<Code Karin>

<Virtual Classroom>
Use Cases

Version <1.3>

Revision History

Date	Version	Description	Author
<10/10/21>	<1.0>	<first iteration, initial documentation>	<Code Karin Team>
<01/19/22>	<1.1>	<update a few use cases>	<Kate Brayshaw>
<01/26/22>	<1.2>	<fill in missing business rules and assumptions. and fix grammar>	<Kate Brayshaw>
<01/28/22>	<1.3>	<Final edits>	<Kate Brayshaw>

Table of Contents

Use Cases

Use Case List

Primary Actor	**Use Cases**
Admin	Manage Classroom
Professor	Manage Student
Student	Manage User Rule
	Manage Contest
	Manage Discussion Board
	Manage Student Participation
	Do Practice (Contest)

Use Case 1: Manage Classroom

UC ID and Name:	UC-1: Manage Classroom		
Created By:	Jacob Hollis	Date Created:	October 10, 2021
Primary Actor:	Professor	Secondary Actors:	Admin
Trigger:	A Patron indicates that she wants to order a meal.		
Description:	A Patron accesses the Cafeteria Ordering System from either the corporate intranet or external Internet, views the menu for a specific date, selects food items, and places an order for a meal to be picked up in the cafeteria or delivered to a specified location within a specified 15-minute time window.		
Preconditions:	PRE-1. The user has professor permissions		
Postconditions:	POST-1. The user can create, view, delete, update virtual classrooms POST-2. If the user does not make any change, the database should remain the same		
Main Success Scenario:	1. The user indicates that he/she wants to add a new classroom 2. The system presents a form for the user to fill in. 3. The user fills in information about the classroom. 4. The system presents the user with a preview of the classroom 5. The user confirms the pre-view information of the classroom is correct 6. The system adds the new classroom to the database 7. The system indicates to the user that the classroom has been created		
Extensions:	1a: The user indicates that he/she wants to delete an existing classroom 1a1. The system presents a prompt to ask the user to type their password to verify for the delete change 1a2. The user type in the password 1a3. The system verifies if the password matches 1a3a. The password is not correct 1a3b. The system indicates to the user that the classroom cannot be deleted 1a3c. The use case returns to 1a. 1a4. The system deletes the classroom 1a5. The system indicates to the user that the classroom has been deleted 1b: The user indicates that he/she wants to update an existing classroom 1b1. The system presents a form for the user to edit information. 1b2. The user fills in information about the classroom. 1b3. The system presents the user with a preview of the classroom 1b4. The user confirms the pre-view information of the classroom is correct 1b5. The system updates the classroom to the database 1b6. The system indicates to the user that the classroom has been edited		
Priority:	High		
Frequency of Use:	Essential use case for professors, as many users as professors that adopt the platform. Initially 1 professor.		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	A front end portal exists for the professor to interact with students.		

Use Case 2: Manage Student

UC ID and Name:	UC-1: Manage Student		
Created By:	Kate Brayshaw	Date Created:	October 10, 2021
Primary Actor:	Professor	Secondary Actor:	Admin
Trigger:	A professor/admin wants to add/delete student		
Description:	The professor or admin can choose to add and remove students from the virtual classroom. The user will be prompted to fill in personally identifiable information which will allow the addition or subtraction of said student.		
Preconditions:	PRE-1. The user has professor permissions with professor interface PRE-2. There are existing classrooms.		
Postconditions:	POST-1. The user can delete/add students to classrooms.		
Main Success Scenario:	1. The user indicates that he/she wants to add a new student to an existing classroom 2. The system presents the user with a form to fill in(Classroom ID, student's email, student's name) 3. The user fills in the required information. 4. The system presents a preview for the user. 5. The user confirms. 6. The system adds the new student to database with Non-registered status 7. The system indicates to the user that the new student is successfully added 8. The system sends email to the new student's email address to verify student account		
Extensions:	1a: The user indicates that he/she wants to delete an existing student of an existing classroom 1a1.The system presents the user with an option to delete an existing student 1a2. The user choose to delete an existing student 1a3. The system verifies the user's password 1a4. The user types in a correct password 1a4a. The user types in an incorrect password 1a4b. The use case returns to 1a3. 1a5. The system presents a confirmation form 1a6. The user confirms to delete the students 1a7. The system deletes the student in the database 1a8. The system indicates to the user that the student has been deleted		
Priority:	High		
Frequency of Use:	Approximate number depends on how many students are in each classroom. But frequency will be high. On average around 20-25 students added. Roughly the size per classroom.		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	A front end portal exists for the professor to interact with students.		

