TCU Computer Science

Project Pulse
Use Cases

Version <1.0>

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

Use Case List 6
Use Case 1: The Admin creates a rubric 7
Use Case 2: The Admin finds senior design sections 9
Use Case 3: The Admin views a senior design section 10
Use Case 4: The Admin creates a senior design section 11
Use Case 5: The Admin edits a senior design section 13
Use Case 6: The Admin sets up active weeks for a senior design section 14
Use Case 7: The Admin/Instructor finds senior design teams 15
Use Case 8: The Admin/Instructor views a senior design team 17
Use Case 9: The Admin creates a senior design team 18
Use Case 10: The Admin edits a senior design team 20
Use Case 11: The Admin invites students to join a senior design section 21
Use Case 12: The Admin assigns students to senior design teams 23
Use Case 13: The Admin removes a student from a senior design team 24
Use Case 14: The Admin deletes a senior design team 25
Use Case 15: The Admin/Instructor finds students 26
Use Case 16: The Admin/Instructor views a student 27
Use Case 17: The Admin deletes a student 28
Use Case 18: The Admin invites instructors to register an account 29
Use Case 19: The Admin assigns instructors to senior design teams 31
Use Case 20: The Admin removes an instructor from a senior design team 32
Use Case 21: The Admin finds instructors 33
Use Case 22: The Admin views an instructor 34
Use Case 23: The Admin deactivate an instructor 35
Use Case 24: The Admin reactivate an instructor 36
Use Case 25: The Student sets up a student account 37
Use Case 26: The Student edits an account 38
Use Case 27: The Student manages activities in a Weekly Activity Report (WAR) 39
Use Case 28: The Student submits a peer evaluation for the previous week 41
Use Case 29: The Student views her own peer evaluation report 43
Use Case 30: The Instructor sets up an instructor account 45
Use Case 31: The Instructor generates a peer evaluation report of the entire senior design section 47
Use Case 32: The Instructor/Student generates a WAR report of a senior design team 49
Use Case 33: The Instructor generates a peer evaluation report of a student 51
Use Case 34: The Instructor generates a WAR report of the student 53
Business Rules 55

Use Cases

Use Case ID and Name
Give each use case a unique integer sequence number identifier. State a concise name for the use case that indicates the value the use case would provide to some user. Begin with an action verb, followed by an object.

Author and Date Created
Enter the name of the person who initially wrote this use case and the date it was written.

Primary and Secondary Actors
An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the primary actor that will be initiating this use case and any other secondary actors who will participate in completing execution of the use case.

Trigger
Identify the business event, system event, or user action that initiates the use case. This trigger alerts the system that it should begin testing the preconditions for the use case so it can judge whether to proceed with execution.

Description
Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

Preconditions
List any activities that must take place, or any conditions that must be true, before the use case can be started. The system must be able to test each precondition. Number each precondition. Example: PRE-1: User's identity has been authenticated.

Postconditions
Describe the state of the system at the successful conclusion of the use case execution. Label each postcondition in the form POST-X, where X is a sequence number. Example: POST-1: Price of item in the database has been updated with the new value.

Main Success Scenario/Normal Flow
Provide a description of the user actions and corresponding system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. Show a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered "X.0", where "X" is the Use Case ID.

Extensions:

- ***Alternative Flows***
  Document other successful usage scenarios that can take place within this use case. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form "X.Y", where "X" is the Use Case ID and Y is a sequence number for the alternative flow. For example, "5.3" would indicate the third alternative flow for use case number 5. Indicate where each alternative flow would branch off from the normal flow, and if pertinent, where it would rejoin the normal flow.
- ***Exceptions***

Describe any anticipated error conditions that could occur during execution of the use case and how the system is to respond to those conditions. Number each alternative flow in the form "X.Y.EZ", where "X" is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, "E" indicates an exception, and "Z" is a sequence number for the exceptions. For example "5.0.E2" would indicate the second exception for the normal flow for use case number 5. Indicate where in the normal (or an alternative) flow each exception could occur.

Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. Use the same priority scheme as that used for the functional requirements.

