List of use cases for the Plagiarism Detection System

1. Register:

Use Case:	Register
Primary Actor:	User could be a Student or a Professor
Goal in Context	For a User to register to use the application
Preconditions	1. Access to Internet 3. A valid NEU id and a valid Email ID. 3. The actor's NEU id and Email ID is not registered already.
Trigger	The user wants to use the Plagiarism system.
Scenario	1. User clicks on "Register" link button on home page 2. User is redirected to "Register" page 3. User enters NEU id 3. User enters his full name 4. User enters Email ID 5. User enters a Password 6. User selects his role, either as a student or Professor. 7. User clicks on the "Submit" button 8. System checks if the NEU Id and Email ID is valid and not already in use 9. System registers the User as a TA or Professor.
Exceptions	<u>1. The user NEU id is already registered</u> • System generates error message prompting the user to Login or register with different Email ID. <u>2. Invalid Email ID is entered.</u> • System generate error message prompting the user to enter a valid Email ID.

2. Login:

Use Case:	Login
Primary Actor:	Student or Professor
Goal in Context	For the Student or Professor to Login into the system.
Preconditions	1. Access to Internet. 2. Existing registered NEU id.
Trigger	The user wants to use the Plagiarism system.
Scenario	1. User clicks on "Login" button on home page 2. User is redirected to "Login" page 3. User enters his registered NEU ID 4. User enters a Password 5. User clicks on the "Login" button 6. System checks if the NEU ID is valid and not already in use 7. System registers the User as a student or Professor.
Exceptions	1. The user is not registered • System generates an error message prompting the user to Sign - up. 2. Invalid NEU ID is entered. • System generate s an error message prompting the user to enter a valid NEU ID. 3. Invalid Password entered. • System generates an error message prompting the user to enter the right password.

ADMIN USE CASES:

1. Create User

Use Case:	Create_User
Primary Actor:	Admin
Goal in Context	To create users.
Preconditions	1. Access to Internet. 2. Details of required user to create
Trigger	The user cannot register himself or is another admin user
Scenario	1. Admin logs into the system with admin credentials. 2. Admin clicks on "New User" button. 3. A new page with a register form is opened. 4. The admin has to enter all the user details and the role of the user 5. Click on submit to create user
Exceptions	1. Database connectivity issues. • The admin is prompted to re - try creating the Users.

2. Udpate User

Use Case:	Update_User
Primary Actor:	Admin
Goal in Context	To update existing user’s details
Preconditions	1. Access to Internet. 2. The user to update should already be registered
Trigger	The user cannot register himself or is another admin user
Scenario	1. Admin logs into the system with admin credentials. 2. Admin clicks on “Update User” button. 3. A new page with a list of all users is opened. 4.The admin then click “edit” to edit details of a specific user 5. The admin has to enter the user details to update for the user 6. Click on “Update” to create user
Exceptions	1. Database connectivity issues. • The admin is prompted to re - try updating the Users.

3. Delete User

Use Case:	Delete_User
Primary Actor:	Admin
Goal in Context	To delete existing user’s details
Preconditions	1. Access to Internet. 2. At least one user should be registered in the system.
Trigger	The user is inactive for over a year; The user was a malicious user.
Scenario	1. Admin logs into the system with admin credentials. 2. Admin clicks on “Delete User” button. 3. A new page with a list of all users is opened. 4. The admin has to select users to be deleted. 5. Click on “Delete” to permanently delete the users.
Exceptions	1. Database connectivity issues. • The admin is prompted to re - try deleting the Users.

4. Grant Professor Role

Use Case:	Grant_User_Role
Primary Actor:	Admin
Goal in Context	To update existing user’s role from “professor_temp” to “professor”
Preconditions	1. Access to Internet. 2. The professor should be legit professor and admin is aware of his identity

Trigger	The user cannot register himself as professor due to false identification threat
Scenario	1. Admin logs into the system with admin credentials. 2. Admin clicks on “Grant Professor Role” button. 3. A new page with a list of all users with role “Professor_temp” is opened. 4.The admin then click “Grant Professor Role” to change role of a specific user and the role of the professor is updated.
Exceptions	1<u>. Database connectivity issues.</u> • The admin is prompted to re - try updating the Users role again.

