

Plagiarism Detection Tool

Team 206

Team Members

Mohd Asim Khan

Sanket Mathur

Jayanth Gangadhar

Jiaxing Yang

Actors in the Application:

1. Instructor/ Student/ Admin
2. Administrator

1)

Use Case	User Registration
Primary Actor	Instructor
Goal in Context	User Registration to the application so that, once approved by admin, the user is authorized to run plagiarism detection on the submitted projects.
Preconditions	User must be an instructor or a teaching assistant.
Triggers	User wants to create an account to access the application.
Scenario	<ol style="list-style-type: none">1. User wants to create an account.2. System prompts the user with the registration menu that includes data fields to be filled by the user3. User enters the data and clicks on register button4. An approval request is sent to the admin to verify and grant access to user.
Exceptions	User is not an instructor or TA and hence will not be assigned credentials to access the application.

2)

Use Case	User Approval
Primary Actor	Administrator
Goal in Context	Admin needs to validate the user in order for the user to access the application.
Preconditions	A user registration request should already be sent to the admin.
Triggers	A new user has registered and is awaiting admin approval
Scenario	<ol style="list-style-type: none">1. Admin gets a account creation request from the instructor.2. Admin validates the user.3. Once the user is approved, an email is sent to user that his/her account request is approved and can now login to the application.
Exceptions	User is not an instructor or TA and hence will not be approved by the admin.

3)

Use Case	User Login
Primary Actor	Administrator/ Instructor
Goal in Context	User validation on the basis of username and password
Preconditions	A user should have a valid account in the application
Triggers	User wants to access the application
Scenario	1)User wants to login to his/her account. 2)System prompts the user with the login menu that includes data fields to be filled by the user 3)User enters the data and clicks on login button 4)On valid credentials, the user is directed to the landing page of application
Exceptions	User enters invalid credentials and the error message will be prompted so that the user can again re-enter the correct credentials.

4)

Use Case	Forget Password
Primary Actor	Administrator/ Instructor
Goal in Context	Regenerate Users password
Preconditions	A user should have a valid account in the application
Triggers	User forgets the password credentials

Scenario	1)User forgets password. 2)User clicks on Forget Password button 3)User enters the registered email id 4)Email is sent to user with new password.
Exceptions	User enters an Email id that is not registered. The system then prompts the error message.

5)

Use Case	Select comparison option
Primary Actor	Instructor
Goal in Context	User manually specify the option of what are the things he wants to compare
Preconditions	The user should know what type of input he is trying to provide
Triggers	The user wants to provide input to the system by uploading files or folders or the text.
Scenario	<ol style="list-style-type: none"> 1. The user will be presented 4 options: <ol style="list-style-type: none"> a. Compare two files b. Compare 2 folders c. Text comparison d. Full directory comparison 2. User select on of the above option based on the input it has 3. User will be redirected to the specific page based on the selected option
Exceptions	User is unaware of the of the text, files or folders to be compared and selected the wrong option

6)

Use Case	Manual comparison of files / Text comparator
Primary Actor	Instructor
Goal in Context	User can manually compare two text files. And see difference bases on highlights done in the text
Preconditions	User has successfully logged in into the system and must have at least two versions of source code.
Triggers	The user decides to detect plagiarism by comparing two files
Scenario	<ol style="list-style-type: none">1. After login in, the instructor clicks the 'Text Comparison' button2. The system displays two text areas where instructor can paste the text3. Instructor enter the student name and the enrollment number of the student under its source.4. Repeat step 2 and 3 to paste the second source input5. Hit compare source button
Exceptions	<ol style="list-style-type: none">1. The user paste the wrong file2. The user doesn't provide two versions

7)

