USE CASES

Use Case:	Login
Primary Actor:	Instructor
Goal in Context:	To differentiate among the actors and grant access to the web application to detect plagiarism between assignments submitted by the students of a course.
Preconditions:	1. The user should have successfully completed the signup process 2. The user should not be already logged in
Trigger:	User needs access to the plagiarism detector
Scenario:	1. The user visits the URL on which the web application is hosted 2. The user enters user ID and password 3. The user clicks on the login button
Exceptions:	Invalid User ID - password combination, Remedy: User has to re-enter their credentials.
Priority:	High priority, it is a basic functionality which must be implemented
Channel to actor:	Browser (Chrome)
Secondary Actor:	Administrator
Channels to Secondary Actors:	Browser (Chrome)

Use Case:	Sign up
Primary Actor:	Instructor
Goal in Context:	To enable user to login into the web application and to link the user's github account
Preconditions:	The user must have a valid github account and must provide correct username and password for the same.
Trigger:	The user wants to register for a new account to access the web application to detect plagiarism
Scenario:	1. The user clicks the signup button 2. The user enters username, password, github credentials 3. The system validates the github credentials, creates an account and displays a success message that account has been successfully created
Exceptions:	1. Username already exists, Remedy: Choose a different username 2. Invalid Credentials for Github, Remedy: Provide correct github credentials 3. Password and Confirm password don't match, Remedy: Enter the same password
Priority:	High priority, it is a basic functionality which must be implemented
Channel to actor:	Browser (Chrome)
Secondary Actor:	Administrator
Channels to Secondary Actors:	Browser (Chrome)

Use Case:	Add course
Primary Actor:	Instructor
Goal in Context:	To detect plagiarism for the assignments in a new course
Preconditions:	1. The user is logged in to the web application 2. A git repository for the course must exist or be created prior to this step 3. The github credentials are valid and have not been changed after the sign-up process
Trigger:	The user wants to detect plagiarism in the assignments of a particular course
Scenario:	1. The user logs in to the web application and clicks the new course button on the home page 2. The user adds the github repository link and submits to add a new course 3. If the github repository link has been linked successfully, the new course is added on home page else error message is displayed
Exceptions:	1. Invalid github repository link, Remedy: Provide the correct github repository link 2. Invalid github credentials, Remedy: Provide valid github credentials
Priority:	High priority, it is a basic functionality which must be implemented
Channel to actor:	Browser (Chrome)

Use Case:	View Assignments
Primary Actor:	Instructor
Goal in Context:	To generate or view plagiarism results for one of the assignments of a particular course
Preconditions:	1. The user is logged in to the web application 2. Course related to that assignment should be added
Trigger:	The instructor decides to view the list of assignments or add a new assignment introduced in the course
Scenario:	1. Instructor logs into the web application 2. Instructor clicks on a specific course 3. Instructor clicks on view assignment for that particular course
Exceptions:	Access denied for fetching the repository, Remedy: Grant access to the repository
Priority:	High priority, it is a basic functionality which must be implemented
Channel to actor:	Browser (Chrome)

Use Case:	Add Assignment
Primary Actor:	Instructor
Goal in Context:	To detect plagiarism on a particular assignment of a course.
Preconditions:	1. The user is logged in to the web application 2. Course related to that assignment should be added.
Trigger:	The instructor decides to add a new assignment and check plagiarism for the same
Scenario:	1. Instructor logs onto the web application 2. Instructor clicks on the particular course 3. Instructor clicks on add assignment for that particular course. 4. The assignment has been added and is displayed along with the list of assignments of the course.
Exceptions:	1. The assignment folder does not exist on GitHub course repository, Remedy: Enter the correct assignment details 2. Assignment already exists, Remedy: Prompt the user to not add the assignment as it already exists
Priority:	High priority, it is a basic functionality which must be implemented
Channel to actor:	Browser (Chrome)