STUDENT USE CASES:

1. Register for course

Use Case:	Register_For_Course
Primary Actor:	Student
Goal in Context	To register for a course created by a professor
Preconditions	1. Access to Internet. 2. Student should have logged in
Trigger	The student wants to register for a course
Scenario	1. Student logs in to the system 2. Student clicks on the Register Course button on the NavBar 3. A new page is opened with the list of courses available 4.The user clicks on the Register button across a course 5. The students also gets an email that he registered for the course
Exceptions	1. Database connectivity issues. • The student is alerted with an error and prompted to try again.

2. Drop a Course

Use Case:	Drop_Course
Primary Actor:	Student
Goal in Context	To a a course created by a professor
Preconditions	1. Access to Internet. 2. Student should have logged in 3. Student should have registered for at least one course
Trigger	The student wants to drop a course
Scenario	1. Student logs in to the system 2. Student clicks on the Drop Course button on the NavBar 3. A new page is opened with the list of courses he is registered for. 4.The user clicks on the Drop button across a course 5. The course no longer appears as part of his registered courses 6. Student also gets an email that the course has been dropped
Exceptions	1. Database connectivity issues. • The student is alerted with an error and prompted to try again.

3. Submit Assignment though github

Use Case:	Submit_Assignment_Github
Primary Actor:	Student
Goal in Context	To submit an assignment
Preconditions	1. Access to Internet. 2. Student should have logged in 3. Student should have registered to at least one course
Trigger	The student wants to submit an assignment for a course
Scenario	1. Student logs in to the system 2. Student clicks on the Submit Assignment button on the NavBar 3. A new page is opened with the list of courses available 4.The user clicks on the View Assignment button across a course 5. An input box pops up, where the student can paste his github link to the assignment 6. Student clicks on the Submit button to make a submission.
Exceptions	1. Database connectivity issues. • The student is alerted with an error and prompted to try again.

4. Submit Assignment through zip files

Use Case:	Submit_Assignment_Folder
Primary Actor:	Student
Goal in Context	To submit an assignment
Preconditions	1. Access to Internet. 2. Student should have logged in 3. Student should have registered to at least one course
Trigger	The student wants to submit an assignment for a course
Scenario	1. Student logs in to the system 2. Student clicks on the Submit Assignment button on the NavBar 3. A new page is opened with the list of courses available 4.The user clicks on the View Assignment button across a course 5. An input box pops up, where the student can drag and drop his zip file of the assignment 6. Student clicks on the Submit button to make a submission.
Exceptions	1. Database connectivity issues. • The student is alerted with an error and prompted to try again.

PROFESSOR USE CASES:

1. Create new courses

Use Case:	Create_new_courses
Primary Actor:	Professor
Goal in Context	To create a new course
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin
Trigger	The professor wants to create a new course
Scenario	1. Professor logs in to the system 2. Professor clicks on the Add New Course button on the NavBar 3. A new page is opened with a Course creation form 4.The professor enters a valid course abbreviation 5. The professor enters a valid course location 6. The professor enters a valid course name 7. The professor enters a valid course term 8. He can select the checkbox if he's adding a second section of the course 9. Professor clicks on the Submit button to create a new course
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

2. View courses

Use Case:	View_Courses
Primary Actor:	Professor
Goal in Context	To view courses
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin
Trigger	The professor wants to view all courses
Scenario	1. Professor logs in to the system 2. Professor clicks on the Courses Page button on the NavBar 3. A new page is opened with all the list of courses created by him
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

3. Delete a course

Use Case:	Delete_courses
Primary Actor:	Professor
Goal in Context	To delete a course
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4. There should be at least one course in the system
Trigger	The professor wants to delete a course
Scenario	1. Professor logs in to the system 2. Professor clicks on the Courses Page button on the NavBar 3. A new page is opened with a list of all available courses 4.The professor clicks on the Delete button across the course 5. the course is no longer part of the system and is deleted from the courses list too.
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

