ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ

VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

REVA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

8th Semester Project Proposal Presentation on

TITLE OF THE PROJECT:

<<TITLE OF THE PROJECT>>

Presented By,

Midhun Mathew (1RE13CS088)
Ruhil Jaiswal (1RE12CS085)
Karthik Hegde (1RE13CS062)

Under the Guidance of

Dr.M Prabhakar,
Associate Professor

Content

- Introduction
- Abstract
- Literature Survey
- Problem Definition
- Enhancements
- Hardware/Software Requirements
- Methodology/ Technique to be applied
- Conclusions
- References

Introduction

In a competitive environment, it is often required to find the program that runs most efficiently among several programs. Currently few to none applications are available that compares programs and analyzes their efficiencies. This system will not only analyze the code, but also have a code repository that stores the most efficient functions as and when encountered.

Abstract

A Performance evaluator engine will analyze the code and give the result in a visual representation. We create a central repository which stores the best functions which can be used by other users and also for further comparisons and optimization.

Literature Survey

Why do we need it?

The major advantages of such a system are :

- reusability of code
- multi language compatibility
- compare code
  - assessments
  - ranking
- visual results for the function
- retrieval of functions, mapping with tags
- testing using restrictions/ constraints on the input

Scope for future work

- handling the library requirements for the function
- tag based retrieval (porting / eval function)
- visual aspects for code development (flowcharts, class diagram)
- optimization
- analyser improvements to identify syntax of code

Hw/Sw Requirement

Software requirements :

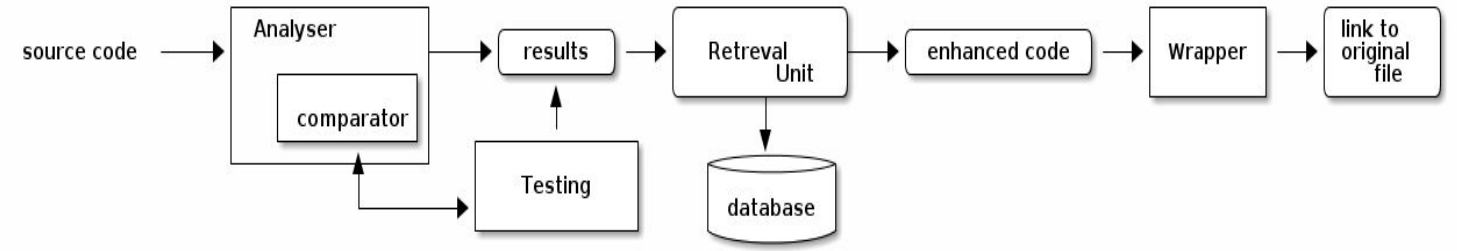
- Operating System :
  - server: Linux (ubuntu 16.04)
  - database: Linux (ubuntu 16.04) / windows
- Tools: Python, Mongodb, CherryPy, Docker

Hardware requirements (developed on)

- Processor : intel core i5
- Memory : 4GB Ram or higher
- Hard disk space : 2GB or more

Note : system requirements maybe subjected to change.

Methodology/ Technique to be applied



Conclusions

- visual aid for development and analysis
- compare code of different languages
- reduced plagiarism
- credit for work
- frequent enhancements

References

Enrique FLores ,Alberto Barron-Cedeno Towards the Detection of Cross-Language Source Code Reuse(2011)

Arwin, C., Tahaghoghi, S.M.M.: Plagiarism Detection across Programming Languages

D. Binkley "Source Code Analysis: A Road Map" IEEE Future of Software Engineering (FOSE'07)2007.