|  |  |
| --- | --- |
| **Project Title:** Code Plagiarism Detector | |
| **Start Date:** 9/18/17 | **End Date:** 12/8/17 |
| **Project Manager:** Kevin Smith | |
| **Project Sponsor:** Professor Bingham | |
| **Customer:** UMKC | |
| **Users:** educators, employers, programmers | |
| **Purpose (Problem or opportunity addressed by the project)**:  Software plagiarism, whether it is intentional or unintentional, is a growing concern with potential liability as application development continues to expand. A tool to identify potential code plagiarism is needed at the academic and professional levels. Programmers themselves may also desire a comparison report before submitting code for review to their teams and/or superiors. For intentional plagiarism, it is easy to manipulate observed code to give the appearance of uniqueness for one’s own code. For unintentional plagiarism, it is easy to assume relative uniqueness of one’s code if little to no code was observed beforehand, or if observed code seemed to solve a relatively similar, yet different issue.  The goal is to provide a comparison metric between documents, not a suggestion or confirmation on plagiarized code. Confirmation of plagiarism should be determined by the user(s) and/or organization based on their own standards. | |
| **Goals and Objectives**: The goal of the application is to provide a metric derived from compared code documents and an output of the similarities.   * The application should be easy to use for users who are familiar with online forms and document uploads. Minimal computer experience should be required. * The application should allow for comparisons for multiple programming languages, specified by the user. Easily manipulated factors, such as variable names, function names, and whitespace, should be accounted for. * Upon completion, a report should generate an outline of the areas of similarity and a percent value of matching code. * Users should be able to review past code comparisons generated by the application. | |
| **Schedule Information (Major milestones and deliverables)**:  09/20/2017 - Preliminary Requirements Complete  09/22/2017 - Project Charter Approved 10/02/2017 - Product Feature Set Baselined 10/04/2017 - Preliminary Project Plan Complete 10/09/2017 - Iteration #1 Complete 10/18/2017 - Candidate Architecture Complete  10/18/2017 - Architecture Document Complete 10/23/2017 - Iteration #2 Complete 11/06/2017 - Iteration #3 Complete 11/27/2017 - Iteration #4 Complete 12/04/2017 - Code Freeze 12/08/2017 - Final Presentation | |
| **Financial Information (Cost estimate and budget information)**: This project is a necessary requirement for completion of COMP-SCI 451R at UMKC and carries no monetary cost. | |
| **Project Priorities and degrees of freedom:** The priorities are that this application should be easy to use, agnostic regarding coding language implementation of the compared files, and able to provide metrics on and locations of code similarities. Coding language freedom of the application itself has been granted by Professor Bingham. | |
| **Approach:** An iterative and incremental approach is planned. The algorithms and design are deemed the highest initial priorities given the nature of the application. Consensus on the coding language(s) and additional tools will occur prior to implementation. | |
| **Constraints**: This must be a web-based application with user login capabilities. | |
| **Assumptions**: We assume that users must create login credentials to use the application and it is free to use. Also, we assume that reports will only be viewable online and not downloadable in a format, such as PDF. | |
| **Success Criteria**: The application will be deemed successful if it is completed by the end of Iteration 5 and, after testing, we are able to determine high degrees of accuracy in analyzing and reporting code similarities. | |
| **Scope**: This application will be designed to analyze and report on code similarities while not deeming code as plagiarized. Confirming that code is or should be considered plagiarized should only be determined after human analysis using standards derived from the user(s)/organization and/or additional software written that understands the standards of the user(s)/organization. | |
| **Risks and obstacles to success**: One or more members of the group may have little to no experience working in a group atmosphere such as this. Varying experience and skillsets could provide challenges regarding how the application should/will be implemented versus how many in the group are comfortable with those technologies. Ideally, acclimation to unfamiliar technologies is minimal, so the workload of each group member is fair and balanced. | |
| **Signatures**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Manager**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Sponsor**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Customer**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Technical Lead** | |