Frequency of Use

Estimate the number of times this use case will be performed per some appropriate unit of time. This gives an early indicator of throughput, concurrent usage loads, and transaction capacity.

Business Rules

List any business rules that influence this use case. Don't include the business rule text here, just its identifier so the reader can find it in another repository when needed.

Associated Information

Identify any additional requirements, such as quality attributes, for the use case that may need to be addressed during design or implementation. Also list any associated functional requirements that aren't a direct part of the use case flows but which a developer needs to know about. Describe what should happen if the use case execution fails for some unanticipated or systemic reason (e.g., loss of network connectivity, timeout). If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception.

Assumptions

List any assumptions that were made regarding this use case or how it might execute.

Use Case List

Primary Actor	Use Cases
Admin	UC-1: Create a rubric UC-2: Find senior design sections UC-3: View a senior design section UC-4: Create a senior design section UC-5: Edit a senior design section UC-6: Set up active weeks for a senior design section UC-7: Find senior design teams UC-8: View a senior design team UC-9: Create a senior design team UC-10: Edit a senior design team UC-11: Invite students to join a senior design section UC-12: Assign students to senior design teams UC-13: Remove a student from a senior design team UC-14: Delete a senior design team UC-15: Find students UC-16: View a student UC-17: Delete a student UC-18: Invite instructors to register an account UC-19: Assign instructors to senior design teams UC-20: Remove an instructor from a senior design team UC-21: Find instructors UC-22: View an instructor UC-23: Deactivate an instructor UC-24: Reactivate an instructor
Student	UC-25: Set up a student account UC-26: Edit an account UC-27: Manage activities in a weekly activity report UC-28: Submit a peer evaluation for the previous week UC-29: View her own peer evaluation report
Instructor	UC-30: Set up an instructor account UC-31: Generate a peer evaluation report of the entire senior design section UC-32: Generate a WAR report of a senior design team UC-33: Generate a peer evaluation report of a student UC-34: Generate a WAR report of the student

Use Case 1: The Admin creates a rubric

UC ID and Name:	UC-1: Create a rubric		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to create a new rubric.		
Description:	The Admin wants to create a new rubric, so that the students can use it for assessing peer performance effectively.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. The new rubric is stored in the System.		
Main Success Scenario:	1. The Admin indicates to create a new rubric. 2. The System asks the Admin to enter the details of this new rubric according to the "Details" defined in the Associated Information of this use case. 3. The Admin enters the details of this new rubric and confirms that she has finished. 4. The System validates the Admin's inputs according to the "Details" defined in the Associated Information of this use case. 5. The System displays the details of the new rubric and asks the Admin to confirm the creation. 6. The Admin either confirms the creation (continues the normal flow) or chooses to modify the details (return to step 3). 7. The System saves the new rubric and informs the Admin that this rubric has been created. 8. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4a1. The System alerts the Admin that an input validation rule is violated and displays the nature and location of the error. 4a2. The Admin corrects the mistake and returns to step 4 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Details: ● Rubric name (must be unique): E.g., Peer Eval Rubric v1 ● Several criteria: each criterion has a name, a description, and a max score (must be positive and can be a decimal number). For example: Criterion 1: Criterion: Quality of work Description: How do you rate the quality of this teammate's work? (1-10) Max score: 10 Criterion 2: Criterion: Productivity Description: How productive is this teammate? (1-10) Max score: 10 Criterion 3: Criterion: Initiative Description: How proactive is this teammate? (1-10) Max score: 10		

	Criterion 4: Criterion: Courtesy Description: Does this teammate treat others with respect? (1-10) Max score: 10 Criterion 5: Criterion: Open-mindedness Description: How well does this teammate handle criticism of their work? (1-10) Max score: 10 Criterion 6: Criterion: Engagement in meetings Description: How is this teammate's performance during meetings? (1-10) Max score: 10 The Admin shall be able to cancel the use case at any time prior to submitting it.
Related Use Cases	
Assumptions:	
Open Issues:	

