**FUNCTIONAL SPECIFICATIONS TEMPLATE**



**FUNCTIONAL SPECIFICATIONS:**

**NIGHTCOREMECH**

**DEVELOPMENT TEAM**

**12/09/2022**

**Version 1.0.0**

| VERSION HISTORY | | | | |
| --- | --- | --- | --- | --- |
| VERSION | APPROVED BY | REVISION DATE | DESCRIPTION OF CHANGE | AUTHOR |
| 1 |  | O8/09/2022 | First Draft | Development Team |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Functional Specifications Document**

**Authorization Memorandum**

I have carefully assessed the Functional Specifications Document for the SimiLabs Plagiarism/Stylometry Checker.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

\_\_\_\_\_\_ The document is accepted.

\_\_\_\_\_\_ The document is accepted pending the changes noted.

\_\_\_\_\_\_ The document is not accepted.

We fully accept the changes as needed improvements and authorize initiation of work to proceed. Based on our authority and judgment, the continued operation of this system is authorized.

Ricus Warmenhoven 2022/09/12

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME DATE

Project Manager

NWU Registrar 2022/09/12

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME DATE

Director

Hanno Visagie – Project Leader

Hano Strydom – Frontend Developer/Prototyping Lead

Llewellyn Anthony – Development Lead

Michael Rosin – Backend Developer

Annika du Toit – SQL Developer/DB Admin

Shené Boshoff – Cloud Administrator

Table of Contents

[1 INTRODUCTION 4](#_Toc113896719)

[1.1 Background 4](#_Toc113896720)

[1.2 Purpose 4](#_Toc113896721)

[1.3 Main Objectives 4](#_Toc113896722)

[1.4 Interfaces to External Systems 5](#_Toc113896723)

[1.5 Points of Contact 5](#_Toc113896724)

[1.6 Reference Documents 5](#_Toc113896725)

[1.7 Abbreviations and Acronyms 5](#_Toc113896726)

[1.8 Document Conventions 5](#_Toc113896727)

[2 GENERAL DESCRIPTION 6](#_Toc113896728)

[2.1 Product Context 6](#_Toc113896729)

[2.2 User Classes and Characteristics 6](#_Toc113896730)

[2.3 Overview of Functional Requirements 6](#_Toc113896731)

[2.4 Overview of Data Requirements 6](#_Toc113896732)

[2.5 Operating Environment 7](#_Toc113896733)

[2.6 General Constraints, Assumptions, Dependencies, Guidelines 7](#_Toc113896734)

[2.7 Design and Implementation Constraints 7](#_Toc113896735)

[2.8 User Documentation 7](#_Toc113896736)

[3 REQUIREMENTS 8](#_Toc113896737)

[3.1 External Interface Requirements 8](#_Toc113896738)

[3.2 User Interfaces 8](#_Toc113896739)

[3.3 Hardware Interfaces 8](#_Toc113896740)

[3.4 Software Interfaces 8](#_Toc113896741)

[3.5 Communications Interfaces 8](#_Toc113896742)

[3.6 Functional Requirements 8](#_Toc113896743)

[3.6.1 Purpose 8](#_Toc113896744)

[3.6.2 Inputs 8](#_Toc113896745)

[3.6.3 Processing 9](#_Toc113896746)

[3.6.4 Outputs 9](#_Toc113896747)

[3.7 Performance Requirements 9](#_Toc113896748)

[3.8 Security 10](#_Toc113896749)

[3.9 Usability 10](#_Toc113896750)

[3.10 Data Flow 10](#_Toc113896751)

[3.11 Other Requirements 10](#_Toc113896752)

[Appendix A. Analysis Models 11](#_Toc113896753)

[Appendix B. Issues List 12](#_Toc113896754)

# INTRODUCTION

## Background

The previous instance of the SimiLabs software was faulty and did not adhere to the NWU Registrar’s requirements. The software failed to produce results as to assist lecturers and the Registrar in identifying contract-cheating among students. The statistics provided by the stylometry functionality failed to implicate any students in contract-cheating, even if it was without a doubt the case. The text comparison functionalities also failed to highlight any texts that was similar between source and comparison documents. The system also lacked a functioning user login and registration form.

