

Software Requirements Specification

for

Plagiarism Checker

Version 2.0

Prepared by: Grenel Sumabat

and

Dong Nguyen

2/19/2017

Table of Contents

- Table of Contents.....1
- Project Agreement and Statement of Work.....2
 - 1. Background and Initial Obligations.....2
 - 1.1 The Buyer.....2
 - 1.2 The Seller.....2
 - 1.3 The Project.....3
 - 1.4 Other Legal Stuff.....4
 - 2. Project Specification and Requirement.....4
 - 2.1 Objectives and Background.....4
 - 2.2 Operating Environment.....4
 - 2.3 Designing.....5
 - 2.4 Implementation Constraints.....5
 - 3. System Features.....5
 - 3.1 Variable Names.....5
 - 3.1.1 Functional Requirements.....5
 - 3.2 Comparing Code.....6
 - 3.2.1 Functional Requirements.....6
 - 3.3 Programming Languages.....6
 - 3.3.1 Functional Requirements.....6
 - 3.4 Syntax.....6
 - 3.4.1 Functional Requirements.....6
 - 3.5 Memory.....6
 - 3.5.1 Functional Requirements.....6
 - 3.6 Uploading Document.....7
 - 3.6.1 Functional Requirements.....7
 - 3.7 Feedback.....7
 - 3.7.1 Functional Requirements.....7
 - 3.8 Archive.....7
 - 3.8.1 Functional Requirements.....7

PROJECT SPECS AND REQUIREMENTS

Document version: 2/19/2017

Project Agreement and Statement of Work

THIS PROJECT AGREEMENT AND STATEMENT OF WORK (this “Agreement”) is entered into by and between the undersigned Buyer and Service Provider as of the Effective Date. The Buyer and Service Provider are sometimes referred to collectively herein as the “Parties” and individually as a “Party”. All capitalized terms not defined in this Agreement have the meanings given to such terms in the Terms of Service Agreement (“Terms of Service”) available, unless the context requires otherwise.

1. BACKGROUND AND INITIAL OBLIGATIONS

1.1 The “Buyer” is:

Frank Witmer
Computer Science & Engineering
ECB 308B
907-786-1637
Email: fwitmer@alaska.edu
Web: <http://www.cse.uaa.alaska.edu/~fwitmer/>

1.2 The “Seller” is:

Grenel Sumabat
510 Tyee Cir
Anchorage, Alaska 99503
Email: gsumabat2@alaska.edu

Dong Nguyen
518W 75th Ave
Anchorage, Alaska 99518
Email: dlnghuyen@alaska.edu

1.3 The Project is:

The Plagiarism Checker is a program that validates a student's submitted program.

Delivery Date:

1. Project Description

The Plagiarism Checker validates whether a student’s submitted program has been copied or plagiarized from other submitted programs. This application will analyze and gather the similarities between the programs that have been submitted to the database.

2. Key Assumptions

Uploading code to the checker field, then it will go through every single line to check if the submitted code is copied or plagiarized from the other sources.

3. Scope of Service

Comparing the given code to the public database. It will go through some specific familiar webpages, such as StackoverFlow, Codein, etc. This program will detect instances of plagiarism within the text document, and will bring a tool to oppose the cut, copy and paste culture which is strongly recommended at the core fundamental values of academic integrity.

4. Milestone Deliverables

Milestone Deliverable (Task need to be done)	Delivery Date
1. Meeting with sponsor and get general idea about the project	February 2, 2017
2. Submission of first draft for requirements and specification	February 5, 2017
3. Submission of rough draft for visual aids	February 12, 2017
4. Submission of second draft for requirements and specification	February 19, 2017
5. Submission for final review	April 1, 2017
6. Final Delivery Date	April 15, 2017

1.4 Other legal stuff:

After final submission and testing of the application, the developers will hand over the source code to the sponsor and will not have any responsibility of maintaining the software.

