Member 1

Input :

- Student’s answer script (source code) as a text file
- Lecturer’s answer script (source code) as a text file

Process :

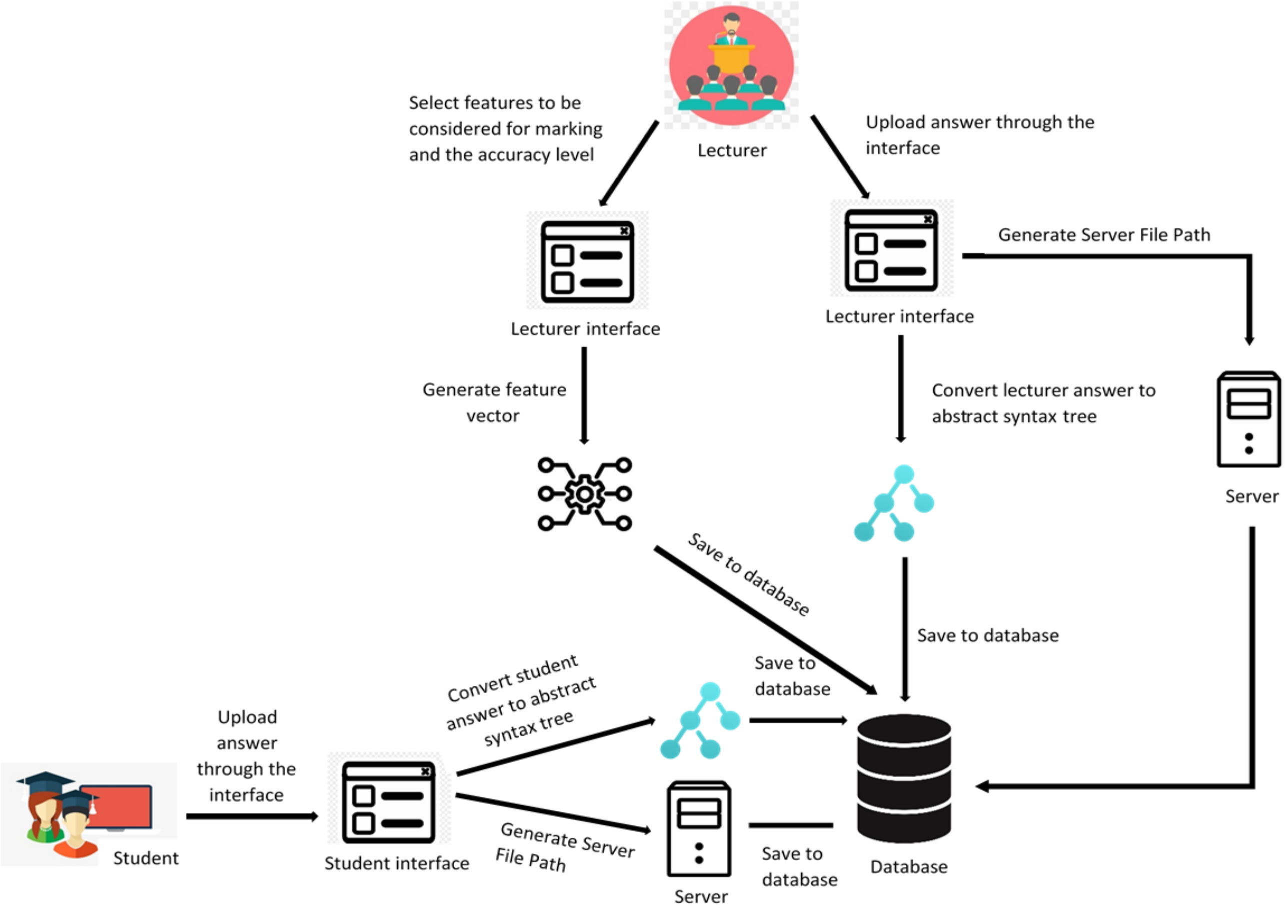
- Initially the lecturer goes to the specific module location and uploads the correct answer sheet. Then the student logs in and uploads his answer sheet.

- The program source code of the answer sheet uploaded by the lecturer will be converted into an abstract syntax tree/parse tree as well as a copy of the original file will be stored in the database. Then the source code of the answer sheet uploaded by each student will be converted into an abstract syntax tree/parse tree and be added to the database along with a copy of the original file. The abstract syntax tree conversion should be done using a slightly modified version of an existing algorithm or a new algorithm should be used. Can use ANTLR package for the conversion and to generate the output as a GUI tree structure.

- Parallely, when the lecturer wants to start comparison, he should be able to generate a customized feature vector. The feature vector would be customizable and configurable by the academic staff. They can configure it based on the areas they want to consider while marking the students source code using a checkbox :
    - ✓ same method names
    - ✓ same variable names
    - ✓ same class Name
    - ✓ Switch case
    - ✓ If else
    - ✓ For loop
    - ✓ For each
    - ✓ Do while
    - ✓ While
- Based on their configuration, different feature vectors would be generated from a source code. The staff can customize the feature vector to be exactly as the source code of the marking rubric, slightly flexible or extremely flexible but having the same logic. If it has to be exactly similar to the marking rubric source code, the feature vector would be all 1s. If it is fully flexible but having the same logic, it will be all 0s. if its slightly flexible it will be having a set of numerical values. Based on the customization level the feature vector would be generated.

Output :

- Abstract syntax tree of Student’s source code to be printed as a GUI structure
- Abstract syntax tree of Lecturer’s source code to be printed as a GUI structure
- Customized feature vector


Lecturer
Select features to be considered for marking and the accuracy level
Upload answer through the interface
Lecturer interface
Lecturer interface
Generate Server File Path
Generate feature vector
Convert lecturer answer to abstract syntax tree
Server
Save to database
Save to database
Convert student answer to abstract syntax tree
Save to database
Upload answer through the interface
Student
Student interface
Generate Server File Path
Server
Save to database
Database