Plagiarism Detector for Code Files

Om Dalwadi, Ravinder Kaur, Yug Shah

20^{th} October 2022

§1 Introduction and Problem Statement:

We have noticed that in multiple Computer Science courses at the University of Regina when the class assignments required submissions of code files, the instructor had issues with the Turnitin software. It does not allow checking code files for plagiarism thereby making it difficult for the instructor to maintain the integrity of the course assignments. It becomes the instructors' responsibility to either manually compare all the code files with each other or introduce some form of assessment criteria in the assignment which could allow for easy checking of the same, for e.g. having some digits of the student's ID as a required input in some function, etc. Since checking the assignments is a time-sensitive task for the teaching assistants and markers, especially during the Spring/Summer semesters, it becomes a difficult task. This is the problem we are attempting to solve through our group project for the CS 375: Database and Information Retrieval course.

§2 Project Description and Overview:

In order to solve the above-mentioned problem, we will be developing a web-based software application that will allow for comparison between two code files and generate a similarity score based on this comparison. The instructor will be able to upload the code files submitted by the student into the application in order to store them in the database. When required, the user (instructor) can compare any of the submitted code files with the other files in the database to see if there are any instances of plagiarism. The application will then generate a similarity score for that specific code file.

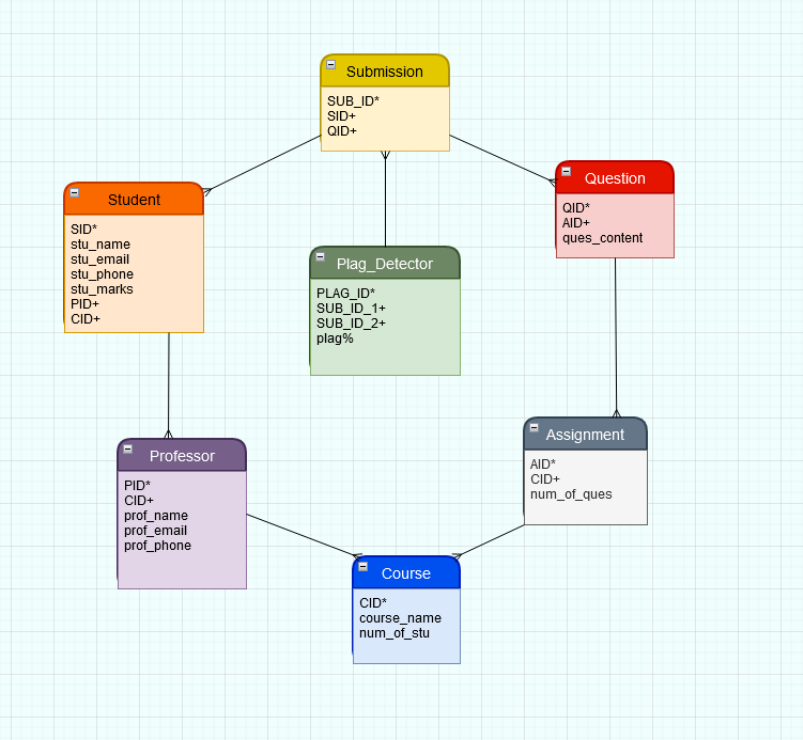

Figure 1: Entity-Relation diagram for the database

§3 Tech-Stack:

1. Database Management System
    - MySQL RDBMS
2. Front-End Development:
    - Hypertext Markup Language(HTML5)
    - Cascading Style Sheets(CSS)
    - ReactJS Framework
    - VisualCode
3. Back-End Development:
    - Node.js, D3.js JavaScript Libraries
    - Express
    - VisualCode
4. Presentation
    - Overleaf - LaTeXEditing
    - Shotcut - Video Editing
    - Free Cam 8 - Video/Screen Recording and Editing

§4 Workload Distribution:

1. **Om Dalwadi: Documentation and Front-End Development**

   ☐ Documentation using LaTeX

   - Project Proposal
   - Implementation of 3 Layered Architecture Diagrams and other required Diagrams

   ☐ Front-End Design

   - Designing the user interface based on the principles of UI/UX design using HTML5 and CSS
   - Creating functions for form validation in JavaScript for sign-up and login
   - Linking the website to the database to store/upload the required files

   ☐ Testing different components for Form Validation

   - Testing the form validation to check the functionality for sign-up requirements, login credentials, uploading required type of files

   ☐ Preparing Presentation

   - Video Editing using Shotcut Software
   - Preparing presentation slides using LaTeX

2. **Ravinder Kaur: Back-End and Database Design**

   ☐ Database Design

   - Specify the primary/foreign key constraints for each relational table.
   - Define and establish the relationship between the tables.
   - Creating the Entity-Relation Diagrams for the database.
   - Apply the database normalization rules to normalize each table.
   - Creating SQL queries for all the required functionalities.

3. **Yug Shah: Documentation and Back-End for Front-End Development**

   ☐ Documentation using LaTeX

   - Prepare the Project Report satisfying all the different requirements.

   ☐ Back-End for Front-End

   - Create a function in JavaScript to compare two code files.
   - Generate similarity score based on the output of the file comparison.
   - Provide functionality to upload code files and store them in the database.

   ☐ Database Implementation and Testing

   - Create and build the database with the MySQL DBMS
   - Test the database for data accuracy, consistency, and integrity.