2. PROJECT SPECIFICATION AND REQUIREMENTS

2.1 Objectives and Background

A “Plagiarism Checker” is a program building base on webpage interface. It helps to check if student’s works are taken from any online resources without citation. The program allows vast collections of documents to be compared to each other, making successful detection much more likely.

Plagiarism checker is used for checking some courses at UAA such as CSCE A201, or A202.

Database capacity will be able to hold all the data from required course.

The program will give a feedback about percentage of similarity between student’s code and the online database with highlighting plagiarized lines.

Each file must be submitted with valid extension, unless the program will terminate with an error message.

Our goal on this project is producing the final project meet sponsor expectation. The “Plagiarism Checker” can help:

- a) To detect instances of plagiarism to ensure honesty, trust, fairness, respect, and responsibility.
- b) To ensure academic integrity by detecting plagiarism.
- c) To create the environment of academic morality and academic honesty.
- d) To encourage academic sincerity and work originality among students.
- e) To abase duplicity, cheating and fraudulent work.

2.2 Operating Environment

This program works on most common web browser such as Safari, Chrome, Firefox, etc. For efficient and effective execution of the software, other requirements are:

- Database Server
- Global Database

- Internet Connection
- Search Engine

2.3 Designing

The program is put into system design, which can be used for future references. Designing focuses mainly on attributes such as:

- Data Structure
- Software Architecture
- Interface Representation
- Procedural Details

2.4 Implementation Constraints

The program will need internet connection because it is linked to a search engine. So, if there is problem with the internet connection then it could hamper the desired result. Search engine uses its own global database to find plagiarized code or code structure in the submitted works.

There are some constraints with this program:

- a. For checking of instances of plagiarism requires that the users should declare the specific programming language they want to check.
- b. Uploading of scanned images or screenshots is not allowed.
- c. Be sure that all syntax and braces are carefully checked before submitting.
- d. Should be used under trustful internet connection.

3. SYSTEM FEATURES

3.1 Variable Names

The application will verify if just the variable and class names were changed. This feature is a medium level priority.

3.1.1 Functional Requirements

After submission, if only the variable names and class names were changed, then they will be highlighted to notify the user has plagiarized everything besides and just changed the names.

3.2 Comparing Code

The similarities of submitted assignments will be detected by comparing parts of the code with other submitted codes within the database. The feature is marked as a high-level priority.

3.2.1 Functional Requirements

The similar codes should be highlighted to acknowledge which parts were too similar or identical. If the compared codes don't have any similarities, then it has successfully passed the checker.

3.3 Programming Languages

This application can validate various types of programming languages and not only applicable to Java and C++. This feature is marked as a low-level priority.

3.3.1 Functional Requirements

TBD

3.4 Syntax

The application will check if the code only contains common syntaxes or similarities in small simple code and if so, the user will be notified. This feature is marked as a low-level priority.

3.4.1 Functional Requirements

TBD

3.5 Memory

This application will contain a back-end design to free up available memory space for later use in the future. This feature is a medium level priority.

3.5.1 Functional Requirements

TBD

3.6 Uploading Document

This application will have a feature to upload a single document or multiple documents to check whether they're in a readable format or not i.e. .zip, .cpp, .doc, .txt, or pdf. This feature is marked as a high-level priority.

3.6.1 Functional Requirements

The user will be able to choose which file they'd like to import and upload to the web application. If the document format is not applicable, then it will return an error message saying that the format type is not supported with this application.

3.7 Feedback

This application will be able to send feedback to the user after submission of the student's code, to be checked. This feature is marked as a medium level priority.

3.7.1 Functional Requirements

The feedback of the submitted code will be a percentage of how much of the submitted program was detected as being plagiarized. If the given percent surpasses the minimum requirement, then the student's code is valid. During batch processing, the program will extract the student's name from code comments when uploaded.

3.8 Archive

This application will contain an archive of all submitted assignments within the recent years. This feature is marked as a high-level priority.

3.8.1 Functional Requirements

The archive will have organized files from years past that are easily accessible. The submitted programs will be checked against the archive database to ensure no plagiarism is occurring with previous student's code.