Use Case	Compare two files
Primary Actor	Instructor
Goal in Context	To specify which file to be used for plagiarism detection
Preconditions	User has successfully logged in into the system and must have two versions of source code in his local storage. The source code must be in a file.
Triggers	The user decides to detect plagiarism by comparing two files
Scenario	<ol style="list-style-type: none">1. After login in, the instructor clicks the 'Compare two files' button2. The system displays a file selector that allows instructor to upload file3. The instructor selects file4. The file selector window closes. The file name displays in the input box.5. Instructor enter the student name and the enrollment number of the student under it's file.6. Repeat step 2~5 to upload second file7. Hit compare files button
Exceptions	<ol style="list-style-type: none">1. The user doesn't have the source files2. The user selects the wrong file3. The user doesn't provide two versions

8)

Use Case	Compare two folders
Primary Actor	Instructor
Goal in Context	To specify which folders to be used for plagiarism detection
Preconditions	User has successfully logged in into the system and must have two versions of source code in his local storage. The source code must be in a folder.
Triggers	The user decides to detect plagiarism by comparing two folders
Scenario	<ol style="list-style-type: none">1. After login in, the instructor clicks the 'Compare two folders' button2. The system displays a folder selector that allows instructor to upload a folder3. The instructor selects a folder4. The folder selector window closes. The folder name displays in the input box.5. Instructor enter the student name and the enrollment number of the student6. Repeat step 2~5 to upload second folder7. Hit compare folder button
Exceptions	<ol style="list-style-type: none">1. The user doesn't have the source folders.2. The user selects the wrong folder.3. The user doesn't provide two versions.

9)

Use Case	Compare Full Directory
Primary Actor	Instructor
Goal in Context	To specify which folders to be used for plagiarism detection
Preconditions	User has successfully logged in into the system. The source code must be in a folder.
Triggers	The user decides to detect plagiarism by comparing all the files and folders with each other.
Scenario	<ol style="list-style-type: none">1. After login in, the instructor clicks the 'Compare Full Directory' button2. The system displays a folder selector that allows instructor to upload folder3. The instructor selects a folder4. The folder selector window closes. The folder name displays in the input box5. Hit 'Compare Directory' button
Exceptions	<ol style="list-style-type: none">1. The user doesn't have the source folders2. The user selects the wrong folder

10)

Use Case	Register Students
Primary Actor	Instructor
Goal in Context	Instructor will be responsible for registering students of their class into this system using a csv file
Preconditions	Instructor is successfully logged in into the system
Triggers	Student is not present in the system whose files and folders are going to be compared
Scenario	<ol style="list-style-type: none">1. Instructor must be successfully logged in into the system2. Click on the 'Register Student' button3. Upload the csv file. The csv file will have 3 columns [Last Name, First Name and Enrollment Number]
Exceptions	CSV file is not in proper format

11)

Use Case	View report
Primary Actor	User
Goal in Context	User views the report to see the details of plagiarism detection.
Preconditions	User must have clicked the 'Compare' button. The server must successfully produce a report.
Triggers	The user has viewed the score and text indication. He wants further details of the score and sees which parts could be the evidence of plagiarism.

Scenario	<ol style="list-style-type: none"> 1. User clicks the 'View report' button. 2. The system displays two lists of file names side by side. The files are sorted by plagiarism score, with highest score at top. 3. User clicks a file from either list. 4. The corresponding file in the other list is highlighted. 5. The system displays the content of the two files in two text areas side by side. 6. Similar code blocks are marked with same color. 7. The user clicks a colored code block in a file. The other file scrolls to the same color block, which indicates detected similar code. 8. The user checks other parts of the code and other files until satisfied.
Exceptions	<ol style="list-style-type: none"> 1. The user accidentally closes the window before he views the report. 2. The reports are not displayed properly. 3. The system doesn't detect any plagiarism. It should inform the user rather than displays an empty report.