## Purpose

Several administrative duties are the responsibility of the Registrar at North-West University. Keeping track of university students' grades and a wide range of other supporting records and documentation are among these duties. In accordance with Gartner's definition of information governance, the university sees it as an all-encompassing framework that gives control over information and the procedures by which it is created, processed, and curated at the institution.

The current demands from the client (the NWU Registrar), with Mr. Zander Janse van Rensburg serving as the project managing manager, requires our company to design and construct a modular workflow system that would assist academic lecturers in identifying and reporting cases of academic misconduct in accordance with the NWU SOPS. To combat contract-cheating, the NWU Registrar must assess each instance separately and employ specialists to provide technical reports. If the technical reports do not self-evidently emphasize the severity of the plagiarism, external subject matter experts (SMEs) are asked to review the technical reports with an additional report that offers a deeper understanding of the suspected plagiarism. The technical need is to manually compare the allegedly plagiarized text in question with the original text used as evidence. The registrar also needs to identify authorship attribution, via the use of stylometry to generate reports. It is rather difficult to manually compare text, so the Registrar encouraged the NightcoreMech development team to develop a system that can automatically detect text comparisons and authorship attribution without the need to manually compare documents.

## Main Objectives

* The automation of text comparisons between the question and the supporting material, which must yield a measure of similarity between the two texts. The software should reduce the amount of time required to manually compare two texts and generalize the evaluation of how severe the conjectured copying is using a similarity measure. To produce more accurate reports, better explain academic misbehavior, and facilitate better decision-making, the program must integrate text-comparison capabilities with stylometric analytics.
* The software that is now available has features like a substandard stylometric tool and a primitive text comparison tool. Technical reports use the basic text comparison tool to identify similarities between texts, but the stylometric tool is underutilized despite having significant potential for helping the investigator in authorship attribution.
* The program must be independent, terrestrial-based, and capable of running on numerous platforms simultaneously.
* Data updates must occur every day.
* The system must be able to sustain ongoing database transactions (storing stylometry statistics and document metadata).
* The project management methodology must be followed by the development methodology. The MPMM/Method 123 approach or a Waterfall/Agile hybrid approach should be used.
* As a minimum requirement, the program should be able to identify textual similarities between two sources.
* The program should have stylometric features, including the ability to identify a document's original author.
* Resource overhead should be avoided, and the program should be memory efficient.
* The system should be able to provide extensive reports on the stylometric analysis and text-comparisons between a source document and comparison document, also via the use of corpora. These results should be able to assist the Registrar in the contract-cheating identification process
* The application must be web-based
* During the development phase, statistical analysis and expertise should be essential.
* The development team can build a local corpus of documents to compare and spot contract cheating instead of building a database to store data.

## Interfaces to External Systems

The system shouldn't need to interface with any outside systems at this time. On the other hand, the client's earlier system might be improved upon and given new functionality.

## Points of Contact

Contact the development team:

<https://github.com/ISE-Project-2022/SimiLabs_2022#readme>

For additional information, contact:

Zander Janse van Rensburg: [zander.jansevanrensburg@nwu.ac.za](mailto:zander.jansevanrensburg@nwu.ac.za)

Prof. Neels Kruger: [Neels.Kruger@nwu.ac.za](mailto:Neels.Kruger@nwu.ac.za)

## Reference Documents

The following documents is necessary to gain additional information on the project:

Tender Bid

Project Proposal and Plan

Business Case

Feasibility Study

<https://github.com/ISE-Project-2022/Documentation>

## Abbreviations and Acronyms

HTTPS: Hyper-Text Transfer Protocol Secure

SQL: Structured Query Language

COBIT: Control Objectives for Information and Related Technologies

NWU: North-West University

SOPS: Standard Operating Procedures

SMEs: Subject Matter Experts

MPMM: Method123 Project Management Methodology

## Document Conventions

The NightcoreMech development team use a very specific document convention. The document description is provided as the first part of the document title, and spaces are replaced by underscores. The document title ends with the company name

# GENERAL DESCRIPTION

## Product Context

To prevent contract cheating, the NWU Registrar must examine each case on an individual basis and request the technical opinions of specialists. If the technical reports do not emphasize the severity of the suspected plagiarism, external subject matter experts (SMEs) are tasked with evaluating the technical reports with an extra report that provides a more detailed explanation of the alleged plagiarism. The technical need is to manually match the allegedly copied information in question with the proof's original text. In order to create reports, the registrar must also assess authorship attribution using stylometry.

## User Classes and Characteristics

The product shall make use of following two types of class:

* Lecturer
* Registrar

Lecturers are the product's key users. The lecturers will analyse the texts for contract cheating or plagiarism and compile a report.

The registrar will also be using the product to view the evidence provided by the reports.

Both the lecturers and the registrar will have full administrative rights to use the program.

## Overview of Functional Requirements

* The users should have access to a straightforward and intuitive interface.
* Users should be able to complete tasks in a time efficient manner.
* Enhance the productivity of employees by providing functional software.
* Compile documentation/user manuals for users using a structure based on categories, which will serve as a guide to minimize misunderstanding.
* At the conclusion of the project’s lifetime, Nightcore Mech's engineers should be able to create a system that enables instructors at NWU to detect academic misconduct and contract cheating.
* The system should provide feedback reports highlighting text similarities between diverse sources in order to identify varying levels of student plagiarism.
* Application should be web-based.
* Automation of feedback reports with stylometric performance indicators displayed to identify contract cheating and original authorship
* Side-by-side comparison of the source document and the evidence texts, without having to manually compare text similarities
* The user should be able to select whether they want to compare 2 documents or compare against a corpus regarding the text comparisons.
* Feedback reports will be generated to assist the contract cheating identification process.
* Machine learning methods and stylometric analysis should be utilized to discover authorship and text similarities.
* A local corpus should be saved on a device for comparing documents and detecting contract cheating and authorship.
* Computing resources should be maintained effectively, and memory overhead should be minimized.
* The system must be capable of maintaining database transactions over time.

## Overview of Data Requirements

The product will make use of the following data throughout the various uses:

* User login credentials such as Username and Password
* Source Text and Alleged text
* Source Text and a Corpus of texts

## Operating Environment

Any Internet-connected device with a web browser.

## General Constraints, Assumptions, Dependencies, Guidelines

**Assumptions:**

* It is assumed the user have basic computer literacy skills.
* The minimum requirements will be met as per scope.
* During the implementation of the solution, there will be no modification of the project scope.
* All milestones will be completed on schedule.
* End-user has a web browser installed.
* End-user has stable, uncapped internet on the NWU campus.

**Constraints**

* The product should meet the minimum criteria determined in the project scope and provide the least viable version of it to the predetermined timeline and budget indicated in the project plan and its’ appendices.
* The development team should adhere to the proposed budget.
* The scope of the project is based on the results and feedback generated by the feasibility study and business case.

**Dependencies**

* The application is dependent on the Internet and various text comparison and stylometry libraries.
* Source texts and alleged texts
* Database that can store statistics provided on the reports.

**Guidelines**

* Project development will flow from the Project plan.
* Implement the COBIT-19 framework standards.

## Design and Implementation Constraints

The product is bound by the specifications provided as part of the bid and, as such, must satisfy the client's minimal needs. Therefore, no further functionality beyond those previously specified are to be implemented.

## User Documentation

A user manual will be provided to the users, that will guide and demonstrates them how to utilize the web application. Additionally, the functionality will also extensively be described.

# REQUIREMENTS

## External Interface Requirements

For additional discussion of what is needed for and from this project to be successful for the client/user, this section will provide clarification on each component that will be listed.

