Grouped by features:

User register, login, update, delete, logout:

Use Case:	Registering as a user in the plagiarism detection web application.
Primary Actor:	Admin, Professor, Teaching Assistant, Student
Goal in Context:	To register as a user for the plagiarism detection web application.
Preconditions:	The user should be connected to the web.
Trigger:	The user chooses to access the plagiarism detection web application and add themselves as a user.
Scenario:	1. The user has opened the plagiarism detection web application on their browser.
Priority:	High

Use Case:	Logging in to plagiarism detection web application.
Primary Actor:	Admin, Professor, Teaching Assistant, Student
Goal in Context:	To log in to the plagiarism detection web application.
Preconditions:	The user should already be a registered and access the software using his/her user ID
Trigger:	User tries to log into the plagiarism detector software web application.
Scenario:	1. The user enters his/her user ID. 2. The user enters a password.
Priority:	High

Use Case:	Logging in to plagiarism detection web application using a Google account.
Primary Actor:	Admin, Professor, Teaching Assistant, Student
Goal in Context:	To log in to the plagiarism detection web application.
Preconditions:	The user should have a Google account.
Trigger:	User tries to log into the plagiarism detector software web application.
Scenario:	1. The user enters the credentials for their Google Account
Priority:	Medium

Use Case:	Updating the user data related to the user currently logged in.
Primary Actor:	Admin, Professor, Teaching Assistant, Student.
Goal in Context:	To update one or many of these values related to the logged in user – username, first name, last name, phone number, password.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user chooses to update the values.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user changes any of the aforementioned fields.
Priority:	Medium

Use Case:	Deleting as a user from the plagiarism detection web application.
Primary Actor:	Admin, Professor, Teaching Assistant, Student.
Goal in Context:	To removing themselves as a user from the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user chooses to delete himself/herself.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses to delete himself/herself.
Priority:	Medium

Use Case:	Logging out of the plagiarism detection web application.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To log out of the plagiarism detection web application.

Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user chooses to log out and close the plagiarism detection web application.
Scenario:	1. The user has finished his work with the plagiarism detection web application.
Priority:	High

Assignment creation, update, upload, delete:

Use Case:	Creating an assignment for a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To add an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor. The course related to the assignment should have been created already.
Trigger:	The user decides to add an assignment to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user adds the new assignment for the course.
Priority:	Medium

Use Case:	Updating an assignment for a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To update an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor. The course and the concerned assignment should have been created already.
Trigger:	The user decides to update an assignment to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses the assignment to be updated and changes the required values.
Priority:	Medium

Use Case:	Upload the submission for an assignment to the plagiarism detection web application.
Primary Actor:	Student
Goal in Context:	To upload a submission for a course assignment.
Preconditions:	Should be logged into the plagiarism detection web application.
Trigger:	The user adds a repository to be checked for plagiarism.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user selects a course and the assignment to submit a submission for. 3. The user adds a repository by choosing one from their local file system or submitting a link to the GitHub repository.
Priority:	High

Use Case:	Deleting an assignment for a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To delete an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor. The course and the concerned assignment should have been created already.
Trigger:	The user decides to delete an assignment to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses the assignment to be deleted and deletes it.
Priority:	Medium

Use Case:	Checking all the submissions for a particular assignment in the application.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To check all the submitted homeworks for an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application. An assignment should have been created for the course in plagiarism detection web application.
Trigger:	The user chooses to check all the available submissions for an assignment of a course in the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses a course from all the available courses. 3. The user chooses the assignment from all the available assignments. 4. The user decides to check all the available submissions for the assignment.
Priority:	Medium

Detecting plagiarism:

Use Case:	Select two assignments to be compared against each other.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To compare two assignments against each other and check for plagiarism.
Preconditions:	Should be logged into the plagiarism detection web application.
Trigger:	The user selects two assignments to be checked for plagiarism.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user selects the course and the assignment for the submissions. 3. The user selects two assignments by choosing them from their local file system or drags the assignments into the web application.
Priority:	Medium

Accessing Results

Use Case:	Download the results of a plagiarism check on the assignments.
Primary Actor:	Professor
Goal in Context:	To download and save a copy of the results of an assignment that has been flagged for plagiarism.
Preconditions:	The plagiarism check should be completed and ready to be displayed.
Trigger:	User decides to download the report of a plagiarism check on the assignments.
Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	Low

