Project Management Document
Parametrized String Matching Implementation for Software Plagiarism Check

Supervised By : **Prof. Srinivasaraghavan G**

Team

amit.tomar
(MT2013008)

siddhesh.dosi
(MT2013150)

srinivas.r.vaidya
(MT2013152)

@iiitb.org

10 - April - 2014

1 Introduction

This project aims at developing a Parametrized String Matching Implementation for Software Plagiarism Check, that given a collection of files which contain code in some programming language, will show a set of possible duplications of parts of the code among these. Comparing pieces of software will require discounting comments (optional and language dependent), extra/blank lines and spaces, variable renaming etc. The theory of parametrized string matching will be used to implementat this project. System will have an easy-to-use UI for selecting files/folders and shall report the plagiarism related information (matches found) in the UI in a nice manner.

2 Project plan

1. **Activity list**

   (a) **Documentation : Requirement Specification** :
   Date of submission : 7 - Feb - 2014
   (Already submitted to Prof. Srinivasraghvan)

   (b) **Literature Survey** :
   Expected date of completion : 15 - April - 2014
   Expected time : 45 Hours

      i. **Reading relevant background information** : 15 Hours
      ii. **Understanding and documenting the requirements** : 15 Hours
      iii. **Discussions** : 15 Hours

   (c) **Designing a solution : Building suffix tree data structure**:
   Expected date of completion : 15 - May - 2014
   Expected time : 30 Hours

   (d) **Coding : Identifying duplicate code using suffix tree**:
   Expected date of completion : 1 - June - 2014
   Expected time : 25 Hours

   (e) **Coding : Implementation of UI**:
   Expected date of completion : 10 - June - 2014

Expected time : 10 Hours

(f) **Coding : Parameterized implementation for software plagiarism check**:
Expected date of completion : 1 - July - 2014
Expected time : 40 Hours

(g) **Integration : UI with parameterized string matching code**:
Expected date of completion : 15 - July - 2014
Expected time : 20 Hours

(h) **Testing**:
Expected date of completion : 31 - July - 2014
Expected time : 20 Hours

2. **List of final deliverables**:

(a) Requirement specification document.

(b) Design document.

(c) User manual.

(d) Deployment manual.

3 Testing Strategy

TestCase	Populate the folder structure of file system.
Input	File option
Output	User prompted to select files from multiple folder.
Responsibility	Amit Tomar

TestCase	File path validation
Input	Browsing and selection of files and next button
Output	User is promted to enter code snippet to be ignored while processing
Responsibility	Amit Tomar

TestCase	Generate parameterized suffix tree to check amount of plagiarism.
Input	Check plagiarism
Output	Plagiarism related log is generated.
Responsibility	Siddhesh Dosi

TestCase	Search for all the files in the selected folder and populate a list.
Input	Folder option
Output	User is prompted to select Folder.
Responsibility	Siddhesh Dosi

TestCase	File path validation
Input	Selection/Deselection of files from the populates list and next button
Output	User is promted to enter code snippet to be ignored while processing
Responsibility	Srinivas Vaidya

TestCase	Generate parameterized suffix tree to check amount of plagiarism.
Input	Check plagiarism
Output	Plagiarism related log is generated.
Responsibility	Srinivas Vaidya

4 Goals of implementation

Plagiarism is a serious issue in computer science courses involving assessment of programming assignments [1]. Being electronic in nature, it is very easy to copy code and it is difficult to differentiate between the original and copied work. Thus, there is a need for a tool to detect plagiarism automatically, assisting professor to check for any kind of copying done by students.