Use Case 3: Manage User Roles

UC ID and Name:	UC-3: Manage User Roles		
Created By:	Ngan Hanh Tran	Date Created:	October 10, 2021
Primary Actor:	Administrator	Secondary Actors:	
Trigger:	The admin wants to create/view/update/delete roles of professors		
Description:	The admin can view, assign new professor roles or update and delete old ones. This feature only appears in the admin interface.		
Preconditions:	PRE-1. The user has administrator permissions.		
Postconditions:	POST-1. The user can create, view, delete, update user roles POST-2. If the user does not make any change, the database should remain the same		
Main Success Scenario:	1. The admin indicates which user they would like to manage (specify by username / email) 2. The system presents the admin form with editable fields for a user 3. The admin fills in desired information 4. The system presents a preview to the admin 5. The admin confirms 6. The system indicates that a user has been successfully edited 7. The system sends an email to the user with an update of their account status		
Extensions:	1a1. if the admin wishes to create a new user, they specify the email of the new user 1a2. the system presents the user a blank user form 1a4. the admin fills in the user information 1a3. the system sends the new user an email with login instructions and temporary password (end use case) 2a1. the system presents the option to delete a user 2a2. the admin selects the option to delete the user and types their password (to ensure no accidents) (continue to 5)		
Priority:	High		
Frequency of Use:	Essential use case for admins. Initially there will be a few admins but the admins need to assign new professor roles for new professors using the platform.		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	A front end portal exists for the admins to interact with professors.		

Use Case 4: Manage Contest

UC ID and Name:	UC-4: Mange Contest		
Created By:	Ngan Hanh Tran	Date Created:	October 10, 2021
Primary Actor:	Professor	Secondary Actors:	Admin
Trigger:	The professor wants to create/view/update/delete contests		
Description:	The professor can view all contests of each classroom, create new contests for each classroom, update existing contest and delete old contests.		
Preconditions:	PRE-1. The user has professor permissions and can see his/her classrooms		
Postconditions:	POST-1. The user can create, view, delete, update contests of each classroom POST-2. If the user does not make any change, the database should remain the same		
Main Success Scenario:	1. The professor indicates he/she would like to create a new contest for a classroom 2. The system present professor with a form to fill in 3. The professor fills in desired information 4. The system presents the professor a way to upload files. a. solution.java - full implementation of practice solution b. boilerplate.java - full skeleton for the practice solution c. tests.java - JUnit test cases for practice problem d. description.json - json declaration of contest or problem, description text and hints text 5. The professor uploads the required files. 6. The system will compile solution.java and test.java to verify a valid practice problem 7. The system will report to the professor that a practice problem has been successfully added		
Extensions:	1a1. The professor indicates he/she would like to delete a contest 1a2. The system asks the professor to type in the password 1a3. The professor types in the correct password. 1a3a. The professor types in incorrect password 1a3b. The use case returns to 1a2. No change in the database 1a4. The system deletes the chosen contest from the database 1a5. The system notices the professor that the chosen contest has been deleted successfully 1a1. The professor indicates he/she would like to update an existing contest 1a2. The system present professor with a form to fill in 1a3. The professor fills in desired information 1a4. The system will report to the professor that a practice problem has been successfully edited 4a. If any of the files are missing or invalid, the system will not attempt to compile 4b. If the uploaded files fail to compile, the system will report the failure to the professor and echo the compilation stack trace. (return to step 3)		
Priority:	High		
Frequency of Use:	Essential use case for professors. This use cases will be used by professors many times a week to create new contests for students to practice.		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	A front end portal exists for the professors to interact with.		

Use Case 5: Manage Discussion Board

UC ID and Name:	UC-1: Manage Discussion Board		
Created By:	Kate Brayshaw	Date Created:	October 10, 2021
Primary Actor:	Professor	Secondary Actor:	Admin
Trigger:	A professor/admin wants to remove a discussion board		
Description:	The professor/admin can remove a discussion board from a virtual classroom. The professor can specify which post to remove and will be prompted upon completion of deletion.		
Preconditions:	PRE-1. The professor owns at least 1 virtual classroom		
Postconditions:	POST-1. The professor is able to remove posts from the discussion board		
Main Success Scenario:	1. The professor selects the discussion board page on a virtual classroom 2. The professor selects the post they would like to remove 3. The system prompts the professor a dialogue box that prompts them to give a reason for removing the message (possible reasons include, inappropriate language, revealing a solution, spam, etc. with option to echo reason to student) 4. The system deletes the message and reports success to the professor		
Extensions:	2a1. the professor selects a student from the discussion board members page 2a2. the system prompts the professor a dialogue box that prompts them to give a reason for removing the student (possible reasons include, inappropriate language, revealing a solution, spam, etc. with option to echo reason to student) 2a3. the system removes the user from the discussion board and reports success to the professor		
Priority:	High		
Frequency of Use:	Important use case for professors. This will be used continuously. Approximate frequency is not known but assumed high.		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	A front end portal exists for the professor to interact with students discussion boards.		

Use Case 6: Manage Student Participation

UC ID and Name:	UC-6: Manage Student Participation		
Created By:	Ngan Hanh Tran	Date Created:	October 10, 2021
Primary Actor:	Professor	Secondary Actors:	Admin
Trigger:	The professors wants to create/view/update/delete students		
Description:	The professor can view students in his/her classroom/ Update students and delete existing ones. This feature will only appears in the professor interface. A student account must be assigned to a classroom. A student account cannot be created without a classroom.		
Preconditions:	PRE-1. The user has professor permissions. The user already has an existing classroom.		
Postconditions:	POST-1. The user can create, view, delete, update students POST-2. If the user does not make any change, the database should remain the same		
Main Success Scenario:	1. The professor is in a classroom and indicates that he/she wants to create a new student. 2. The system presents the professor form with editable fields for a new student. 3. The professor fills in desired information 4. The system presents a preview to the professor 5. The professor confirms		

	6. The system indicates that a student has been successfully added 7. The system sends an email to the new student with a link to set password and log in.
Extensions:	1a1. If the professor wishes to edit an existing student, he/she specifies the student name or email 1a2. The system presents the professor with a form to edit 1a4. The professor fills in the form 1a3. The system indicates that the student has been successfully edited 1b1. The professor is in a classroom and indicates that he/she wants to delete a student. 1b2. The professor specifies the student name or email 1b3. The system verifies the professor password 1b4. The professor types in a correct password 1b4a. The professor types in an incorrect password. 1b4b. The use case returns to step 1b3. No change made in the database. 1b5. The system deletes the chosen student from the database and notices the professor that the student has been deleted successfully from the classroom. 1c1. The professor indicates that he/she wants to see all the students in an existing classroom 1c2. The system provides the professor with a list of students of that chosen classroom
Priority:	High
Frequency of Use:	Essential use case for professors. This is the main feature that will be used frequently many times a day by each professor.
Business Rules:	N/A
Other Information:	N/A
Assumptions:	A front end portal exists for the professors to interact with students.

Use Case 7: Do Practice (Contest)

UC ID and Name:	UC-7: Do practice (Contest)		
Created By:	Kate Brayshaw	Date Created:	October 10th, 2021
Primary Actor:	Student	Secondary Actor:	
Trigger:	The students participate in the coding contest.		
Description:	The students will have access to the virtual classrooms where they will be allowed to participate in coding contests. This will be accomplished through submitting their own code into the designated submission box. After submission the student will receive a grade.		
Preconditions:	PRE-1. The student account exists in at least 1 virtual classroom and that classroom has at least 1 entry in its problem set.		
Postconditions:	POST-1. The student will be able to upload and test code solutions for practice problems.		
Main Success Scenario:	1. the student accesses the Virtual Classroom testing set page 2. the system will serve the student a web text editor auto filled with “boilerplate.java” (mentioned in Manage Practice Use Case) 3. the student will either copy and paste a solution from their own editor or write code in the browser to solve the problem 4. the student will submit their code to be built		

	5. the system will attempt to compile the student's code 6. the system will report to the user an "in progress" status 7. when the system compiles successfully, it will echo a "success" status to the user 8. the system will report a "success" status on the practice problem instance for the student 9. the system will increment the students "score"
Extensions:	7a1. if the system fails to compile student code, the stack trace from the compiler should be echoed back to the user 7a2. a "failed" status should be reported to the user (return to 2; replace "boilerplate.java" with "student_code.java")
Priority:	High
Frequency of Use:	This use case will be extremely important for students and will be used regularly.
Business Rules:	N/A
Other Information:	N/A
Assumptions:	A front end portal exists for the student to interact with.