Use Case 2: The Admin finds senior design sections

UC ID and Name:	UC-1: Find senior design sections		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to find senior design sections.		
Description:	The Admin wants to find senior design sections which match specific criteria, so that she can decide what to do next.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. A list of matching senior design sections is returned and displayed to the Admin. It is possible that the list is empty.		
Main Success Scenario:	1. The Admin indicates to find senior design sections. 2. The System asks the Admin to enter search values according to the "Search criteria" defined in the Associated Information of this use case. 3. The Admin enters one or more search values and confirms that she has finished entering. 4. The System finds all senior design sections that match the provided search criteria values. 5. The System displays the matching senior design sections according to the "Search results display strategy" and the "Sort criteria" defined in the Associated Information of this use case. 6. Use case ends.		
Extensions:	**4a. No matching senior design sections are found:** 4a1. The System alerts the Admin that no matching senior design sections are found. 4a2. The Admin either chooses to UC-3: Create a senior design section or chooses to terminate the use case or chooses to return to step 2 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, average of 5 usages per week.		
Business Rules:			
Associated Information:	Search criteria (aka search fields, search attributes/properties, search details, searchable qualities):		

Search property name	Data type	Validation rule	Security/access concerns	Reference to glossary
Section name	String	Optional		

Search results display strategy (specify which properties to display for each matching senior design section):
- Section name, team names

Sort criteria:
- Section name in descending order, team names are in ascending order

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 3: The Admin views a senior design section

UC ID and Name:	UC-2: View a senior design section		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to view the details of a senior design section.		
Description:	The Admin wants to view the details of a senior design section, so that she can get a better idea of the senior design section.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. The details of the specified senior design section are displayed to the Admin.		
Main Success Scenario:	1. The Admin indicates to view the details of a senior design section. 2. The Admin finds a list of senior design sections through UC-1: Find senior design sections. 3. The Admin views the list and chooses to view the details of one specific senior design section. 4. The System retrieves and displays the details of this senior design section according to the "Details" defined in the Associated Information and the "Security/access concerns" defined in the Business Rules of this use case. 5. The Admin views the details of this senior design section. 6. Use case ends.		
Extensions:			
Priority:	High		
Frequency of Use:	1 user, average of 5 usages per week.		
Business Rules:			
Associated Information:	Details:		

Property name	Data type	Editability	Security/access concerns	Reference to glossary
Section name	String			
startDate	String			
endDate	String			
Teams, team members, and instructors				
Instructors not assigned to a team				
Students not assigned to a team				
Rubric used				

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 4: The Admin creates a senior design section

UC ID and Name:	UC-3: Create a senior design section		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to create a new senior design section.		
Description:	The Admin wants to create a new senior design section, so that she can invite students to join the section.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. The new senior design section is stored in the System.		
Main Success Scenario:	1. The Admin indicates to create a new senior design section. 2. The System asks the Admin to enter the details of this new senior design section according to the "Details" defined in the Associated Information of this use case. 3. The Admin enters the details of this new senior design section and confirms that she has finished. 4. The System asks the Admin to choose a rubric for the peer evaluation. 5. The Admin selects an existing rubric. 6. The System displays the criteria of the rubric. 7. The Admin confirms the usage of this rubric. 8. The System validates the Admin's inputs according to the "Details" defined in the Associated Information of this use case. 9. The System validates that the creation of the new senior design section will not duplicate any existing senior design section according to the "Duplication detection rules" defined in the Associated Information of this use case. 10. The System displays the details of the new senior design section and asks the Admin to confirm the creation. 11. The Admin either confirms the creation (continues the normal flow) or chooses to modify the details (return to step 3). 12. The System saves the new senior design section and informs the Admin that this senior design section has been created. 13. Use case ends.		
Extensions:	**5a. Rubric does not exist or the Admin wants to create a new rubric:** 5a1. The Admin chooses to UC-6: Create a rubric. 5a2. Returns to step 6 of the normal flow. **7a. The Admin indicates to edit some criteria in the rubric:** 7a1. The Admin edits the name, description, and the max score of a criterion. See the Associated Information for more information. 7a2. The System saves the changes. 7a3. Returns to step 6 of the normal flow. **8a. Input validation rule violation:** 8a1. The System alerts the Admin that an input validation rule is violated and displays the nature and location of the error. 8a2. The Admin corrects the mistake and returns to step 8 of the normal flow. **9a. The System finds possible duplicates from the existing senior design sections:** 9a1. The System alerts the Admin that the senior design section she is trying to create already exists in the System. 9a2. The Admin either chooses to correct the mistake and return to step 8 of the normal flow or chooses to terminate the use case.		
Priority:	High		
Frequency of Use:	1 user, 1 usage per year.		