Use Case:	View the results of a plagiarism check on the assignments.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To view a detailed result report of the plagiarism check where the similarities in assignments have been highlighted.
Preconditions:	The plagiarism check should be completed and ready to be displayed.
Trigger:	The user chooses to view a detailed report of the plagiarism check of the assignments.
Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	Medium

Use Case:	View the results of a plagiarism check on the assignments.
Primary Actor:	Professor, Teaching Assistant

Goal in Context:	To view a non-detailed result report of the plagiarism check where the similarity in the assignments is displayed as a percentage.
Preconditions:	The plagiarism check should be completed and ready to be displayed.
Trigger:	The user chooses to view a non-detailed report of the plagiarism check of the assignments.
Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	High

Use Case:	Share the results of a plagiarism check run on assignments.
Primary Actor:	Teaching Assistant
Goal in Context:	To report an assignment that is flagged for plagiarism to the Professor.
Preconditions:	The file should have been checked for plagiarism by the web application.
Trigger:	User decides to share the results of a plagiarism check on assignments with the Professor.
Scenario:	1. The application displays a report to the user.
Priority:	Medium

Use Case:	Share the results of a plagiarism check run on assignments.
Primary Actor:	Professor
Goal in Context:	To report an assignment that is flagged for plagiarism to OSCCR.
Preconditions:	The file should have been marked as plagiarized by the web application.
Trigger:	User decides to share the results of a plagiarism check on assignments with OSCCR.
Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	Medium

Course creation, update, join, drop, delete:

Use Case:	Creating a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To add a course in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor.
Trigger:	The user decides to add a course to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user adds the new course to the list of available courses.
Priority:	Medium

Use Case:	Updating a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To update a course in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor and the course should have been created already.
Trigger:	The user decides to update a course to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses the course to be updated and changes the required values.
Priority:	Medium

Use Case:	Joining a course in the plagiarism detection web application.

Primary Actor:	Professor, Student, TA
Goal in Context:	To join a course in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user decides to join a course to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses a course to join.
Priority:	Medium

Use Case:	Dropping a course in the plagiarism detection web application.
Primary Actor:	Professor, Student, TA
Goal in Context:	To drop a course in the plagiarism detection web application.
Preconditions:	The user should have joined the course already.
Trigger:	The user decides to drop a course to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all the courses joined. 3. The user chooses a course to drop.
Priority:	Medium

Strategy selection:

Use Case:	Selecting a strategy for detecting plagiarism in the web application.
Primary Actor:	Professor, TA
Goal in Context:	To choose a strategy for detecting plagiarism in the submitted homeworks.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user decides to run a plagiarism check on the assignments.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses a course and an assignment from the course. 3. The user chooses a strategy for comparison of the assignments.
Priority:	Medium

Semester/Section selection:

Use Case:	Selecting a section/semester for detecting plagiarism in the web application.
Primary Actor:	Professor, TA
Goal in Context:	To choose a section/semester for which the submissions should be checked for plagiarism.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user decides to run a plagiarism check on the assignments for a specific section or semester.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses a course and an assignment from the course. 3. The user chooses a section/semester for comparison of the assignments.
Priority:	Medium

Admin tasks:

Use Case:	Approving a user for the plagiarism detection web application.
Primary Actor:	Admin
Goal in Context:	To approve a registered user as a valid user for the plagiarism detection web application.
Preconditions:	The user should have registered in the plagiarism detection web application.
Trigger:	The Admin decides to approve the user for the plagiarism detection web application.
Scenario:	1. The admin logs into the plagiarism detection web application.

	2. The admin views all the available registered users. 3. The user chooses a user to approve.
Priority:	Medium

Use Case:	Removing a user for the plagiarism detection web application.
Primary Actor:	Admin
Goal in Context:	To remove a registered user as an invalid user for the plagiarism detection web application.
Preconditions:	The user should have registered in the plagiarism detection web application.
Trigger:	The Admin decides to remove the user for the plagiarism detection web application.
Scenario:	1. The admin logs into the plagiarism detection web application. 2. The admin views all the available registered users. 3. The user chooses a user to remove.
Priority:	Medium

Grouped by completion status:

Done:

Use Case:	Registering as a user in the plagiarism detection web application.
Primary Actor:	Admin, Professor, Teaching Assistant, Student
Goal in Context:	To register as a user for the plagiarism detection web application.
Preconditions:	The user should be connected to the web.
Trigger:	The user chooses to access the plagiarism detection web application and add themselves as a user.
Scenario:	1. The user has opened the plagiarism detection web application on their browser.
Priority:	High