Use Case 8: View Results in Practice Contest

UC ID and Name:	UC-1: View Results in Practice Contest		
Created By:	Jacob Hollis	Date Created:	October 10th, 2021
Primary Actor:	Professor	Secondary Actor:	Student
Trigger:	The Professor wants to view Students grades in practice contest.		
Description:	A professor would like to view a student's list of practice / contest results		
Preconditions:	PRE-1. The professor must own at least 1 virtual classroom PRE-2. The classroom must contain at least 1 student PRE-3. The classroom problem set contains at least 1 practice / contest		
Postconditions:	POST-1. The professor can view students practice / contest results		
Main Success Scenario:	1. The professor goes to the practice problem interface 2. The professor indicates which problem they would like to view 3. The system presents the professor with a list of students corresponding with a status of "complete", "in progress", or "did not attempt"		
Extensions:	1. The student wishes to view past practice / contest results a. the student goes to the practice problem interface b. the student indicates which practice / contest they would like to view c. the problem will list a status of "complete", "in progress" or "did not attempt"		
Priority:	Medium		
Frequency of Use:	The use case is necessary for professors and students. all users of the platform		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	A front-end portal will be available for the professor and the student to view practice problems.		

Use Case 9: Discuss in Discussion Board

UC ID and Name:	UC-1: Discuss in Discussion Board		
Created By:	Nithesh Bonugu	Date Created:	October 10th, 2021
Primary Actor:	Student, Professor	Secondary Actor:	Admin
Trigger:	The Professor/Student wants to post a question/comment in the discussion board		
Description:	The Professor/Student can make comments/ask questions in the discussion board to keep a conversation going about a specific topic.		
Preconditions:	PRE-1. The user with professor permissions views the interface of the discussion, and the student also views the same interface but with different permissions.		
Postconditions:	POST-1. The user can post a comment/question/response/reply in the discussion board.		
Main Success Scenario:	4. The user goes to the discussion board interface. 5. He/She indicates that they want to make a post in the discussion board. 6. The system presents the user with a window to type in their query/response. 7. The user inputs their comment. 8. The system asks the: 5a. User with professor permission to read and confirm their message to be sent. 5b. User with student permission if they would like an alias or their actual name to be displayed when their response is posted. 5c. The user with the student permission inputs their comment. 9. The system displays that the comment has been posted.		
Extensions:	2. The user with student permission tried to post some code in the discussion board. 1a. The system shows an error message. 1b. The error message says that content of code type cannot be posted in the discussion board. 1c. The system prompts the user to change their post to normal text type. 3. The user with student permission tries to access the discussion post during a contest session. 2a. The system will display a prompt saying that there is a contest session going on, and that they can access it when the contest session is terminated by the user with professor permissions.		
Priority:	Medium		
Frequency of Use:	Use cases are important. It will be frequently used but not as frequent as the *Do Practice* use case.		
Business Rules:	N/A		
Other Information:	N/A		
Assumptions:	A front-end portal will be available for the professor and the student to make their comments.		

Use Case 10: Login

UC ID and Name:	UC-10: Login		
Created By:	Ngan Hanh Tran	Date Created:	October 10, 2021
Primary Actor:	All users	Secondary Actors:	
Trigger:	Non-registered students, registered students, invalid users, professors and admins want to log in.		
Description:	Non-registered students can log in and create a password. Invalid users cannot log in/ Registered students can log in with a correct password. Admins and professors can log in with a correct password. Successfully logged in user will receive a notification through registered email.		

Preconditions:	PRE-1. The non-registered students use a link sent from the professor to register or the professor already added the students' emails to an existing classroom in the site.
Postconditions:	POST-1. The user can log in, register, or cannot log in. POST-2. If the user can log in successfully, they will see user-interface according to their roles POST-3. If the user can register, they will be able to log in after registering. POST-4. If they cannot log in, they remain in the log in page.
Main Success Scenario:	1. The user logs in with email and password. 2. The system verifies that the email does exist in the system and the password matches 3. The user successfully logs in and sees the user interface that matches their roles. 4. The system sends an email to notify the user that there is a log in.
Extensions:	1a: The user typed an email that is not in the system. 1a1. The system shows a warning that the user does not belong to any classroom in the site. 1a2. The use case returns to step 1 and continues. 1b: The user typed an email that is in the system, but the password is not matched. 1b1. The system shows a warning that email or password is not correct 1b2. The user request a Forgot Password reset 1b2a. The use case returns to step 1 and continue 1b3. The system will request the user to type in an email 1b4. The user types in an email that exists in the system 1b5. The system sends an email with a reset password link 1b6. The user types in new password 1b7. The system accepts the new password and updates it on the database. 1b8. The user is redirected to the login page 1b9. The use case returns to step 1 and continue
Priority:	High
Frequency of Use:	Extremely important for all users.
Business Rules:	N/A
Other Information:	N/A
Assumptions:	A front end form exists for all users to type an email and password.