Plagiarism Detection System

Project Plan

Objective:

The goal of this project is to build an interactive tool for detecting plagiarism in programming assignments. This is an application which will determine if there are any similarities between two sets of programs. The application would check for transformations like renaming of variables, classes and methods, extracting code into methods and code being moved around. The User Interface of the system will be user friendly making sure every user can easily interact with the system and use its functionalities at ease.

Technologies:

The logic for the plagiarism detector will be implemented in Java. However, the detector will target **Python** Code.

We will be using a Web based approach for the system. The front end will be written using HTML5, CSS3, JQuery, JQueryUI and REACT. We will be using Bootstrap framework for the designing of the website. In order to keep track of two types of users for the application we will be using MySQL to store the state of the user. The backend will be implemented using Java which will run the process of plagiarism detection on the system. The frontend will show a detailed report of the comparison based on what the backend provides.

Project Plan:

- **Sprint 1:** Use cases are created. Languages and Frameworks which are going to be used in the system are decided. User Requirements are gathered. Basic mockups for the User Interface are created.
- **Sprint 2:** Design Patterns will be generated. The basic design of the system will then be converted into UML diagrams.
- **Sprint 3:** The actual implementation of the application starts here. Initially, the environment of the system is setup which includes software installations, IDEs and prerequisites. Since there are two types of users, at this stage a database is set up.
- **Sprint 4:** The backend development begins here where the logic for the plagiarism detection is implemented.
- **Sprint 5:** Once we have a few features of the backend ready, the frontend implementation is also initiated such that by the end of this stage both the components are ready.
- **Sprint 6:** Integration of the front end with the backend to make sure the functionalities as stated in the requirements are fulfilled.
- **Sprint 7:** At this stage, functional testing and structural testing is done based on which the bugs are fixed.
- **Sprint 8:** Refactoring of code and regression testing is done to make sure all the functionalities are still working as expected.