12)

Use Case	Email report
Primary Actor	User
Goal in Context	User emails the generated report to specified email address
Preconditions	The report has already been generated on the system.
Triggers	The user has viewed he report on its machine and wants to share the report generated with someone else.
Scenario	<ol style="list-style-type: none"> 1. The user has generated by the detailed report 2. Click on share report option and specify the email address of the person, he wants to share the report. 3. Click on send

Exceptions	<ol style="list-style-type: none"> 1. Report has not been generated and user has clicked on the send button. 2. Email field is empty and user has clicked on the send button. 3. Email is not a valid email address.
-------------------	---

13)

Use Case	Download report
Primary Actor	User
Goal in Context	User downloads and save the generated report in his local repository
Preconditions	The report has already been generated on the system.
Triggers	The user has viewed the report on its machine and wants to save the report in its local repository
Scenario	<ol style="list-style-type: none"> 1. The user has generated by the detailed report 2. Click on download report option. A screenshot of the report will be downloaded on the user's system.
Exceptions	Report has not been generated and user has clicked on the download button.

14)

Use Case	Checks history of comparison analysis
Primary Actor	Administrator
Goal in Context	Admin can check the history of all the comparisons made by various instructors
Preconditions	<ol style="list-style-type: none">1. Admin must be successfully logged in into the system2. The report has already been generated in the system
Triggers	Admin wants to look at all the plagiarisms detected in various comparison made by all the instructors
Scenario	<ol style="list-style-type: none">1. Admin logged into the system2. Click on 'Generate All Plagiarism Report' button
Exceptions	No comparison has ever been made by any instructor

15)

Use Case	Manage Users
Primary Actor	Administrator
Goal in Context	Admin will be responsible for managing all the users in the systems
Preconditions	<ol style="list-style-type: none">1. Admin must be successfully logged in into the system2. There must be at least one user in the system
Triggers	Instructor is no more associated with his organization Student is no more associated with his organization
Scenario	<ol style="list-style-type: none">1. Admin logged into the system2. Click on 'Manage Users' button. It will get a list of all the users present in this system with their roles (Admin, Instructor or Student)3. It can delete any user from the database
Exceptions	No user is present in the system with the role as Student or Instructor

Use Cases

Feature: All user functionalities

No.	Use Case	Completion	Summary
1	User registration	Done	User chooses a role (student, professor and admin) and creates an account.
2	User login	Done	User inputs the email and password to login.
3	Update profile	Done	User can update his information.
4	User logout	Done	User logout the system.

Feature: Professor functionalities – course and assignment

No.	Use Case	Completion	Summary
5	Create course	Done	Professor can create courses.
6	Edit course	Done	Professor can update and delete his courses
7	View course	Done	Professor views a list of his courses.
8	View enrolled students	Done	Professor views the list of students enrolled in each course.
9	Create assignment	Done	Professor can create assignments for courses.
10	Edit assignment	Done	Professor can update and delete assignments
11	View assignment	Done	Professor views a list of assignments on selecting a course.
12	View submissions	Done	Professor views a list of student submissions to an assignment

Feature: Professor functionalities – compare

No.	Use Case	Completion	Summary
13	Change threshold	Done	Professor can change threshold which indicates how tolerant to plagiarism.
14	Compare all submissions	Done	Professor compares all submissions of an assignment to see if there is any plagiarism
15	Compare selected submissions	Done	Professor selects some submissions of an assignment to see if there is any plagiarism
16	Text comparison	Done	Professor inputs two python program for comparison
17	File comparison	Done	Professor uploads two python files for comparison
18	Folder comparison	Done	Professor uploads two folders (python project) for comparison
19	View comparison result	Done	Professor views the result of students and their submitted files
20	View file details	Done	Professor views the similarity score and matched lines of two files.
21	View comparison history	Done	Professor views the history of comparisons he performed

Feature: Student functionalities

No.	Use Case	Completion	Summary
22	Enroll course	Done	Student enrolls in a course

23	Drop course	Done	Student drops an enrolled course
24	View course	Done	Student views a list of enrolled courses
25	View assignment	Done	Student views assignments of a course
26	Submit assignment	Done	Student upload file or folder for an assignment

Feature: Admin functionalities

No.	Use Case	Completion	Summary
27	Approve user	Done	Admin approves a user's registration request
28	Reject user	Done	Admin rejects a user's registration request
29	Manage users	Done	Admin can view, create, update and delete users
30	Manage semester	Done	Admin can view, create, update and delete semesters