Use Case:	Logging in to plagiarism detection web application.
Primary Actor:	Admin, Professor, Teaching Assistant, Student
Goal in Context:	To log in to the plagiarism detection web application.
Preconditions:	The user should already be a registered and access the software using his/her user ID
Trigger:	User tries to log into the plagiarism detector software web application.
Scenario:	1. The user enters his/her user ID. 2. The user enters a password.
Priority:	High

Use Case:	Logging in to plagiarism detection web application using a Google account.
Primary Actor:	Admin, Professor, Teaching Assistant, Student
Goal in Context:	To log in to the plagiarism detection web application.
Preconditions:	The user should have a Google account.
Trigger:	User tries to log into the plagiarism detector software web application.
Scenario:	1. The user enters the credentials for their Google Account
Priority:	Medium

Use Case:	Updating the user data related to the user currently logged in.
Primary Actor:	Admin, Professor, Teaching Assistant, Student.
Goal in Context:	To update one or many of these values related to the logged in user – username, first name, last name, phone number, password.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user chooses to update the values.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user changes any of the aforementioned fields.
Priority:	Medium

Use Case:	Deleting as a user from the plagiarism detection web application.
Primary Actor:	Admin, Professor, Teaching Assistant, Student.
Goal in Context:	To removing themselves as a user from the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user chooses to delete himself/herself.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses to delete himself/herself.
Priority:	Medium

Use Case:	Logging out of the plagiarism detection web application.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To log out of the plagiarism detection web application.

Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user chooses to log out and close the plagiarism detection web application.
Scenario:	1. The user has finished his work with the plagiarism detection web application.
Priority:	High

Use Case:	Creating an assignment for a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To add an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor. The course related to the assignment should have been created already.
Trigger:	The user decides to add an assignment to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user adds the new assignment for the course.
Priority:	Medium

Use Case:	Updating an assignment for a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To update an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor. The course and the concerned assignment should have been created already.
Trigger:	The user decides to update an assignment to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses the assignment to be updated and changes the required values.
Priority:	Medium

Use Case:	Upload the submission for an assignment to the plagiarism detection web application.
Primary Actor:	Student
Goal in Context:	To upload a submission for a course assignment.
Preconditions:	Should be logged into the plagiarism detection web application.
Trigger:	The user adds a repository to be checked for plagiarism.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user selects a course and the assignment to submit a submission for. 3. The user adds a repository by submitting a link to the GitHub repository.
Priority:	High

Use Case:	Deleting an assignment for a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To delete an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor. The course and the concerned assignment should have been created already.
Trigger:	The user decides to delete an assignment to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses the assignment to be deleted and deletes it.
Priority:	Medium

Use Case:	Checking all the submissions for a particular assignment in the application.

Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To check all the submitted homeworks for an assignment in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application. An assignment should have been created for the course in plagiarism detection web application.
Trigger:	The user chooses to check all the available submissions for an assignment of a course in the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses a course from all the available courses. 3. The user chooses the assignment from all the available assignments. 4. The user decides to check all the available submissions for the assignment.
Priority:	Medium

Use Case:	Select two assignments to be compared against each other.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To compare two assignments against each other and check for plagiarism.
Preconditions:	Should be logged into the plagiarism detection web application.
Trigger:	The user selects two assignments to be checked for plagiarism.
Scenario:	4. The user logs into the plagiarism detection web application. 5. The user selects the course and the assignment for the submissions. 6. The user selects two assignments by choosing them from their local file system or drags the assignments into the web application.
Priority:	Medium

Use Case:	Download the results of a plagiarism check on the assignments.
Primary Actor:	Professor
Goal in Context:	To download and save a copy of the results of an assignment that has been flagged for plagiarism.
Preconditions:	The plagiarism check should be completed and ready to be displayed.
Trigger:	User decides to download the report of a plagiarism check on the assignments.
Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	Low

Use Case:	View the results of a plagiarism check on the assignments.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To view a detailed result report of the plagiarism check where the similarities in assignments have been highlighted.
Preconditions:	The plagiarism check should be completed and ready to be displayed.
Trigger:	The user chooses to view a detailed report of the plagiarism check of the assignments.
Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	Medium

Use Case:	View the results of a plagiarism check on the assignments.
Primary Actor:	Professor, Teaching Assistant
Goal in Context:	To view a non-detailed result report of the plagiarism check where the similarity in the assignments is displayed as a percentage.
Preconditions:	The plagiarism check should be completed and ready to be displayed.
Trigger:	The user chooses to view a non-detailed report of the plagiarism check of the assignments.

Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	High

Use Case:	Share the results of a plagiarism check run on assignments.
Primary Actor:	Teaching Assistant
Goal in Context:	To report an assignment that is flagged for plagiarism to the Professor.
Preconditions:	The file should have been checked for plagiarism by the web application.
Trigger:	User decides to share the results of a plagiarism check on assignments with the Professor.
Scenario:	1. The application displays a report to the user.
Priority:	Medium

Use Case:	Creating a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To add a course in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor.
Trigger:	The user decides to add a course to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user adds the new course to the list of available courses.
Priority:	Medium

Use Case:	Updating a course in the plagiarism detection web application.
Primary Actor:	Professor
Goal in Context:	To update a course in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application as a professor and the course should have been created already.
Trigger:	The user decides to update a course to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses the course to be updated and changes the required values.
Priority:	Medium

Use Case:	Joining a course in the plagiarism detection web application.
Primary Actor:	Professor, Student, TA
Goal in Context:	To join a course in the plagiarism detection web application.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user decides to join a course to the plagiarism detection web application.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all available courses. 3. The user chooses a course to join.
Priority:	Medium

Use Case:	Dropping a course in the plagiarism detection web application.
Primary Actor:	Professor, Student, TA
Goal in Context:	To drop a course in the plagiarism detection web application.
Preconditions:	The user should have joined the course already.
Trigger:	The user decides to drop a course to the plagiarism detection web application.

Scenario:	1. The user logs into the plagiarism detection web application. 2. The user views all the courses joined. 3. The user chooses a course to drop.
Priority:	Medium

Use Case:	Selecting a strategy for detecting plagiarism in the web application.
Primary Actor:	Professor, TA
Goal in Context:	To choose a strategy for detecting plagiarism in the submitted homeworks.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user decides to run a plagiarism check on the assignments.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses a course and an assignment from the course. 3. The user chooses a strategy for comparison of the assignments.
Priority:	Medium

Use Case:	Approving a user for the plagiarism detection web application.
Primary Actor:	Admin
Goal in Context:	To approve a registered user as a valid user for the plagiarism detection web application.
Preconditions:	The user should have registered in the plagiarism detection web application.
Trigger:	The Admin decides to approve the user for the plagiarism detection web application.
Scenario:	1. The admin logs into the plagiarism detection web application. 2. The admin views all the available registered users. 3. The user chooses a user to approve.
Priority:	Medium

Use Case:	Removing a user for the plagiarism detection web application.
Primary Actor:	Admin
Goal in Context:	To remove a registered user as an invalid user for the plagiarism detection web application.
Preconditions:	The user should have registered in the plagiarism detection web application.
Trigger:	The Admin decides to remove the user for the plagiarism detection web application.
Scenario:	1. The admin logs into the plagiarism detection web application. 2. The admin views all the available registered users. 3. The user chooses a user to remove.
Priority:	Medium

In progress:

Use Case:	Upload the submission for an assignment from the local file system.
Primary Actor:	Student
Goal in Context:	To upload a submission for a course assignment.
Preconditions:	Should be logged into the plagiarism detection web application.
Trigger:	The user adds a repository to be checked for plagiarism.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user selects a course and the assignment to submit a submission for. 3. The user adds a file by choosing one from their local file system.
Priority:	High

Use Case:	Share the results of a plagiarism check run on assignments.
Primary Actor:	Professor

Goal in Context:	To report an assignment that is flagged for plagiarism to OSCCR.
Preconditions:	The file should have been marked as plagiarized by the web application.
Trigger:	User decides to share the results of a plagiarism check on assignments with OSCCR.
Scenario:	1. The user runs a plagiarism check on a programming assignment. 2. The application displays a report to the user.
Priority:	Medium

Use Case:	Selecting a section/semester for detecting plagiarism in the web application.
Primary Actor:	Professor, TA
Goal in Context:	To choose a section/semester for which the submissions should be checked for plagiarism.
Preconditions:	The user should have logged in the plagiarism detection web application.
Trigger:	The user decides to run a plagiarism check on the assignments for a specific section or semester.
Scenario:	1. The user logs into the plagiarism detection web application. 2. The user chooses a course and an assignment from the course. 3. The user chooses a section/semester for comparison of the assignments.
Priority:	Medium

Not started: