USE CASE1:

Use Case:	Administrator registers/removes users
Primary Actor:	Administrator
Goal in Context:	To register/remove user access to the plagiarism detection system
Preconditions:	Application is up and running and administrator is logged in to the system.
Trigger:	Users want to gain access to the plagiarism detection system so the administrator needs to add/delete users to the system to give them access to the plagiarism system.
Scenario:	Administrator clicks Register to add user data into the system and creates an id and password for the same given by user.
Exceptions:	User was not added /deleted successfully. In this case, ask administrator to redo the process.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface

USE CASE2:

Use Case:	Login to the plagiarism system
Primary Actor:	Instructor/TA
Goal in Context:	To let the user login to the plagiarism system
Preconditions:	Application is up and running and administrator has added instructor/TA to the system.
Trigger:	The instructor/TA wants to use the plagiarism detection system.
Scenario:	1. Instructor/TA launches the application 2. Logs in to the system
Exceptions:	Invalid (not registered) user tries to log in. In this case, ask administrator to give access to the user. See use case 1.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface
Secondary actor:	Administrator
Channels to secondary actor:	Email

USE CASE 3:

Use Case:	Upload files which need to be checked for plagiarism
Primary Actor:	Instructor/TA
Goal in Context:	To upload files that are to be checked for plagiarism.
Preconditions:	Application is up and running, administrator has added instructor/TA to the system and instructor/TA is logged in to the system.
Trigger:	The instructor/TA wants to run plagiarism detection for some files
Scenario:	1. Instructor/TA launches the application 2. Logs in to the system 3. Uploads source code files
Exceptions:	Source code files not properly uploaded. In this case, re-upload the files.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface
Open Issues:	What if uploaded source code files are not python files?

USE CASE 4:

Use Case:	Run Plagiarism check for uploaded files
Primary Actor:	Instructor/TA
Goal in Context:	To check if files uploaded are plagiarized.
Preconditions:	Application is up and running, user is logged in to the system and files are uploaded successfully.
Trigger:	The instructor/TA wants to detect if some source code files are plagiarized.
Scenario:	Instructor launches the application 2. Logs in to the system 3. Uploads source code files 4. Runs the plagiarism check.
Exceptions:	Application execution encountered some error. In this case, go back, refresh the page and try again.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface

USE CASE 5:

Use Case:	Check whether plagiarism check is completed
Primary Actor:	Instructor/TA
Goal in Context:	To check whether the plagiarism check run is completed.
Preconditions:	Application is up and running, user is logged in to the system, files are uploaded successfully and plagiarism check is running.
Trigger:	The instructor/TA wants to know if the plagiarism check run has been completed.
Scenario:	1. Instructor launches the application 2. Logs in to the system 3. Uploads source code files 4. Runs the plagiarism check. 5. Recieves a pop up when completed.
Exceptions:	Popup does not close – In this case, refresh the page.
Priority:	Optional, may be implemented
Channel to actor:	Web User Interface
Open Issues:	What to do if there's no pop-up for a long time after running the check. Where to place the pop-up?

USE CASE 6:

Use Case:	View plagiarism analysis
Primary Actor:	Instructor/TA
Goal in Context:	To view analysis of the plagiarism check.
Preconditions:	Application is up and running, user is logged in to the system, files are uploaded successfully and plagiarism check is running.
Trigger:	The instructor/TA wants to view plagiarism check analysis.
Scenario:	1. Instructor launches the application 2. Logs in to the system 3. Uploads source code files 4. Runs the plagiarism check. 5.Clicks the view analysis report button.
Exceptions:	On clicking the view analysis report button, report does not open. In this case, refresh page and try again.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface

USE CASE 7:

Use Case:	View code similarities
Primary Actor:	Instructor/TA
Goal in Context:	The instructor/TA wants to view similarities in plagiarized files.
Preconditions:	Application is up and running, user is logged in to the system, files are uploaded successfully, plagiarism check is completed and the user has chosen to view the plagiarism analysis.
Trigger:	The instructor/TA wants to view plagiarism check analysis.
Scenario:	1. Instructor/TA launches the application 2. Logs in to the system 3. Uploads source code files 4. Runs the plagiarism check. 5. Views the plagiarism analysis. 6. Clicks on the files to view similarities
Exceptions:	Similar code isn't highlighted. In this case - rerun the check.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface

USE CASE 8:

Use Case:	Show Top 10 most similar files
Primary Actor:	Instructor/TA
Goal in Context:	The instructor/TA wants to view top 10 files which are similar
Preconditions:	Application is up and running, user is logged in to the system, files are uploaded successfully, plagiarism check is completed, user has chosen to view the plagiarism analysis.
Trigger:	The instructor wants to view the top 10 most plagiarized files.
Scenario:	1. Instructor launches the application 2. Logs in to the system 3. Uploads source code files 4. Runs the plagiarism check. 5. Views the plagiarism analysis. 6. User selects to view the top 10 plagiarized files.
Exceptions:	Top 10 files are not displayed. In this case, refresh the page.
Priority:	Optional, may be implemented
Channel to actor:	Web User Interface
Open Issues:	If less than 10 files uploaded then what?

USE CASE 9:

Use Case:	Download Plagiarism report
Primary Actor:	Instructor/TA
Goal in Context:	To download the plagiarism report
Preconditions:	Application is up and running, user is logged in to the system, files are uploaded successfully, plagiarism check is completed, the user has chosen to view the plagiarism analysis and wants to download a report for the same.
Trigger:	The instructor wants to download the plagiarism report.
Scenario:	1. Instructor launches the application 2. Logs in to the system 3. Uploads source code files 4. Runs the plagiarism check. 5. Views the plagiarism analysis. 6. User clicks the download report button.
Exceptions:	Download of the report fails. In this case, please refresh the page and try again.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface

USE CASE 10:

Use Case:	Logout user from system
Primary Actor:	Instructor/TA
Goal in Context:	To let user logout from the system
Preconditions:	Application is up and running, user is logged in to the system.
Trigger:	The instructor wants to logout of the system.
Scenario:	**Scenario 1:** 1. Instructor launches the application. 2. Logs in to the system. 3. Uploads source code files. 4. Runs the plagiarism check. 5. Views the plagiarism analysis. 6. User clicks the download report button. 7. User logs out of the system. **Scenario 2** 1. Instructor launches the application. 2. Logs in to the system. 3. User logs out of the system.
Exceptions:	Logout failed. In this case, check if you are able to access the plagiarism application. If yes, please logout, else close the browser.
Priority:	Essential, must be implemented
Channel to actor:	Web User Interface