array1[y]=holdS.get(y).toString();
        array2[y]=Integer.parseInt(holdS1.get(y).toString());//fill start index
        array3[y]=Integer.parseInt(holdEnd.get(y).toString());//fill end index
    }
    String [] arr1=rearrange(array1,array2,array3);
    String[] arr2 = check(arr1,array2,array3);
    String tt2 ="";
    for (int i=0;i<arr2.length;i++)
        tt2+=arr2[i];
        //JOptionPane.showMessageDialog(null, "The Selected List now \n"+tt2);
    //5. now perform calculation of uniqueness and duplicate
     double total = getLength(Area1.getText());
        double X = 0;
        if(tt2.length()>0){
        String [] collect1 = tt2.split(" ");//to collect matched string
        //String report1 = "X = "+collect1.length+"\nY(total)= "+total;
        ///JOptionPane.showMessageDialog(null,tt2);//to print selected words
         X = collect1.length;
        }
        if (X>total)X=total;
     double percentDup = result(X,total);
     ///JOptionPane.showMessageDialog(null,"matched words ="+X+"\nTotal = "+total);//to print
no. of matched words and total
     //uniqueness
     double percentUnique = (100 - percentDup);
     double pDup = Math.round(percentDup);
     double pUnique = Math.round(percentUnique);
     //Setting Status Report
               dup.setText(""+pDup+"% Duplicate");
               unq1.setText(""+pUnique+"% Unique");//String resultReport = "The Project is
"+pDup+"% Duplicate  and \n"+pUnique+"% Unique ";
```

```
     //unq1.setText(resultReport);

      //JOptionPane.showMessageDialog(null, "The Project is "+percentDup+"% Duplicate
approximately "+pDup+" and \n"+percentUnique+"% Unique approximately "+pUnique);

  //6. match first text area

      for (int i=0;i<holdS.size();i++){

              justMatch(Area1,word1,holdS.get(i).toString());

          }//end loop

        }//end 1st condition

    }

//method to collect strings from a text component n return just words.

 public String collectString(JTextComponent comp){

     String words = comp.getText();

      String ct = "";//to hold final words

        String words2 = words.trim();

       //String words1=words2.toUpperCase();

     //trim then convert any enter key to space before applying the split

     String words3 = words2.replaceAll("\n*\t", " ");

     //String words3a = words3.replaceAll("\t","");

     String[] c = words3.split(" ");

     //then remove any empty string array

     for (int i=0; i < c.length; i++){

         String chk = c[i];
        //System.out.println(chk);

        if (!(c[i].equalsIgnoreCase("
"))&&!(c[i].equalsIgnoreCase("\n"))&&!(c[i].equalsIgnoreCase("\t"))){

          //if ((chk.charAt(0)!='\t')&&(chk.charAt(0)!='\n')&&(chk.charAt(0)!=' ')){//to remove
any tab or any white space btw

           ct = ct+c[i]+" ";

        }//end if

    }

    return ct;

 }//end method to collect string from a component

  //method to arrange string in a n by n array

 public ArrayList arrange1(String words, int n){

    String[] hold = words.split(" ");

    String hold2 = "";

    int counter =0;

    int firstcount=0;

    int i = 0;

    ArrayList wd = new ArrayList();

    while (firstcount <n){

    for ( i=firstcount; i<hold.length;i++){

        hold2 += hold[i]+" ";

        counter++;
        if (counter==n){
            wd.add(hold2);
            hold2 = "";
            counter = 0;
           // if ((hold.length-i)==(n+1))
            if(i==hold.length-n){
                   i=hold.length;//end loop
                   //hold2="";
            }
        }//end if
    }//end for loop
    firstcount++;
    hold2="";
    counter=0;
    }//end while loop
    return wd;
}//end method arrange1
//method to match string
public void matchText(JTextComponent comp, String compText, String matchText){
     String collect="";
    int z=0;//to locate where the first letter is in the string
int counter=0;//to count the number of matches

String singlecha = matchText.toUpperCase();//to be used to verify the first letter

Highlighter h = comp.getHighlighter();//to highlight

 // h.removeAllHighlights();//remove all previous highligts

  //String text = comp.getText().toUpperCase();//getText from component.