Business Rules:	
Associated Information:	Details: ● Section name: E.g., Section 2023-2024 ● Start and end date of the section: E.g., 08/21/2023 and 05/01/2024 Editing a rubric: When the Admin edits an existing rubric, behind the scenes, the System shall first duplicate the existing rubric and then let the Admin edit it. In other words, a new rubric is created. Duplication detection rules: ● Section name is used as the unique identifier for a section. The Admin shall be able to cancel the use case at any time prior to submitting it.
Related Use Cases	The Admin may first choose to UC-1: Find senior design sections but cannot find any, then decide to create one.
Assumptions:	
Open Issues:	

Use Case 5: The Admin edits a senior design section

UC ID and Name:	UC-4: Edit a senior design section		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to change the details of an existing senior design section.		
Description:	The Admin wants to change the details of an existing senior design section, so that the section information is correct and up-to-date.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. Changes made to the senior design section are stored in the System.		
Main Success Scenario:	1. The Admin indicates to change the details of an existing senior design section. 2. The Admin views the details of this senior design section through UC-2: View a senior design section. 3. The Admin chooses to change the details of this senior design section. 4. The System asks the Admin to make changes to this senior design section where allowed according to the "Details" defined in the Associated Information and the "Security/access concerns" defined in the Business Rules of this use case. 5. The Admin makes changes to this senior design section until she confirms that she has finished changing. 6. The System validates the Admin's changes and alerts warning messages according to the "Details" defined in the Associated Information of this use case. 7. The Admin acknowledges the warnings and chooses to continue. 8. The System displays the updated details of this senior design section and alerts the Admin to confirm the change. 9. The Admin either confirms the change (continues the normal flow) or chooses to continue to change the details (return to step 5). 10. The System saves the changes, carries out the effect of change according to the "Details" defined in the Associated Information of this use case, and informs the Admin that this senior design section has been changed. 11. Use case ends.		
Extensions:	**6a. Input validation rule violation:** 6a1. The System alerts the Admin that an input validation rule is violated and displays the nature and location of the error. 6a2. The Admin corrects the mistake and returns to step 6 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, average of 1 usage per year.		
Business Rules:			
Associated Information:	Details: **Property name** / **Data type** / **Editability** / **Validation rule** / **Effect of change** / **Warning** / **Reference to glossary** Section name / String / Yes Start and end date of the section / / Yes Rubric used / / Yes The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases:			
Assumptions:			
Open Issues:			

Use Case 6: The Admin sets up active weeks for a senior design section

UC ID and Name:	UC-5: Set up active weeks for a senior design section		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to set up active weeks for a senior design section.		
Description:	The Admin wants to set up weeks for a section, so that the senior design students know in which weeks they need to submit WARs and peer evaluations.		
Preconditions:	PRE-1. The Admin is logged into the System. PRE-2. At least one senior design section is created.		
Postconditions:	POST-1. The active weeks for a senior design section are stored in the System.		
Main Success Scenario:	1. The Admin indicates to set up weeks for a senior design section. 2. The System displays all the weeks of this section according to the start and end date of the section. 3. The Admin specifies the weeks during which the students do not need to submit WARs and peer evaluations, and confirms that she has finished. 4. The System displays the active weeks of the section and asks the Admin to confirm the setup. 5. The Admin either confirms the setup (continues the normal flow) or chooses to modify the details (return to step 3). 6. The System saves the active weeks for this section and informs the Admin that this setup has been done. 7. Use case ends.		
Extensions:			
Priority:	High		
Frequency of Use:	1 user, 1 usage per year.		
Business Rules:	BR-2		
Associated Information:	The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases			
Assumptions:			
Open Issues:			

Use Case 7: The Admin/Instructor finds senior design teams

UC ID and Name:	UC-7: Find senior design teams		
Created By:		Date Created:	
Primary Actor:	Admin, Instructor	Secondary Actors:	
Trigger:	The User indicates to find senior design teams.		
Description:	The User wants to find senior design teams which match specific criteria, so that she can decide what to do next.		
Preconditions:	PRE-1. The User is logged into the System.		
Postconditions:	POST-1. A list of matching senior design teams is returned and displayed to the User. It is possible that the list is empty.		
Main Success Scenario:	1. The User indicates to find senior design teams. 2. The System asks the User to enter search values according to the "Search criteria" defined in the Associated Information of this use case. 3. The User enters one or more search values and confirms that she has finished entering. 4. The System finds all senior design teams that match the provided search criteria values. 5. The System displays the matching senior design teams according to the "Search results display strategy" and the "Sort criteria" defined in the Associated Information of this use case. 6. Use case ends.		
Extensions:	**4a. No matching senior design teams are found:** 4a1. The System alerts the User that no matching senior design teams are found. 4a2. The User either chooses to UC-9: Create a senior design team or chooses to terminate the use case or chooses to return to step 2 of the normal flow.		
Priority:	High		
Frequency of Use:	2 users, average of 2 usages per week.		
Business Rules:			
Associated Information:	Search criteria (aka search fields, search attributes/properties, search details, searchable qualities):		

Search property name	Data type	Validation rule	Default value	Reference to glossary
Section Id				
Section name	String	Optional		
Team name	String	Optional		
Instructor		Optional		

Search results display strategy (specify which properties to display for each matching senior design team):

- Team name
- Team description
- Team website URL
- Team members
- Instructors

Sort criteria:

- First, section name in descending order
  - Then, team name in ascending order

Related Use Cases:	
Assumptions:	

Open Issues:	

Use Case 8: The Admin/Instructor views a senior design team

UC ID and Name:	UC-8: View a senior design team		
Created By:		Date Created:	
Primary Actor:	Admin, Instructor	Secondary Actors:	
Trigger:	The User indicates to view the details of a senior design team.		
Description:	The User wants to view the details of a senior design team, so that she can get a better idea of the senior design team.		
Preconditions:	PRE-1. The User is logged into the System.		
Postconditions:	POST-1. The details of the specified senior design team are displayed to the User.		
Main Success Scenario:	1. The User indicates to view the details of a senior design team. 2. The User finds a list of senior design teams through UC-7: Find senior design teams. 3. The User views the list and chooses to view the details of one specific senior design team. 4. The System retrieves and displays the details of this senior design team according to the "Details" defined in the Associated Information and the "Security/access concerns" defined in the Business Rules of this use case. 5. The User views the details of this senior design team. 6. Use case ends.		
Extensions:			
Priority:	High		
Frequency of Use:	2 users, average of 2 usages per week.		
Business Rules:			
Associated Information:	Details:		

Property name	Data type	Editability	Security/access concerns	Reference to glossary
Team name				
Team description				
Team website URL				
Team members				
Instructors				

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 9: The Admin creates a senior design team

