

USECASES:

Use Case:	Registration Action
Primary Actor:	Student and Professor
Goal in Context:	To Signup/ register an account on the CopyCatch website by providing the new username, password and email.
Preconditions:	Registered for a course subject under NEU CCIS or should be a Professor/TA under NEU CCIS.
Trigger:	By clicking the Register button in the Login Page.
Scenario:	<ol style="list-style-type: none"> 1. Student/Professor: Visit the CopyCatch login Page 2. Student/Professor: Click the register button for signing up. 3. Student/Professor: Enter the user name in the first text field. 4. Student/Professor: Enter the password in the second text field. 5. Student/Professor: Enter the password again to confirm it in the third text field. 6. Student/Professor: Enter full name in the fourth text field(optional). 7. Student/Professor: Enter the email id in the fifth text field. 8. Student/Professor: Click Register if he/she want to register in this website. 9. Professor: If the user wants to sign up as a professor he need to select the click here link in the highlighted message at top of the page which pops a new dialog box for registration. 10. Professor: In addition to the details that has be entered (step 3 to step 7), a professor can also add message to the admin
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Registration Error: <ol style="list-style-type: none"> a) Should not leave the username or password text field or email field empty. b) Ensure that he/she entered the same password in the second and third text field. c) The User name entered should be unique across Copycatch database. If the User sees the "UserName already taken" error message on the top then he needs to enter new unique name until it shows account created successful message. d) The Password should be of length between 5 and 25 and the user name should be of length 3 and email should be a valid one.

	e) Page Crashes/ Registration failed: The Server may be down. The Primary actor should retry again after some time.
--	---

Use Case:	Login Action
Primary Actor:	Student, Professor and Admin
Goal in Context:	To login into the CopyCatch system using his/her credentials
Preconditions:	Student/Professor/Admin must be registered to login into this system authentically.
Trigger:	By clicking the login button
Scenario:	<ol style="list-style-type: none"> 1. Primary Actor: visits the Copycatch website. 2. Primary Actor: enters his/her unique User Name in the first text field. 3. Primary Actor: enters his/her password in the Second text field. 4. Primary Actor: Click the login button for checking his/her authentication details. 5. Primary Actor: If a user wants to see the password that he/ she enters he can click the show password checkbox in the login page. 6. Student/Professor: Click Register button if he/she has no login credentials. 7. Professor: Click Send Registration Request? Link if a user wants to register himself as a professor
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Login Error: <ol style="list-style-type: none"> a) Should check if he/she entered the correct login credentials. b) Ensure that he/she is entering the credentials in the correct Login Page (Student/Professor). c) Complete the Registration/Signup page if he/she does not have any login credentials. 3. Page Crashes: The Server may be down. The Primary actor should retry again after some time.

Use Case:	User selects to edit their profile (Edit/ Update Action)
Primary Actor:	Student, Professor and Admin (User)
Goal in Context:	Primary actor decides to edit the profile
Preconditions:	<ol style="list-style-type: none"> 1. User is registered as Student/ Professor/ Admin. 2. User must be logged into the CopyCatch website.
Trigger:	By clicking the person icon on the left bottom of any page after logging into the system.
Scenario:	<ol style="list-style-type: none"> 1. The user logs in to the application as Student/ Professor/ Admin. 2. User selects the person icon on the left bottom of any page. 3. User edits some parameters of their profile based on the label displayed above each text field. 4. User clicks the update button. 5. The edits made to the profile are saved into the system. 6. Student/ Professor: Can click "Take me to course page" button to view the user's course page. 7. Admin: Can click "View users" button which is present above all text fields to go to the admin view of the application.
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Update Error: <ol style="list-style-type: none"> a) Should not leave the username or password text field or email field empty. b) Ensure that he/she entered the same password in the third and fourth text field. c) The User name entered should be unique across Copycatch database. If the User sees the "UserName already taken" error message on the top then he needs to enter new unique name until it shows account updated successful message. d) The Password should be of length between 5 and 25 and the user name should be of length 3 and email should be a valid one. e) Page Crashes/ Update failed: The Server may be down. The Primary actor should retry again after some time.

Use Case:	Student/ Professor Course view page
Primary Actor:	Student/ Professor (User)
Goal in Context:	Primary actor views the course page and also can perform create, read, update, and delete action on course across multiple semesters
Preconditions:	1. User is registered as Student/ Professor. 2. User must be logged into the CopyCatch website.
Trigger:	1. By logging into the CopyCatch website. 2. By clicking the “Take me to course page” on the profile page.
Scenario:	1. The user logs in to the application as Student/ Professor. 2. User can view the multiple course that he/she has enrolled for or added to teach across multiple semester. 3. User can click on a particular course that will take him to the assignments view page. 4. User can sort the semester course number, course name and department by clicking the “sort by” button. 5. Student: can click “enroll” for a course under particular semester. 6. User can select different semester from the drop down provided to select semester. 7. Professor: can click “add” a course under particular semester 8. Professor: Can click “view violations” to view all the violations (plagiarism detected) for all the courses that he/she takes.
Exceptions:	1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes/ Update failed: The Server may be down. The Primary actor should retry again after some time.

Use Case:	Student Course Enrollment view page
Primary Actor:	Student
Goal in Context:	Student can enroll for a course under the chosen semester.
Preconditions:	1. Student must be registered on the CopyCatch website. 2. Student must have logged into the CopyCatch website.
Trigger:	1. By clicking the “Enroll” button on the Course view page.
Scenario:	1. Student logs in to the CopyCatch website. 2. Student can view the multiple course that he/she has enrolled for under particular semester. 3. Student can click on the “Enroll” button on the right side which will take them to the course enrollment page. 4. Student can view all the courses that has been offered for that particular semester which he has not registered before. 5. Student: can click “enroll” for any course that has been listed in table. 6. After the course enrollment is successful, the student is redirected to the course view page.

Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes/ Update failed: The Server may be down. The Primary actor should retry again after some time.
-------------	--

Use Case:	Assignment upload page
Primary Actor:	Student
Goal in Context:	Upload an assignment for a particular course.
Preconditions:	<ol style="list-style-type: none"> 1. Student must be registered on the CopyCatch website. 2. Student must have logged into the CopyCatch website.
Trigger:	1. By clicking the particular course on the course view page.
Scenario:	<ol style="list-style-type: none"> 1. Student logs into the CopyCatch website. 2. Student can view the multiple course that he/she has enrolled for under particular semester. 3. Student should click on the particular course for which he/she wants to upload an assignment. 4. Student views all the assignments under the selected course and he clicks any assignment. 5. Student Assignment uploading form is displayed after the assignment table. 6. The student should enter the submission name and GitHub link which is mandatory for an assignment submission in the first and third text field and also, they may enter file description in the second text field. 7. After entering the necessary details on the assignment upload form, the student should click submit button. 8. The student can view the response of their submission above the assignment form.
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Submission Error: <ol style="list-style-type: none"> a) Should not leave the submission name or upload link field empty. b) The uploaded link should be a valid GitHub link. 3. Page Crashes: The Server may be down. The Primary actor should retry again after some time.

Use Case:	Student views submission history
Primary Actor:	Student
Goal in Context:	<ol style="list-style-type: none"> 1. Student must be registered on the CopyCatch website. 2. Student must have logged into the CopyCatch website. 3. View the past submitted assignment.
Preconditions:	Must have submitted some assignments beforehand.
Trigger:	Clicking on "View my Submissions" button from Assignment upload page which will lead to submission history for the particular assignment.

Scenario:	<ol style="list-style-type: none"> 1. Student logs into the CopyCatch website. 2. Student can view the multiple course that he/she has enrolled for under particular semester. 3. Student should click on the particular course for which he/she wants to view the submission history for an assignment. 4. Student: Upload assignments in the assignment submission page if he/she has not submitted any assignment so far. 5. For each assignment under that course, "view my submissions" button will take the students to view the multiple stats of that particular assignment. 6. Student can go back to the assignment upload page by clicking the "Go to Assignment Submission Page".
Exceptions:	<ol style="list-style-type: none"> 1. Page Crashes: The Server may be down. The Primary actor should retry again after some time. 2. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website.

Use Case:	Professor Assignment View page
Primary Actor:	Professor
Goal in Context:	View the Assignments that has been created for a particular course under particular semester
Preconditions:	<ol style="list-style-type: none"> 1. Professor must be registered on the CopyCatch website. 2. Professor must have logged into the CopyCatch website. 3. Professor must have added at least 1 course to his/her course list and added atleast 1 assignment to it.
Trigger:	1. By clicking the particular course on the course view page.
Scenario:	<ol style="list-style-type: none"> 1. Professor logs into the CopyCatch website. 2. Professor can view the multiple course that he/she has added for teaching under particular semester. 3. Professor should click on the particular course for which he/she wants to view the assignments. 4. Professor views all the assignments of the selected course under selected semester. 5. In the table that lists all the assignment, the table header such as name, and description has arrow mark near them clicking which sort that column alphabetically. 6. For each assignment provided an option to edit and delete the assignments. 7. Add Button after the table in the right bottom can be used to add new assignment.
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes: The Server may be down. The Primary actor should retry again after some time.

Use Case:	Professor Assignment Create page
Primary Actor:	Professor
Goal in Context:	Professor creates an Assignment for a particular course under particular semester
Preconditions:	<ol style="list-style-type: none"> 1. Professor must be registered on the CopyCatch website. 2. Professor must have logged into the CopyCatch website. 3. Professor must have added at least 1 course to his/her course list.
Trigger:	1. By clicking the Add button on the assignment view page.
Scenario:	<ol style="list-style-type: none"> 1. Professor logs into the CopyCatch website. 2. Professor can view the multiple course that he/she has added for teaching under particular semester. 3. Professor should click on the particular course for which he/she wants to add the assignments. 4. Professor views all the assignments of the selected course under selected semester. 5. Professor should click "Add" Button after the table in the right bottom to add new assignment. 6. The Page has two parts now, in the left side the professor can view the old assignments and on the right side the professor can create a new assignment. 7. In the first text field, the professor can enter the new assignment name. 8. In the second text field, the professor can enter the description of the new assignment. 9. The tick button creates the new assignment and the delete button resets the text field to empty. 10. After successfully the assignment has been created, the professor can now view the created assignment on the left side of the page.
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes: The Server may be down. Therefore, the assignment has not been created in the database and so the new assignment is not reflected in the left part of the page The Primary actor should retry again after some time.

Use Case:	Professor Assignment Edit page
Primary Actor:	Professor
Goal in Context:	Professor edits an Assignment for a particular course under particular semester
Preconditions:	<ol style="list-style-type: none"> 1. Professor must be registered on the CopyCatch website. 2. Professor must have logged into the CopyCatch website. 3. Professor must have added at least 1 course to his/her course list. 4. Professor must have created at least 1 assignment before.

Trigger:	1. By clicking the Edit icon on a particular assignment inside the table on the assignment view page.
Scenario:	<ol style="list-style-type: none"> 1. Professor logs into the CopyCatch website. 2. Professor can view the multiple course that he/she has added for teaching under particular semester. 3. Professor should click on the particular course for which he/she wants to edit the assignments. 4. Professor views all the assignments of the selected course under selected semester. 5. Professor should click "Edit" icon on a particular assignment which he/she wants to edit. 6. The Page has two parts now, in the left side the professor can view the old assignments and on the right side the professor can edit the assignment that he has selected. 7. In the first text field, the professor can enter the updated assignment name. 8. In the second text field, the professor can enter the description of the updated assignment. 9. The tick button updates the assignment and after successfully the assignment has been updated, the professor can now view the created assignment on the left side of the page.
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes: The Server may be down. Therefore, the assignment has not been updated in the database and so the edited assignment is not reflected in the left part of the page The Primary actor should retry again after some time.

Use Case:	Professor Assignment Delete page
Primary Actor:	Professor
Goal in Context:	Professor deletes an Assignment for a particular course under particular semester
Preconditions:	<ol style="list-style-type: none"> 1. Professor must be registered on the CopyCatch website. 2. Professor must have logged into the CopyCatch website. 3. Professor must have added at least 1 course to his/her course list. 4. Professor must have created at least 1 assignment before.
Trigger:	1. By clicking the Delete icon on a particular assignment inside the table on the assignment view page.
Scenario:	<ol style="list-style-type: none"> 1. Professor logs into the CopyCatch website. 2. Professor can view the multiple course that he/she has added for teaching under particular semester. 3. Professor should click on the particular course for which he/she wants to edit the assignments.

	<p>4. Professor views all the assignments of the selected course under selected semester.</p> <p>5. Professor should click “Delete” icon on a particular assignment which he/she wants to delete.</p> <p>6. The Page is basically for the professor confirmation to delete the assignment.</p> <p>7. The Title and About is used for describing the assignment and delete button is for the confirmation of assignment deletion.</p> <p>8. By clicking the delete button the assignment is deleted and the professor is now redirected to the Assignment view page.</p> <p>9. After successful deletion the professor can see the assignment that is removed from the assignment view table.</p>
Exceptions:	<p>1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website.</p> <p>2. Page Crashes: The Server may be down. Therefore, the assignment has not been deleted in the database and so the assignment to be deleted is still reflected inside the assignment table. The Primary actor should retry again after some time.</p>

Use Case:	Professor Course Create page
Primary Actor:	Professor
Goal in Context:	Professor creates a course under particular semester
Preconditions:	<p>1. Professor must be registered on the CopyCatch website.</p> <p>2. Professor must have logged into the CopyCatch website.</p>
Trigger:	1. By clicking the Add button on the course view page.
Scenario:	<p>1. Professor logs into the CopyCatch website.</p> <p>2. Professor can view the multiple course that he/she has added for teaching under particular semester.</p> <p>3. Professor should click the “Add +” button on the course view page.</p> <p>4. The Page has two parts now, in the left side the professor can view the old courses that he has created before for that semester and on the right side the professor can create a new course.</p> <p>5. In the first text field, the professor should enter the new Course ID.</p> <p>6. In the second text field, the professor should enter the new Course Name.</p> <p>7. In the third text field, the professor should enter the Section Name for the new Course.</p> <p>8. In the fourth text field, the professor should enter the Department under which the new course belongs to.</p> <p>9. The tick button creates the new assignment and the delete button resets the text field to empty.</p> <p>10. After the course has been created successfully, the professor can now view the created course on the left side of the page.</p>

Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes: The Server may be down. Therefore, the course is not created in the database and so the new course is not reflected in the left part of the page The Primary actor should retry again after some time.
-------------	--

Use Case:	Professor Course Edit page
Primary Actor:	Professor
Goal in Context:	Professor edits a course under particular semester
Preconditions:	<ol style="list-style-type: none"> 1. Professor must be registered on the CopyCatch website. 2. Professor must have logged into the CopyCatch website. 3. Professor must have added at least 1 course to his/her course list.
Trigger:	1. By clicking the Edit icon on a particular course on the course view page.
Scenario:	<ol style="list-style-type: none"> 1. Professor logs into the CopyCatch website. 2. Professor can view the multiple course that he/she has added for teaching under particular semester. 3. Professor should click "Edit" icon on the particular course which he/she wants to edit on the course view page. 4. The Page has two parts now, in the left side the professor can view all the course created before and on the right side the professor can edit the course that he has selected. 5. On the left side of the page, the professor can choose any course that he/she wants to edit. 6. On the right side of the page, the professor can edit the course that has/she has selected 7. In the first three text fields, the professor can enter the updated ID/ name/ department. 8. The fourth text field is to set the threshold percentage for plagiarism detection which is an integer that has range between 1 to 100. 9. After that the professor can see the dropdown button to choose the programming language for the selected course. 10. The tick button updates the course and after the course has been updated successfully, the professor can now view the updated course on the left side of the page.
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes: The Server may be down. Therefore, the course has not been updated in the database and so the edited course is not reflected on the left part of the page. The Primary actor should retry again after some time.

Use Case:	Professor Course Delete page
Primary Actor:	Professor
Goal in Context:	Professor deletes a particular course under particular semester
Preconditions:	<ol style="list-style-type: none"> 1. Professor must be registered on the CopyCatch website. 2. Professor must have logged into the CopyCatch website. 3. Professor must have added at least 1 course to his/her course list.
Trigger:	<ol style="list-style-type: none"> 1. By clicking the Delete icon on a particular course on the course view page.
Scenario:	<ol style="list-style-type: none"> 1. Professor logs into the CopyCatch website. 2. Professor can view the multiple course that he/she has added for teaching under particular semester. 3. Professor should click "Delete" icon on a particular course which he/she wants to delete on the course view page. 4. Now, the redirected page is basically for the professor confirmation to delete the assignment. 5. The Title and About is used for describing the course and delete button is for the confirmation of course deletion. 6. By clicking the delete button the course is deleted from the course list and the professor is now redirected to the course view page. 7. After successful deletion the professor can see the course that is removed from the course view page.
Exceptions:	<ol style="list-style-type: none"> 1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website. 2. Page Crashes: The Server may be down. Therefore, the course has not been deleted in the database and so the course to be deleted is still reflected on the course view page. The Primary actor should retry again after some time.

Use Case:	Admin View
Primary Actor:	Admin
Goal in Context:	Admin can create a user, can approve professor registration request, and can do edit/ delete a student or professor.
Preconditions:	<ol style="list-style-type: none"> 1. Admin must be registered on the CopyCatch website. 2. Admin must have logged into the CopyCatch website.
Trigger:	<ol style="list-style-type: none"> 1. By logging into the CopyCatch Website as an Admin user.
Scenario:	<ol style="list-style-type: none"> 1. Admin login to the CopyCatch website. 2. Now, the admin can see the webpage divided in 2 parts, the left side describes the type of operations he can perform and right side is appropriate content displayed for the selected operation on the left side. 3. By clicking the first functionality" Professor Registration Requests" on the left side, the admin can view all the approved and pending professor registration request on the right side. 4. The admin can make decisions for each professor registration request by clicking the accept and reject button.

	<p>5. By clicking the second functionality “All Professors” on the left side, the admin can see the list of all professors on the right side. The admin can perform edit, update, and delete operation on any professor record through this page.</p> <p>6. By clicking the third functionality “All Students” on the left side, the admin can see the list of all students on the right side. The admin can perform edit, update, and delete operation on any student record through this page.</p> <p>7. By clicking the fourth functionality “Create an User” button, the admin can create/register a user as a student/ professor/ admin through this page.</p>
Exceptions:	<p>1. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website.</p> <p>2. Admin Page Error:</p> <p>a) Page Crash/Admin operation may fail: The Server may be down which cause create/ edit/ update/ delete operation to fail.</p> <p>b) If the entered value does not meet the username, password, email criteria then it will throw the error message. The Admin has to enter the data that satisfies the required constraints.</p>

Use Case:	Professor Submission comparison page
Primary Actor:	Professor
Goal in Context:	To initiate the Assignment comparison for a particular course and comparison can be initiated across multiple section and multiple semester by professor discretion.
Preconditions:	<p>1. Professor must be registered on the CopyCatch website.</p> <p>2. Professor must have logged into the CopyCatch website.</p> <p>3. Professor must have added at least 1 course to his/her course list.</p> <p>4. Professor must have created at least 1 assignment before.</p>
Trigger:	1.By clicking the Compare/ Compare All button in the Assignment Submission Comparison Page
Scenario:	<p>1. Professor logs into the CopyCatch website.</p> <p>2. Professor can view the multiple course that he/she has added for teaching under particular semester.</p> <p>3. Professor should click on the particular course for which he/she wants to perform the comparison.</p> <p>4. Professor views all the assignments of the selected course under selected semester.</p> <p>5. Professor can click on any assignment to perform the comparison operation.</p> <p>6. Now, the professor is redirected to the page where he/she can see all submissions for the particular assignment.</p> <p>7. There are two additional & optional functionalities added such as cross semester and cross section comparison which can be selected by the professor by adding the semesters to the</p>

	<p>comparison list and also selecting the cross-section check box respectively.</p> <p>8. Compare all button performs a brute force way of comparing all assignment submissions on that page.</p> <p>9. Professor can also initiate the comparison by selecting few assignments listed on the Submission page.</p> <p>10. Once the comparison is initiated, the professor can see the progression status bar running, once it is completed, the view result button is showed.</p> <p>11. By clicking the view results button, the professor can see the results of the comparison in which the red highlighted parts are due to the plagiarism detected students and others are not.</p> <p>12. The professor can view/ share report to his/her email.</p>
Exceptions:	<p>1. Page Crashes: The Server may be down. The Primary actor should retry again after some time.</p> <p>2. If there is no submission for that particular assignment then the viewing results after the comparison action will result in an empty table with no reports to view.</p> <p>3. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website.</p>

Use Case:	Professor View Violation Summary page
Primary Actor:	Professor
Goal in Context:	To view the Plagiarized reports directly and quickly of all his/her courses because of incremental comparison approach implementation without initiating any comparisons from the assignment submission page.
Preconditions:	<p>1. Professor must be registered on the CopyCatch website.</p> <p>2. Professor must have logged into the CopyCatch website.</p> <p>3. Professor must have added at least 1 course to his/her course list.</p> <p>4. Professor must have created at least 1 assignment before.</p>
Trigger:	1. By clicking the View Violations button on the professor course view page.
Scenario:	<p>1. Professor logs into the CopyCatch website.</p> <p>2. Professor can view the multiple course that he/she has added for teaching under particular semester.</p> <p>3. By clicking the "View Violations" button on the course view page the professor can view all the plagiarized records quickly for all the courses that he/she is taking.</p> <p>4. Professor views all the assignments of the selected course under selected semester.</p> <p>5. Each entry in the table have a brief view about the student, course, section, and assignment that the student has been caught for plagiarism</p> <p>6. The view report button shows the detailed report of plagiarism.</p>

	<p>are due to the plagiarism detected students and others are not.</p> <p>7. The results are accumulated over the time as the student get caught for plagiarism as the submission is checked for plagiarism incrementally after each student makes their submissions.</p>
Exceptions:	<p>1. Page Crashes: The Server may be down. The Primary actor should retry again after some time.</p> <p>2. If there is no submissions/ no assignment added for any of the courses that the professor takes or the professor doesn't have any course added to his/her course list then the view results page shows empty table.</p> <p>3. CopyCatch Website not loaded: Check network connection, website link is correct and revisit the website.</p>