

# Use Cases

## Use Case 1

<b>Use Case</b>	Register a new user
<b>Primary Actor</b>	Professor/Student
<b>Goal in Context</b>	To register a user in the system.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• System should be fully configured.</li><li>• User should be a new user and should have never been registered with the system.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• Professor decides to look inside files for detecting plagiarism.</li><li>• Student needs to upload the assignments in the system.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• The user should navigate to the <b>CodeX</b> website.</li><li>• The user should click on Register new user button.</li><li>• The system displays specified fields to be filled by the user.</li><li>• Fields can be mandatory or optional.</li><li>• User should fill mandatory fields (First Name, Last Name, Username, Email, Password, Verify Password, User Type (Student/Professor)).</li><li>• Optional fields include Phone number.</li><li>• User should click on register button.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• UserId should be unique. If userId already exists user would be prompted to enter a new Id.</li><li>• Password should match a specific password criterion.</li><li>• Password should be a combination of alphanumeric characters and should include a special character (*, @, %, &amp;).</li><li>• Min length of the password should be at least 8 characters.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Moderate</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 4</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Frequent</li></ul>

## Use Case 2

<b>Use Case</b>	Login existing user
<b>Primary Actor</b>	Professor/Student/Admin
<b>Goal in Context</b>	Login an existing user in the system.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• System should be fully configured.</li><li>• User should be already registered in the system.</li><li>• Appropriate user Id and password should be obtained.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• Professor decides to look inside files for detecting plagiarism.</li><li>• Student decides to upload assignments.</li><li>• Admin decides to perform CRUD operations for users.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• The user should navigate to <b>CodeX</b> website.</li><li>• The user should enter appropriate username in the specified field.</li><li>• The user should enter appropriate password in the specified field.</li><li>• User should click on Login button.</li><li>• The system navigates to the user's dashboard page where the system displays all the major functions based on user type (Professor/Student/Admin).</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• Ids or passwords are incorrect or not recognized.</li><li>• Unregistered user tries to login and the system is not able to recognize the username.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Moderate</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 4</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Frequent</li></ul>

### Use Case 3

<b>Use Case</b>	Uploading Assignment file/folder
<b>Primary Actor</b>	Student
<b>Goal in Context</b>	To upload assignment files. Student is supposed to upload a given homework file.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• System should be fully configured.</li><li>• User should be a student.</li><li>• Student should be logged into the system.</li><li>• Student should be in his/her homepage.</li><li>• Homework file/folder should be ready to be uploaded.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• The student decides to upload the assignment files/folders for submission.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• The user should click on upload button.</li><li>• Student should navigate to the folder where the file is located and click on ok button in the pop up window.</li><li>• Student should click on upload button.</li><li>• File will start uploading.</li><li>• Student can add comment related to the assignment submission.</li><li>• Student should click on the submit button.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• User tries to upload a file without selecting any file.</li><li>• Uploaded files are not in specified programming language.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Moderate</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 5</li></ul>
<b>Frequency of Use</b>	<ul style="list-style-type: none"><li>• Frequent</li></ul>

#### Use Case 4

<b>Use Case</b>	Review Plagiarism Summary
<b>Primary Actor</b>	Professor
<b>Goal in Context</b>	To detect plagiarism for student submissions.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• System should be fully configured.</li><li>• User should be Professor.</li><li>• Professor should be logged into the system.</li><li>• Professor should navigate to the home page.</li><li>• Students should have submitted the assignment.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• Professor decides to see plagiarism summary.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• System displays the assignment submitted by all the students.</li><li>• System shows the check plagiarism button next to each submission.</li><li>• Professor should click on check plagiarism button for the student he wishes to check plagiarism for.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• Only one submission has been made. The system would generate an error as there are no files to compare with.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• High</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 6</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Frequent</li></ul>

### Use Case 5

<b>Use Case</b>	Generate Plagiarism Report
<b>Primary Actor</b>	Professor
<b>Goal in Context</b>	To generate detected plagiarism report.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• User should be Professor.</li><li>• Professor should be on plagiarism summary page.</li><li>• Plagiarism should have been detected.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• The user decides to view the plagiarised code for a student with another student.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• The system displays the plagiarism summary of a student for a file with all the other students' submissions.</li><li>• A 'details' button is displayed next to each student.</li><li>• Professor should click on the 'details' button.</li><li>• System displays the detailed report for two files.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• Only one submission has been made. The system would generate an error and would not navigate to the report page.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Moderate</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 7</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Infrequent</li></ul>

### Use Case 6

<b>Use Case</b>	Download Report
<b>Primary Actor</b>	Professor
<b>Goal in Context</b>	To download the detected plagiarism report.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• User should be Professor.</li><li>• Professor should be on plagiarism report page.</li><li>• A detailed plagiarism report should be displayed.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• Professor decides to download the viewed report.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• System displays the detailed report for two students.</li><li>• Professor should click on the Download Report button.</li><li>• Report is downloaded.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• If the file is not plagiarised the professor would not be able to download the report.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Moderate; to be implemented after basic functionalities.</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 7</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Infrequent</li></ul>

### Use Case 7

<b>Use Case</b>	Logout user
<b>Primary Actor</b>	Professor/Student
<b>Goal in Context</b>	Logout logged in user from the system.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• System should be fully configured.</li><li>• User should be already logged in the system.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• The user decides to log out from the system.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• The user should click on Log out button.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• Professor tries to log out from the system while plagiarism report being generated.</li><li>• Student tries to log out while uploading assignment.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Moderate</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 4</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Frequent</li></ul>

### Use Case 8

<b>Use Case</b>	Update Professor/Student details
<b>Primary Actor</b>	Admin
<b>Goal in Context</b>	To update user(Professor/Student) details in the system.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• User should be able to login to the system as an Admin.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• Admin accepts the request to update user(professor/student) details.</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• System displays the detailed profile view of all the users.</li><li>• Admin selects the user for whom profile needs to be updated.</li><li>• Admin can upgrade/revoke a student access to that of a professor.</li><li>• Admin can also upgrade other profile fields except password.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• Admin tries to upgrade the user not present in the system.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Low</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 5</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Infrequent</li></ul>

### Use Case 9

<b>Use Case</b>	Delete Professor/Student
<b>Primary Actor</b>	Admin
<b>Goal in Context</b>	To delete user(Professor/Student) details in the system.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• User should be able to login to the system as an Admin.</li></ul>
<b>Trigger</b>	<ul style="list-style-type: none"><li>• Admin decides to remove a professor/student from the system</li></ul>
<b>Scenario</b>	<ul style="list-style-type: none"><li>• System displays the detailed profile view of all the users.</li><li>• Admin selects the user for whom profile needs to be deleted.</li><li>• Admin click the delete button next to the selected user.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• Admin tries to delete the user not present in the system.</li></ul>
<b>Priority</b>	<ul style="list-style-type: none"><li>• Moderate</li></ul>
<b>When Available</b>	<ul style="list-style-type: none"><li>• Sprint 5</li></ul>
<b>Frequency</b>	<ul style="list-style-type: none"><li>• Infrequent</li></ul>