### User Interfaces

The following components form the user interface that will be utilized for this project:

* Login Screen
* Registration Screen
* Home Screen
* Quick Text Comparison Screen
* Extensive Text Comparison Screen
* Stylometry Screen
* Corpus Stylometry Comparison Screen
* Help User Manual Screen

### Hardware Interfaces

There are no specific hardware needs.

### Software Interfaces

The only software interface the user needs is an internet browser such as Chrome, Firefox, Edge etc.

### Communications Interfaces

There is no specific communication interface implemented in this project/product.

## Functional Requirements

### Purpose

The NWU Registrar must investigate each case individually and seek the technical advice of experts in order to prevent contract cheating. External subject matter experts (SMEs) are entrusted with analysing the technical reports with an additional report that offers a more thorough explanation of the claimed plagiarism if the technical reports do not underline the seriousness of the suspected plagiarism. The technical need is to manually compare the allegedly duplicated data in question with the original wording of the evidence. The registrant must also use stylometry to evaluate authorship attribution in order to provide reports. The most important features of the application are:

* Users should be able to complete tasks in a time efficient manner.
* Enhance the productivity of employees by providing functional software.
* At the conclusion of the project’s lifetime, Nightcore Mech's engineers should be able to create a system that enables instructors at NWU to detect academic misconduct and contract cheating.
* The system should provide feedback reports highlighting text similarities between diverse sources in order to identify varying levels of student plagiarism.
* Automation of feedback reports with stylometric performance indicators displayed to identify contract cheating and original authorship.

### Inputs

From the Registrar and lecturers:

* Login Details
* Registration Details
* Source Documents
* Comparison Documents
* Corpora of documents to identify authorship identification

### Processing

**Login/Register**

Once the user is registered and logged in, they can decide whether to perform text comparison or authorship attribution.

**Text Comparison**

Regarding text comparisons, the user can decide whether to do a quick or extensive text comparison. When the user selects a quick comparison, the user needs to upload a source and comparison document respectively.

**Quick Text Comparison**

The user needs to select an algorithm to compare the text, either via line, sentences, or substrings. If substring is selected, the length of the substring needs to be specified. Once the user submitted the documents, a new browser window will open, displaying highlighted text similarities between the source and comparison document. A percentage of text similarity will be displayed on the screen.

**Extensive Text Comparison**

The user needs to upload a source document and a corpus respectively. Once the user submitted the documents, a new browser window will open, displaying highlighted text similarities between the source and the documents from the uploaded corpus. A percentage of text similarity will be displayed on the screen.

**Stylometry Comparison**

The user needs to upload a source document and a corpus respectively to identify authorship attribution. Once the user submitted the documents, a new browser window will open. On this window, the result will display if the student is indeed the author of the submitted document. Stylometry statistics will be displayed to indicate authorship as well as a clustering chart.

**Reports**

A report is generated including the statistics of the text comparison and charts\statistics of the stylometry authorship functions.

### Outputs

**Help**

A user manual indicating how the user should utilize the application, as well as an in depth explanation of the functionalities of the application.

**Reports**

A report will be generated that will display the text comparison statistics as well as the stylometry statistics and charts/graphs indicating authorship attribution. The reports should contribute to identify contract cheating. Additionally, a user should be provided with options to save reports locally to their device or upload the reports to cloud storage. As reports are generated, source documents should be added to the corpus to increase the application’s accuracy in identifying contract cheating.

## Performance Requirements

* The application must be responsive.
* The software should adhere to end-users and the Registrar’s requirements.
* The application should be able to handle large loads.
* The application should be memory efficient.
* The application must be web optimized.
* The reports should be generated as quickly as possible.
* The accuracy of the reports is of utmost importance.
* The application must be user friendly

