

Use Cases for Kitty Professor

Feature: - Sign Up / Login/ Sign Out

Case 1: User chooses to create an account	
Use Case:	User chooses to create an account.
Primary Actor:	Anonymous user
Goal in Context:	To enable the user to create an account on the plagiarism checker
Preconditions:	User should be on the homepage of the plagiarism checker.
Trigger:	User clicks on the 'Sign Up' button
Scenario:	User clicks on the 'Sign Up' button User types in the required details User clicks on the 'Sign Up' button
Exceptions:	User might enter leave a field blank
Status	Completed

Case 2: User logs into the system	
Use Case:	User logs into the system
Primary Actor:	User (Faculty, Admin, Student)
Goal in Context:	To enable the user to log in
Preconditions:	User should have created an account
Trigger:	User clicks on "Log In"
Scenario:	User enter his/her credentials and clicks on Log In
Exceptions:	The user doesn't have an account The user enters the wrong combination of Username and Password
Status	Completed

Case 3: User logs into the system using Google	
Use Case:	User logs into the system using google
Primary Actor:	User (Faculty, Admin, Student)
Goal in Context:	To enable the user to log in
Preconditions:	User should be on the website
Trigger:	User clicks on "Sign in with Google"
Scenario:	User clicks on "Sign in with Google" User enters his/her Google credentials Users logs in
Exceptions:	The user doesn't have a Google account The user enters the wrong combination of Username and Password
Status	Not started

Case 4: User logs into the system using MyNEU	
Use Case:	User logs into the system using MyNEU
Primary Actor:	User (Faculty, Admin, Student)
Goal in Context:	To enable the user to log in
Preconditions:	User should be on the website
Trigger:	User clicks on "Sign in with MyNEU"
Scenario:	User clicks on "Sign in with MyNEU" User enters his/her MyNEU credentials Users logs in
Exceptions:	The user doesn't have a MyNEU account The user enters the wrong combination of Username and Password
Status	Not started

Case 5: Sign out	
Use Case:	User clicks on the 'Sign Out' button
Primary Actor:	User (Faculty, Admin, Student)
Goal in Context:	Enable the user to log out of the plagiarism checker
Preconditions:	User should be logged into the plagiarism checker
Trigger:	User clicks the 'Sign Out' button.
Scenario:	User clicks on the 'User Name' drop down User clicks on the 'Sign Out' button User is logged out of the plagiarism checker
Exceptions:	The user has already signed out and this is just another tab
Status	Completed

Feature: - Submission of assignment

Case 6: Student uploads a zip file as a submission to an assignment	
Use Case:	Student uploads a zip file as a submission to an assignment
Primary Actor:	User (Student)
Goal in Context:	To enable the student to upload a zip file as a submission to an assignment
Preconditions:	Student should be logged in and should be enrolled in the course
Trigger:	User clicks on Submit Assignment
Scenario:	User clicks on Submit Assignment and choose a zip file
Exceptions:	User submits an empty zip file
Status	Completed

Case 7: Student gives a github link as a submission to an assignment	
Use Case:	Student gives a github link as a submission to an assignment
Primary Actor:	User (Student)
Goal in Context:	To enable the student to give a github link as a submission to an assignment
Preconditions:	Student should be logged in and should be enrolled in the course
Trigger:	User clicks on Submit Assignment
Scenario:	User clicks on Submit Assignment and choose to gives a github link User gives the github link
Exceptions:	The links provided is an empty repository
Status	Completed

Case 8: Student submits an assignment again	
Use Case:	Student submits an assignment again
Primary Actor:	User (Student)
Goal in Context:	To enable student to submit an assignment again
Preconditions:	Student should be logged in and should be enrolled in the course
Trigger:	User clicks on Submit Assignment
Scenario:	User clicks on Submit Assignment and choose a zip file/ github link
Exceptions:	User submits an empty zip file and or an empty repo
Status	Completed

Feature: - Admin rights

Case 9: Admin changes role of a user	
Use Case:	Admin changes role of a user
Primary Actor:	User (Admin)
Goal in Context:	To enable the admin changes the role of users
Preconditions:	A user should be present in the system
Trigger:	Admin clicks on a user
Scenario:	Admin clicks on a user Admin choose a new role for the user
Exceptions:	No users are present in the system
Status	Completed

Case 10: Admin deletes a user	
Use Case:	Admin deletes a user
Primary Actor:	User (Admin)
Goal in Context:	To enable the admin delete a user
Preconditions:	A user should be present in the system
Trigger:	Admin clicks on a user
Scenario:	Admin chooses a user Admin clicks on the delete button
Exceptions:	No users are present in the system
Status	Completed

Case 11: Admin renames a user	
Use Case:	Admin renames a user
Primary Actor:	User (Admin)
Goal in Context:	To enable the admin to rename a user
Preconditions:	A user should be present in the system
Trigger:	Admin clicks on a user
Scenario:	Admin clicks on a user Admin gives a new name to the user
Exceptions:	No users are present in the system
Status	Completed

Feature: Selecting Semester, adding courses and assignments, setting threshold and selecting language