4. Update a course

Use Case:	Update_Course
Primary Actor:	Professor
Goal in Context	To update a course
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4. There should be at least one course in the system
Trigger	The professor wants to update a course
Scenario	1. Professor logs in to the system 2. Professor clicks on the Add New Course button on the NavBar 3. A list of all available courses is displayed 4. user clicks on the Edit button across a course 5. A new form pops up below with the current course details and the user can edit those details 6.The professor enters a valid course abbreviation 7. The professor enters a valid course location 8. The professor enters a valid course name 9. The professor enters a valid course term 10. Professor clicks on the Submit button to update a course
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

5. Create Assignments

Use Case:	Create_New_Assignment
Primary Actor:	Professor
Goal in Context	To create a new assignment
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system
Trigger	The professor wants to create a new assignment
Scenario	1. Professor logs in to the system 2. Professor clicks on the Add New Assignment button on the NavBar 3. A list of courses is added to the page. 4. User clicks on “Add Assignment” button across a course 5. The page is appended with an Assignment creation form 6.The professor enters a valid assignment number 7. The professor enters a valid assignment name 8. The professor enters a valid submission date 9. Professor clicks on the Submit button to create a new assignment for the course
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

6. View Assignments

Use Case:	View_Assignments
Primary Actor:	Professor
Goal in Context	To view assignments
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system
Trigger	The professor wants to view assignments created for a course
Scenario	1. Professor logs in to the system 2. Professor clicks on the "Assignments Page" button on the NavBar 3. A list of courses is added to the page. 4. User clicks on "View Assignment" button across a course 5. The page is appended with a list of Assignments created for the course
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

7. Delete Assignments

Use Case:	Delete_Assignments
Primary Actor:	Professor
Goal in Context	To delete assignments
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system 5. There should be at least one assignment created
Trigger	The professor wants to delete assignments created for a course
Scenario	1. Professor logs in to the system 2. Professor clicks on the "Assignments Page" button on the NavBar 3. A list of courses is added to the page. 4. User clicks on "View Assignment" button across a course 5. The page is appended with a list of Assignments created for the course 6. The professor clicks on the "Delete" button across the assignment
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

8. <u>Update Assignments</u>

Use Case:	Update_Assignments
Primary Actor:	Professor
Goal in Context	To update assignments
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system 5. There should be at least one assignment created
Trigger	The professor wants to update assignments created for a course
Scenario	1. Professor logs in to the system 2. Professor clicks on the “Assignments Page” button on the NavBar 3. A list of courses is added to the page. 4. User clicks on “View Assignment” button across a course 5. The page is appended with a list of Assignments created for the course 6. The professor clicks on the “Edit” button across the assignment 7. The assignments form is appended at the bottom of the screen and professor can update changes to the assignment The professor enters a valid assignment number 8. The professor enters a valid assignment name 9. The professor enters a valid submission date 10. Professor clicks on the Submit button to update the assignment for the course
Exceptions	1<u>. Database connectivity issues.</u> • The professor is alerted with an error and prompted to try again.

9. Standard comparison

Use Case:	Standard_Comparision
Primary Actor:	Professor
Goal in Context	To check plagiarism for a course
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system 5. There should be at least one assignment created 6. At least two students should have made submissions
Trigger	The professor wants to generate plagiarism reports for an assignment
Scenario	1. Professor logs in to the system 2. Professor clicks on the "Generate Reports" button on the NavBar 3. A list of courses is added to the page. 4. User clicks on "View Assignment" button across a course 5. The page is appended with a list of Assignments created for the course 6. The professor clicks on the "Check Plagiarism" button across the assignment 7. Professor enters a valid threshold value. 8. Selects the language of comparision 9. Clicks on the button "Fetch Submissions" 10. Clicks on the button "Generate Reports" and a list of reports with the percentage match is generated. 11. The professor can click on the s3 link to download the reports
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