  String text = compText.toUpperCase();//getText from component.

  for (int j=0; j<text.length(); j+=1){

      z=0;

      char ch = text.charAt(j);//get first character in the component

      if(ch==singlecha.charAt(0)){//if the first char of the word found

          //1st ensure that the subsequent ones are there also

          z=j;

          for (int k = 0;k<matchText.length();k++){

            //collect holds the temp string from comptext to match

              if(z<text.length()){

                  collect+=compText.charAt(z);

                  z++;

              } else k=matchText.length();

          }//JOptionPane.showMessageDialog(null, "working well "+z);

          if (collect.equalsIgnoreCase(matchText)){//check for match

          try {//match found so go ahead to highlight
```

```
                h.addHighlight(j, j+matchText.length(), DefaultHighlighter.DefaultPainter);

                //send text to global array

                String[] temp = matchText.split(" ");//

                holdS.add(matchText);

                holdS1.add(j);//to hold index

                holdEnd.add(j+matchText.length());//to hold end index of selected text

                for (int k=0;k<temp.length;k++) {

                    holdString.add(temp[k]);

                }

                //temp = null;

            }catch(BadLocationException ble)

            {

            }

            counter ++;

            }//end if collect comparison

         collect="";}

      }

      //JOptionPane.showMessageDialog(null, "Number of match is "+counter+"\nTotal number
of words is "+countwords(compText));

 }//end method

 //method to just match

 public void justMatch(JTextComponent comp, String compText, String matchText){
```

```
 String collect="";
int z=0;//to locate where the first letter is in the string
int counter=0;//to count the number of matches
String singlecha = matchText.toUpperCase();//to be used to verify the first letter
Highlighter h = comp.getHighlighter();//to highlight
  // h.removeAllHighlights();//remove all previous highligts
   String text = comp.getText().toUpperCase();//getText from component.
   for (int j=0; j<text.length(); j+=1){
       z=0;
       char ch = text.charAt(j);//get first character in the component
       if(ch==singlecha.charAt(0)){//if the first char of the word found
           //1st ensure that the subsequent ones are there also
           z=j;
           for (int k = 0;k<matchText.length();k++){
             //collect holds the temp string from comptext to match
               if(z<text.length()){
                   collect+=compText.charAt(z);
                   z++;
               }
           }//JOptionPane.showMessageDialog(null, "working well "+z);
           if (collect.equalsIgnoreCase(matchText)){//check for match
```

```
            try {//match found so go ahead to highlight

               h.addHighlight(j, j+matchText.length(), DefaultHighlighter.DefaultPainter);

               }catch(BadLocationException ble){}

            counter ++;

            }//end if collect comparison

         collect="";}

      }

 }//end method

 //method to rearrange contents in the array



public String[] rearrange(String[] arr1, int[] index,int[] endIndex){

   int temp;

   String temp2;

   int temp3;//for selected end index

      for(int i=0; i<index.length-1;i++){//bubble sort

         for(int j=1; j<index.length-i;j++){

            if(index[j-1]>index[j]){

               temp=index[j-1];//hold index to change

               temp2=arr1[j-1];//hold string to change

               temp3=endIndex[j-1];//hold end index to change
```

```
                index[j-1]=index[j];//illeterate index
                arr1[j-1]=arr1[j];//illeterate String
                endIndex[j-1]=endIndex[j];//illeterate end index
                index[j] = temp;//change index
                arr1[j] = temp2;//change string
                endIndex[j] = temp3;//change end Index
            }//end if
        }//end inner for loop
    }//end for loop
    return arr1;
}//end method
//method to check selected text
public String[] check(String[] arr, int[] index, int[] endIndex){
    String temp="";
    String []arr2 = new String [arr.length];
    for (int i=0;i<arr.length;i++){
        if(i==arr.length-1){
            arr2[i]=arr[i];
        }else
        if (endIndex[i]>index[i+1]){
            arr2[i]=arr[i].substring(0, (index[i+1]-index[i]));
        }//end if
        else arr2[i]=arr[i];
    }//end for loop
    return arr2;
}//end method to check
 //method to get length
 int getLength(String words){
     String word1 = words.replaceAll("\n", " ");
     String [] arr = word1.split(" ");
     return arr.length;
 }//end method
 //method to implement formula
 double result(double x, double y){
     double c = ((x/y)*100);
     return c;
 }//end method
}//end class
```