Case 12: Faculty selects a semester	
Use Case:	Faculty selects a semester
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to select a semester
Preconditions:	Faculty should be logged in
Trigger:	Faculty clicks on the semester drop-down
Scenario:	Faculty clicks on the semester drop-down Faculty selects a semester
Exceptions:	Faculty clicks on Add without giving the course code and name
Status	Completed

Case 13: Faculty adds a course	
Use Case:	Faculty adds a course
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to add a course
Preconditions:	Faculty should be logged in and should have selected a semester
Trigger:	Faculty clicks on add a new course
Scenario:	Faculty clicks on add a new course Gives the course code Gives the course name Clicks on Add
Exceptions:	Faculty clicks on Add without giving the course code and name
Status	Completed

Case 14: Faculty names an assignment	
Use Case:	Faculty names an assignment
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to name an assignment
Preconditions:	Faculty should be logged in and should have added a course
Trigger:	Faculty clicks on add a new assignment
Scenario:	Faculty selects a course Then clicks on Assignments Clicks on Add a new assignment Names the assignment
Exceptions:	Faculty clicks on Add without giving the Name
Status	Completed

Case 15: Faculty sets the threshold for the assignment	
Use Case:	Faculty sets the threshold for the assignment
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to set threshold for an assignment
Preconditions:	Faculty should be logged in and should have added a course
Trigger:	Faculty clicks on add a new assignment
Scenario:	Faculty types in the threshold value
Exceptions:	Faculty clicks on Add without giving the threshold value
Status	Completed

Case 16: Faculty selects a language and adds the assignment	
Use Case:	Faculty selects a language and adds the assignment
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to select a language and add the assignment
Preconditions:	Faculty should be logged in and should have added a course
Trigger:	Faculty clicks on add a new assignment
Scenario:	Faculty selects the language of the assignment Clicks on Add
Exceptions:	Faculty clicks on Add without selecting the language
Status	Completed

Feature: Viewing plagiarism report

Case 17: Faculty views comparison reports of students	
Use Case:	Faculty views comparison reports of students
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to view comparison reports of students
Preconditions:	Students should have submitted assignments
Trigger:	Faculty clicks on view all reports and selects a report
Scenario:	Faculty clicks on view all reports Faculty selects a report Faculty checks the report
Exceptions:	Students haven't submitted their assignments
Status	Completed

Feature: Informing students caught in plagiarism

Case 18: Faculty sends emails to students caught in plagiarism	
Use Case:	Faculty sends emails to students caught in plagiarism
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to send emails to students caught in plagiarism
Preconditions:	Students should be caught in plagiarism
Trigger:	Faculty clicks on view all reports and selects a report
Scenario:	Faculty clicks on view all reports Faculty selects a report Faculty adds a message and sends Students receive an email with the content as the message
Exceptions:	Students haven't been caught in plagiarism
Status	Completed

Feature: Plagiarism checking and informing faculty about plagiarism

Case 19: System runs the plagiarism checker	
Use Case:	System runs the plagiarism checker
Primary Actor:	User (System)
Goal in Context:	To enable the system to run the plagiarism checker
Preconditions:	Students should have submitted assignments
Trigger:	Students submit their assignments
Scenario:	Student submit their assignment System runs the checker
Exceptions:	Students haven't submitted their assignments
Status	Completed

Case 20: System sends an email to faculty about student caught in plagiarism	
Use Case:	System sends an email to faculty about student caught in plagiarism
Primary Actor:	User (System)
Goal in Context:	To enable the system to send an email to faculty about student caught in plagiarism
Preconditions:	Students should be caught in plagiarism
Trigger:	Student submit a similar assignment that matches above threshold given by the professor
Scenario:	Students submit a similar assignment that matches the threshold given by the professor System runs the checker System catches the student who copied System make a report on plagiarism and send an email to the professor about it with a link to the report
Exceptions:	Students haven't been caught in plagiarism
Status	Completed

Case 21: System uses more than one comparison strategy and computes an overall score using a weighted polynomial function.	
Use Case:	System uses more than one comparison strategy and computes an overall score using a weighted polynomial function.
Primary Actor:	User (System)
Goal in Context:	To enable the system to use more than one comparison strategy and computes an overall score using a weighted polynomial function.
Preconditions:	System should have atleast two submitted assignments
Trigger:	System gets the second assignment submission
Scenario:	System gets the second assignment submission
Exceptions:	System doesn't have two assignments
Status	Not started

Feature: Plagiarism checking across other sections and semesters

Case 22: Faculty chooses the semesters and sections with which the assignments will be compared for plagiarism	
Use Case:	Faculty chooses the semesters and sections with which the assignments will be compared for plagiarism
Primary Actor:	User (Faculty)
Goal in Context:	To enable the faculty to choose the semesters and sections with which the assignments will be compared for plagiarism
Preconditions:	There should be some courses present in the system
Trigger:	Faculty clicks on add a new course
Scenario:	Faculty clicks on add a new course Faculty enter a course code If the course code is same as of a section of this semester or some other semester, then faculty get a prompt and can choose to select the section If selected, the assignment of this new course will be compared with the assignments of the selected course Faculty clicks on Add
Exceptions:	No courses are present in the system
Status	Completed