10. Comparison across semesters

Use Case:	Semester_Comparision
Primary Actor:	Professor
Goal in Context	To check plagiarism across semesters
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system 5. There should be at least one assignment created 6. At least two students should have made submissions 7. The previous semester courses and their assignments and student submissions should also be part of the system
Trigger	The professor wants to generate plagiarism reports for an assignment across previous semesters
Scenario	1. Professor logs in to the system 2. Professor clicks on the “Generate Reports” button on the NavBar 3. A list of courses is added to the page. 4. User clicks on “View Assignment” button across a course 5. The page is appended with a list of Assignments created for the course 6. The professor clicks on the “Check Plagiarism” button across the assignment 7. Professor enters a valid threshold value. 8. Selects the language of comparision 9. Clicks on the button “Fetch Terms” 10. Clicks on the button “Generate Reports” and a list of reports with the percentage match is generated. 11. The professor can click on the s3 link to download the reports
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

11. Comparison across sections

Use Case:	Comparision_Across_Sections
Primary Actor:	Professor
Goal in Context	To check plagiarism across sections
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system and should have at least two sections 5. There should be at least one assignment created 6. At least two students should have made submissions from each section
Trigger	The professor wants to generate plagiarism reports for an assignment across different sections
Scenario	1. Professor logs in to the system 2. Professor clicks on the "Generate Reports" button on the NavBar 3. A list of courses is added to the page. 4. User clicks on "View Assignment" button across a course 5. The page is appended with a list of Assignments created for the course 6. The professor clicks on the "Check Plagiarism" button across the assignment 7. Professor enters a valid threshold value. 8. Selects the language of comparision 9. Clicks on the button "Fetch Sections" 10. Clicks on the button "Generate Reports" and a list of reports with the percentage match is generated. 11. The professor can click on the s3 link to download the reports
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.

12. Email Students

Use Case:	Email_Students
Primary Actor:	Professor
Goal in Context	To notify the students they have been caught for plagiarism
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system 5. There should be at least one assignment created 6. At least two students should have made submissions 7. There should be at least one plagiarism check
Trigger	The professor wants to notify the students they have been caught for plagiarism
Scenario	1. Professor logs in to the system 2. Professor clicks on the "Generate Reports" button on the NavBar 3. A list of courses is added to the page. 4. User clicks on "View Assignment" button across a course 5. The page is appended with a list of Assignments created for the course 6. The professor clicks on the "Check Plagiarism" button across the assignment 7. Professor enters a valid threshold value. 8. Selects the language of comparision 9. Clicks on the button "Fetch Submissions" 10. Clicks on the button "Generate Reports" and a list of reports with the percentage match is generated. 11. The professor can click on the "Send" button across Email Students section to notify the students 12. A pop up occurs on successfully sending a mail
Exceptions	1. Database connectivity issues.

	• The professor is alerted with an error and prompted to try again.

13. Email Reports

Use Case:	Email_Reports
Primary Actor:	Professor
Goal in Context	To send a copy of the report to himself
Preconditions	1. Access to Internet. 2. Professor should have logged in 3. Professor should be a legit user verified by the admin 4.There should be at least one course registered in the system 5. There should be at least one assignment created 6. At least two students should have made submissions 7. There should be at least one plagiarism check
Trigger	The professor wants to keep a copy of the report and mail it to himself
Scenario	1. Professor logs in to the system 2. Professor clicks on the "Generate Reports" button on the NavBar 3. A list of courses is added to the page. 4. User clicks on "View Assignment" button across a course 5. The page is appended with a list of Assignments created for the course 6. The professor clicks on the "Check Plagiarism" button across the assignment 7. Professor enters a valid threshold value. 8. Selects the language of comparision 9. Clicks on the button "Fetch Submissions" 10. Clicks on the button "Generate Reports" and a list of reports with the percentage match is generated.

	11. The professor can click on the “Send” button across Email Reports section to notify the students 12. A pop up occurs on successfully sending a mail 13. The professor should have received a mail
Exceptions	1. Database connectivity issues. • The professor is alerted with an error and prompted to try again.