Use Case:	View/Generate results
Primary Actor:	Instructor
Goal in Context:	To obtain the plagiarism result for an assignment of a particular course
Preconditions:	1. The user is logged in to the web application 2. Course should be added 3. An assignment belonging to that particular course should be added
Trigger:	Instructor decides to find the students involved in plagiarism for the assignment of that course.
Scenario:	1. Instructor logs onto the web application 2. Instructor clicks on the particular course 3. Instructor clicks on add assignment for that particular course 4. Instructor clicks on view result button to view an already generated result or clicks on the generate result button to obtain plagiarism results for the particular assignment chosen
Exceptions:	Unable to clone from the GitHub repository, Remedy: Check for access privileges and if the correct URL is entered
Priority:	High priority, it is a functionality which must be implemented
Channel to actor:	Browser (Chrome)

Use Case:	Cancel result generation
Primary Actor:	Instructor
Goal in Context:	To stop the result generation while running the plagiarism detector on the assignment
Preconditions:	1. The instructor is logged into the web application 2. The result generation must be in progress
Trigger:	1. A wrong course or assignment has been uploaded by the instructor for detecting plagiarism. 2. The instructor decides to run the plagiarism detector on the assignments at his convenience.
Scenario:	1. Instructor logs onto the web application 2. Instructor goes to the particular course 3. Instructor adds assignment for that particular course 4. Instructor clicks on generate result 5. When in progress, instructor clicks on cancel button
Exceptions:	If the user lands to the same page again after the results have been successfully created, he would not be able to cancel the task, Remedy: User should get back to the assignment page.
Priority:	Moderate priority; it is a basic functionality which might be implemented when required
Channel to actor:	Browser (Chrome)

Use Case:	View detailed results
Primary Actor:	Instructor
Goal in Context:	To analyze the plagiarized content and view the details of the students involved in plagiarism.
Preconditions:	1. The instructor is logged into the web application 2. The plagiarism results has been successfully generated
Trigger:	The instructor is interested in comparing the files and getting more details regarding a particular result
Scenario:	1. Instructor logs onto the web application 2. Instructor clicks on the particular course 3. Instructor adds assignment for that particular course 4. Instructor clicks on generate result 5. Instructor clicks on one of the generated results, to view the detailed result
Exceptions:	Results have not been previously generated, Remedy: Refresh page for the view results button to appear
Priority:	High priority; it is a functionality which is to be implemented
Channel to actor:	Browser (Chrome)

Use Case:	Save Report to Desktop
Primary Actor:	Instructor
Goal in Context:	To save the detailed result for emailing it to the concerned students and higher authorities.
Preconditions:	1. The instructor is logged into the web application 2. The plagiarism results has been generated for that assignment
Trigger:	1. The instructor finds the assignment to be plagiarized upon reviewal and decides to take further action. 2. The instructor decides to email the result to the students and higher authorities.
Scenario:	1. Instructor logs onto the web application 2. Instructor goes to the particular course 3. Instructor adds assignment for that particular course 4. Instructor clicks on generate result 5. Instructor clicks on one of the generated results, to view the detailed result 6. Instructor clicks on the export button
Exceptions:	No results to save, Remedy: User needs to generate the result
Priority:	Moderate priority, might be implemented after basic functions
Channel to actor:	Browser (Chrome)

Use Case:	Delete account
Primary Actor:	Administrator
Goal in Context:	To remove an invalid user account
Preconditions:	The user is logged in as the administrator into the system.
Trigger:	The instructor is no more involved in a course.
Scenario:	1. Administrator logs onto the web application 2. Administrator clicks on the account name to be deleted 3. Administrator clicks on delete account button 4. Administrator confirms by clicking the confirm button on the popup
Exceptions:	Account has already been deleted, Remedy: Refresh the page
Priority:	Moderate priority; it is a basic functionality which might be implemented when required
Channel to actor:	Browser (Chrome)