## Security

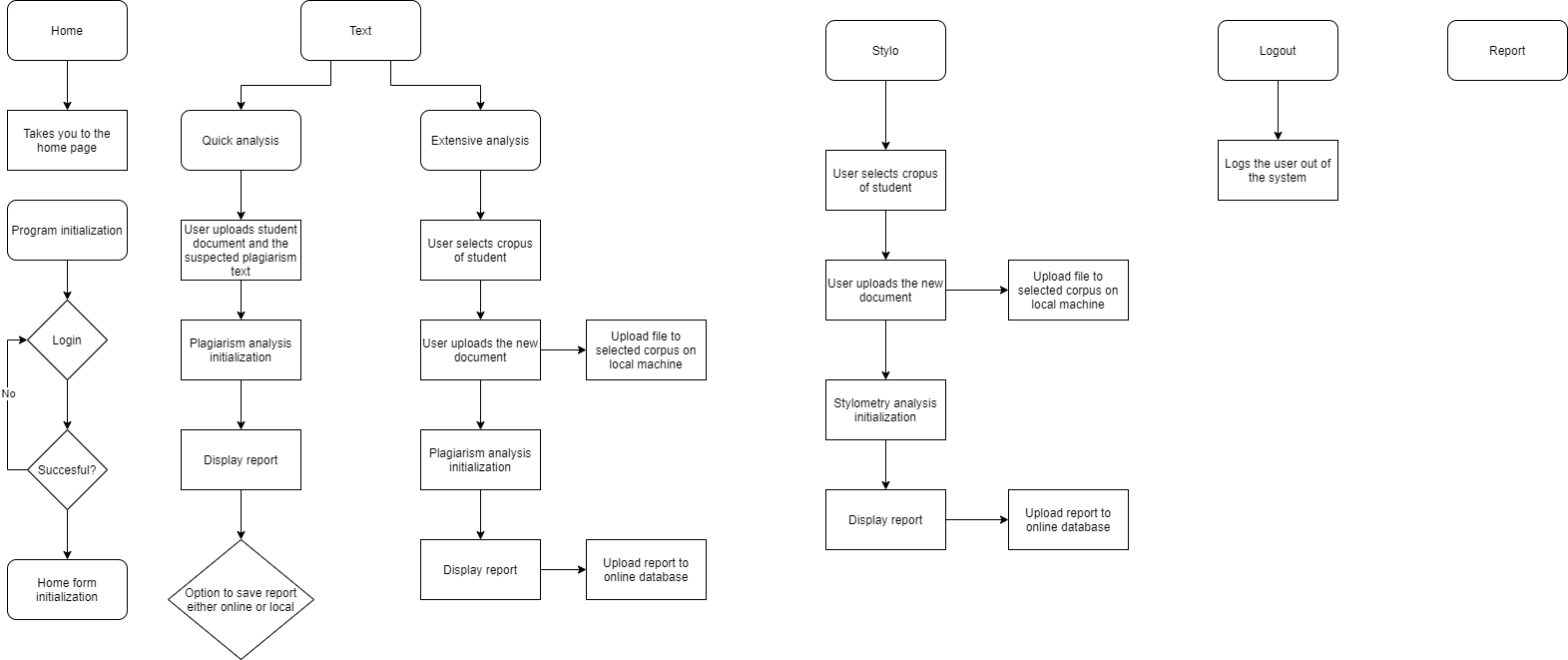
* The password will be hashed.
* The user and the Registrar will have admin privileges.
* The web application will implement HTTPS for secure communications.

## Usability

Usability describes how the user should experience the system's functioning.

* The user interface of the system must be simple to understand.
* Task efficiency is a requirement that the system must meet.
* Recallability; the system must be simple for the user to recall.
* The system's functionality must be clear to the user.
* The system must leave the user happy.

## Data Flow



## Other Requirements

There are currently no other requirements.

1. Analysis Models

List any attached / referenced documentation such as data flow diagrams, class diagrams, state-transition diagrams, entity-relationship diagrams, etc.

| ANALYSIS MODELS | | |
| --- | --- | --- |
| DOCUMENT NAME | DESCRIPTION | LOCATION |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Issues List

Detail any unresolved issues.

| ISSUES LIST | | |
| --- | --- | --- |
| ISSUE ID | ISSUE DESCRIPTION | STATUS |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |