

Use Cases- Team 203

1.

| | |
|------------------|--|
| Use Case: | Login Action |
| Primary Actor: | Student and Professor |
| Goal in Context: | To login into the CopyCatch system using his/her credentials |
| Preconditions: | Student/Professor must be registered to login into this system authentically. |
| Trigger: | By clicking the login button |
| Scenario: | <ol style="list-style-type: none">1. Student/Professor: visits the Copycatch website.2. Professor: If the User is a Professor then he will click the 'Are you a Professor?' hyperlink and gets redirected to Professor Login Page otherwise, will remain in the same Login Page.3. Student/Professor: enters his/her User Name in the first text field.4. Student/Professor: enters his/her password in the Second text field.5. Student/ Professor: Click the login button for checking his/her authentication details.6. Student/Professor: Click Register button if he/she has no login credentials. |
| 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. |

2.

| | |
|------------------|---|
| 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 and password. |
| Preconditions: | Registered for a course subject 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: Click the Student radio button if the primary actor is a student. 7. Professor: Click the Professor radio button if the primary actor is a Professor. 8. Student/Professor: Click Register if he/she want to register in this website. 9. Student/Professor: By clicking the cancel button he/she can cancel this use case action anytime. |
| 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 empty. b) Ensure that he/she entered the same password in the second and third text field. <p>Page Crashes: The Server may be down. The Primary actor should retry again after some time.</p> |

3.

| | |
|------------------|---|
| Use Case: | Student views submission history |
| Primary Actor: | Student |
| Goal in Context: | View the past submitted assignment. |
| Preconditions: | Must have submitted some assignments beforehand. |
| Trigger: | Clicking on assignment submission button from student page which will lead to submission history. |
| Scenario: | <ol style="list-style-type: none"> 1. Student: If the student does not have login credentials/ visiting the website for first time. He/she should register to use the CopyCatch. 2. Student: Login to CopyCatch website using his/her login credentials. 3. Student: In the Student welcome page, he/she must click the assignment submission button to view/submit their assignments. 4. Student: Upload assignments in the assignment submission page if he/she has not submitted any assignment so far. 5. Assignment must be uploaded successfully. 6. Student: After the success of assignment upload action then the student can view the submitted assignments listed on the 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. <ol style="list-style-type: none"> a) The Student must login again if logged out and reach this upload assignment page to view submission history and upload it again if the upload action is failed. 2. Upload action crashes: <ol style="list-style-type: none"> a) The student then need to reload the page and continue the upload action to view the desired submission history. |

4.

| | |
|------------------|--|
| Use Case 2: | <i>View Report After Plagiarism Check</i> |
| Primary Actor: | Professor |
| Goal in Context: | Views the report created by the application after the professor has requested for a comparison of home works, either within the same section or across multiple sections. It shows the summary of files scanned under the chosen section(s) for plagiarism detection |
| Preconditions: | <ol style="list-style-type: none"> 1. The software should be appropriately installed and the professor should be able to log into the system. 2. Professor selected the section(s) to compare the assignments and plagiarism scan has been completed. |
| Trigger: | Click on the “view reports” button and the user is navigated to the reports page |
| Scenario: | <ol style="list-style-type: none"> 1. Professor: Logs into the application 2. Professor: Enters the user name and password 3. Professor: Enters his/her homepage and observes the courses he teaches. 4. Professor: Checks a single course if he only wants to check plagiarism within a class. 5. Professor. Checks multiple sections of the same course if he wants to check plagiarism of student submissions across multiple sections. 6. Professor: Click on “Compare” button once checking single or multiple sections for plagiarism comparison. 7. Professor: Clicks on the “view results” button once the scan is complete 8. The view results page will have a list of files that matched above a certain threshold 9. Clicking on view reports takes you to a page which shows similarities between two codes. This will highlight the text that is similar between two codes. |

Use Cases- Team 203

| | |
|-----------------|--|
| | Also a percentage match along with student information is shown |
| Exception: | <ol style="list-style-type: none">1. Unable to login due to incorrect username or password. Remedy: Try logging in again.2. Fails to load the report page due to any failure. Remedy: Start Scan again. |
| Priority: | High Priority |
| When available: | First release. |

5.

| | |
|------------------|---|
| Use Case 1: | <i>Student Uploads Homework</i> |
| Primary Actor: | Student |
| Goal in Context: | Uploads the homework into the web application |
| Preconditions: | <ol style="list-style-type: none">1. The software should be appropriately installed and the student should be able to log into the system.2. There is an upload button in the student page for uploading assignments in the home page.3. The format and extension of the files to be uploaded are specified on the home page. |
| Trigger: | Click on the “upload” button |
| Scenario: | <ol style="list-style-type: none">1. Student: Logs into the application2. Student: Enters the user name and password3. Student: Enters his/her homepage and sees the courses he/she has been enrolled in.4. Student: Selects the course for uploading the homework. |

Use Cases- Team 203

| | |
|------------|---|
| | <ol style="list-style-type: none"> 5. Student: Clicks on "Assignment submissions" 6. Student: Choose the assignment file and uploads it. 7. After the file is successfully uploaded, the filename appears in the table of submitted files |
| Exception: | <ol style="list-style-type: none"> 1. Unable to login due to incorrect username or password. Remedy: Try logging in again. 2. Web page throws an error message saying files are not in recognizable format. Remedy: Modify and upload files in the allowed extensions and format specified on the page. 3. Fails upload when the files exceed the size limit. Remedy: Upload files within the allowed size limit. 4. Upload times out due to any failure. Remedy: Try uploading again |

6.

| | |
|-------------------------|--|
| Use Case: | Professor selects to compare the code within same section |
| Primary Actor: | User registered as Professor in system |
| Goal in Context: | Professor decides to check for plagiarism in source code within single section of a class that he/she teaches |
| Preconditions: | <ol style="list-style-type: none"> 1. User is registered as Professor 2. User added themselves to a class as professor 3. The class under a section contains submissions of students |
| Trigger: | The source code files within a single section will be compared for plagiarism |
| Scenario: | <ol style="list-style-type: none"> 1. The user logs in to the application as Professor 2. The user is navigated to the Class View page 3. Professor can view all the classes that he is teaching 4. Professor selects the checkbox besides one class for example: CS5500 Section 1 5. The option to Compare code is enabled 6. The professor selects that option and the plagiarism detector runs for class CS5500 Section 1 7. The loader sign comes up showing the process is comparing source files 8. Once loader reaches 100% and shows finished means the comparisons are done and report is generated 9. View report button is visible once the processing is done |
| Exceptions: | <ol style="list-style-type: none"> 1. Id or password to log in to the system is incorrect 2. The incorrect submission was not successful and user cannot see his/her submission |
| Priority: | High Priority |
| Use Case: | Professor selects to compare the code across sections |

Use Cases- Team 203

| | |
|-------------------------|---|
| Primary Actor: | User registered as Professor in system |
| Goal in Context: | Professor decides to check for plagiarism in source code between two section of a class that he/she teaches |
| Preconditions: | <ol style="list-style-type: none"> 1. User is registered as Professor 2. User added themselves to a class as professor 3. The class under a section contains submissions of students |
| Trigger: | The source code files within one section will be compared with another selected section for plagiarism |
| Scenario: | <ol style="list-style-type: none"> 1. The user logs in to the application as Professor 2. The user is navigated to the Class View page 3. Professor can view all the classes that he is teaching 4. Professor selects the checkbox besides one class for example: CS5500 Section 1 5. Professor selects the checkbox besides second class for example: CS550 Section 2 6. All the other checkboxes for courses under sections are disabled 7. With this option, the comparison will be done cross section i.e. codes from section 1 will be checked for codes from section 2 for plagiarism 8. The option to Compare code is enabled 9. The professor selects that option and the plagiarism detector runs for class CS5500 Section 1 and CS5500 Section 2 10. The loader sign comes up showing the process is comparing source files 11. Once loader reaches 100% and shows finished means the comparisons are done and report is generated |
| Exceptions: | <ol style="list-style-type: none"> 1. Id or password to log in to the system is incorrect |
| Priority: | High Priority |

7.

| | |
|-------------------------|---|
| Use Case: | User selects to edit their profile |
| Primary Actor: | User registered as Professor in system |
| Goal in Context: | Professor decides to edit the profile |
| Preconditions: | <ol style="list-style-type: none"> 1. User is registered as Professor 2. Professor has created their profile on the system |
| Trigger: | The profile is edited for the Professor |
| Scenario: | <ol style="list-style-type: none"> 1. The user logs in to the application as Professor 2. The user is navigated to the Class View page 3. Professor selects to edit their own profile 4. Professor edits some parameters of their profile 5. The user Professor selects save |

Use Cases- Team 203

| | |
|--------------------|--|
| | 6. The edits made to the profile are saved into the system |
| Exceptions: | 1. Id or password to log in to the system is incorrect 2. User enters incorrect values in username, password fields |

8.

| | |
|-------------------------|---|
| Use Case: | User selects to edit their profile |
| Primary Actor: | User registered as Student in system |
| Goal in Context: | Student decides to edit the profile |
| Preconditions: | 1. User is registered as Student 2. Student has created their profile on the system |
| Trigger: | The profile is edited for the Student |
| Scenario: | 1. The user logs in to the application as Student 2. The user is navigated to the Class View page 3. Student selects to edit their own profile 4. Student edits some parameters of their profile 5. The user Student selects save 6. The edits made to the profile are saved into the system |
| Exceptions: | 1. Id or password to log in to the system is incorrect 2. User enters incorrect values in username, password fields |