COS 301 Project Proposal

The Code Detective

version 1.0

Client Name: Department of Computer Science
Contact Person: Prof AP Engelbrecht
Contact Email: engel@cs.up.ac.za
Contents

1 Project Name 1

2 Vision and objectives 1
   2.1 Vision ........................................... 1
   2.2 Objectives ...................................... 1

3 Owner 1

4 Scope 2

5 Architectural requirements 2
   5.1 System Performance .......................... 2
   5.2 Quality Requirements ......................... 2
   5.3 Security ........................................ 3
   5.4 Integration and Access Requirements ........ 3
   5.5 Technology Preferences ....................... 3
   5.6 Maintainability ................................ 3
   5.7 Administrability ................................ 3

6 Skills Requirements 4

7 Project Deliverables 4

8 Intellectual Ownership 4

9 Client Commitments 4
1 Project Name

This project will be referred to as the Code Detective (CD).

2 Vision and objectives

Code plagiarism has become a serious issue, both at Universities where programming is taught and in industry. Detection of code plagiarism is important for both: It is essential that a student is not credited for work that he/she has not done, and for industry, intellectual property needs to be protected, as it inevitably is direct linked to money.

2.1 Vision

A very extensive sets of plagiarism detetion tools will be developed, and a web-based front-end to this tool set will be provided. The plagiarism detection tools will examine both the literal expression and the functionality of software.

2.2 Objectives

The objectives of CD are as follows:

[a] To provide a set of validated tools to detect code plagiarism.

[b] Provide a web-iterface for interaction with the tool set, allowing for uploads of code to be examined and for reports to be presented, viewed, and/or exported to pdf format.

[c] It should be possible to compare only two sets of programs, where each may consist of a number of files, organized in folders or all in one folder. It should also be possible to compare a collection of programs amongst one another.

- It should examine syntactic similarity, grammatical/semantic similarity, and functional similarity.

- If an application makes use of a database, then the structure of the database also needs to be examined, by examining the entity-relation diagrams (ERDs).

3 Owner

The owner will be the Department of Computer Science, University of Pretoria. The contact person is Prof Andries Engelbrecht, head of the deaprtment, and he will be the contact person. Appointments to see him should be made via Mrs Angela Bekker (abekker@cs.up.ac.za), or alternatively, communication can be via email to engel@cs.up.ac.za.
4 Scope

Examination of programs for detection of plagiarism will be done via a web-based interface. Users will be provided a user name and password to the system. Provisioning should be made for a system where users pay for usage based on different packages (e.g. monthly/yearly license, pay per analysis, pay per number of aspects analyzed).

The system should not be limited to a specific programming language or specific database management system.

The system should include a number of plagiarism detection tools, including mutual syntactical similarity, directional syntactical similarity, file similarity source code correlation measuring code evolution, excluding certain source code such as open source or third party code from the analysis, analysis with the original code, with white space removed, with anonymization of identifiers, and then also a comparison of the resulting grammar trees.

When a database is used, the corresponding ERDs will be compared.

The user should be allowed to specify which tools from the tool set to be used.

Reporting should be done in an efficient manner, with cross-linking between the different files where similarities have been detected. Plagiarism scores will be reported.

Security of stored results have to be ensured.

5 Architectural requirements

5.1 System Performance

Plagiarism detection is usually an offline process, and it is not necessary to have real-time response. However, the detection tools are relatively computationally expensive, and execution time will grow significantly with increase in lines of code, number of files, and number of programs to examine. All effort must therefore be invested in optimizing computational efficiency of all of the components of the CD system.

Potential to parallelize execution of the detection tools exist.

Care should be given to how the data (code) will be handled. Will the analyses be done on the client or the server. If the latter, then there will be significant demands on CPU, memory, hard drive space. For the latter, reports will then also be archived. For the former, data and reports will not be archived.

5.2 Quality Requirements

The system should be generic and scalable. It should be very easy to add a new detection tool to the tool set. It should be scalable to handle thousands of lines of codes and hundreds of programs to be compared amongst one another.
The actual tools have to be validated, tested; they have to be bullet proof should the tool be used in plagiarism cased that will end up in legal actions against those who have plagiarised.

When a tool is executed a number of times on the same code set, the results should not differ. That is, only deterministic methods should be used to do the analysis.

5.3 Security

The scope of data visibility will be determined by the user, who will specify access rights such as private (no one else may see), protected (only a group of specified users may see data and reports), public (anyone can see the data, and may also compare their data against public data).

Generated reports, if exported to an electronic format such as pdf should be authenticated in some way. Either by means of a unique report number, or by some bar code, QR code or data matrix. For data stored on the server, it should be possible to repeat the analysis associated with a given code and produce the same result.

5.4 Integration and Access Requirements

Access will only be via the web-based interface.

The system should be a stand-alone application, but also should allow integration with the department’s assignment upload system.

5.5 Technology Preferences

There are no preferences and the team should identify the most appropriate frameworks, programming languages, platforms, etc.

5.6 Maintainability

Because there is potential that new detection tools will be added over time, the system must be very maintainable. The web-based interface has to be very adaptable, and not require redesign and re-implementation as the number of tools increases.

5.7 Administrability

Creation of new users, creation of user groups, provisioning of user names and passwords have to be an efficient process. Creation of a new user has to be made secure, by providing an email to the new user to confirm registration to the system before user account is created. No password should be emailed.
6 Skills Requirements

Instead of listing skills, we list below the Computer Science subject areas that will be very important for this project:

- Very good programming skills
- Netcentric computing
- Information security
- A very good understanding of programming languages and compilers

In addition, the team needs to show innovative skills and research skills.

7 Project Deliverables

The following deliverables:

- all of the source code;
- all of the documents, most importantly a user’s manual, installation manual;
- all of the installation scripts;
- a working prototype system.

The objective of this project is to, in the end produce a product that can be installed and used.

8 Intellectual Ownership

The IP will belong to the University of Pretoria.

9 Client Commitments

The departmental servers will be used to host the system.

Literature about code plagiarism detection will be provided, but, the team also needs to search for more information and detection mechanisms The departmental servers will be used to host the system.

Literature about code plagiarism detection will be provided, but, the team also needs to search for more information and detection mechanisms.

Two-weekly meetings will be scheduled with the team, and it will be required that all members of the team attend these meetings.