UC ID and Name:	UC-9: Create a senior design team		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to create a new senior design team.		
Description:	The Admin wants to create a new team, so that students can be assigned to it.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. The new senior design team is stored in the System.		
Main Success Scenario:	1. The Admin indicates to create a new senior design team for a senior design section. 2. The System asks the Admin to enter the details of this new senior design team according to the "Details" defined in the Associated Information of this use case. 3. The Admin enters the details of this new senior design team and confirms that she has finished. 4. The System validates the Admin's inputs according to the "Details" defined in the Associated Information of this use case. 5. The System validates that the creation of the new senior design team will not duplicate any existing team according to the "Duplication detection rules" defined in the Associated Information of this use case. 6. The System displays the details of the new senior design team and asks the Admin to confirm the creation. 7. The Admin either confirms the creation (continues the normal flow) or chooses to modify the details (return to step 3). 8. The System saves the new senior design team and informs the Admin that this team has been created. 9. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4a1. The System alerts the Admin that an input validation rule is violated and displays the nature and location of the error. 4a2. The Admin corrects the mistake and returns to step 4 of the normal flow. **5a. The System finds possible duplicates from the existing teams:** 5a1. The System alerts the Admin that the senior design team she is trying to create already exists in the System. 5a2. The Admin either chooses to correct the mistake and return to step 4 of the normal flow or chooses to terminate the use case.		
Priority:	High		
Frequency of Use:	1 user, 5-10 usages per year.		
Business Rules:			
Associated Information:	Details: Senior design team name: E.g., Peer Evaluation Tool team Team description Team website URL Duplication detection rules: ● Team name must be unique The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases	The Admin may first choose to UC-7: Find senior design teams but cannot find any, then decide to create one.		
Assumptions:			
Open Issues:			

Use Case 10: The Admin edits a senior design team

UC ID and Name:	UC-10: Edit a senior design team		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to change the details of an existing senior design team.		
Description:	The Admin wants to change the name of an existing senior design team.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. Changes made to the senior design team are stored in the System.		
Main Success Scenario:	1. The Admin indicates to change the details of an existing senior design team. 2. The Admin views the details of this senior design team through UC-8: View a senior design team. 3. The Admin chooses to change the details of this senior design team. 4. The System asks the Admin to make changes to this senior design team where allowed according to the "Details" defined in the Associated Information and the "Security/access concerns" defined in the Business Rules of this use case. 5. The Admin makes changes to this senior design team until she confirms that she has finished changing. 6. The System validates the Admin's changes and alerts warning messages according to the "Details" defined in the Associated Information of this use case. 7. The Admin acknowledges the warnings and chooses to continue. 8. The System displays the updated details of this senior design team and alerts the Admin to confirm the change. 9. The Admin either confirms the change (continues the normal flow) or chooses to continue to change the details (return to step 5). 10. The System saves the changes, carries out the effect of change according to the "Details" defined in the Associated Information of this use case, and informs the Admin that this senior design team has been changed. 11. Use case ends.		
Extensions:	**6a. Team name conflict:** 6a1. The System alerts the Admin that the team name has been used. 6a2. The Admin corrects the mistake and returns to step 6 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, average of 6 usages per year.		
Business Rules:			
Associated Information:	Details: **Property name** / **Data type** / **Editability** / **Validation rule** / **Effect of change** / **Warning** / **Reference to glossary** Team name / String / Yes Team description / / Yes Team website URL / / Yes No two teams can have the same name. The team name must be unique. The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases:			
Assumptions:			
Open Issues:			

Use Case 11: The Admin invites students to join a senior design section

UC ID and Name:	UC-11: Invite students to join a senior design section		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Student
Trigger:	The Admin indicates to invite students to join a senior design section.		
Description:	The Admin wants to send invitation emails to students, so that they can join a senior design section.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. Invitation emails are sent to all the students.		
Main Success Scenario:	1. The Admin indicates to invite students to join a senior design section. 2. The System asks the Admin to provide students' emails. See the Associated Information section for format. 3. The Admin provides student emails and confirms that she has finished. 4. The System validates the Admin's inputs according to the emails format defined in the Associated Information of this use case. 5. The System displays the number of emails. 6. The System displays the email message. See the Associated Information section for the default message. 7. The Admin either confirms to send the invitation (continues the normal flow) or chooses to modify the details (return to step 3). 8. The System sends out an email to each email address. 9. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4a1. The System alerts the Admin that an input validation rule is violated and displays the nature and location of the error. 4a2. The Admin corrects the mistake and returns to step 4 of the normal flow. **6a. The Admin indicates to personalize the default email message:** 6a1. The Admin customizes the email content and confirms the message. 6a2. Returns to step 6 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Email format: emails shall be separated by semicolon and the System shall ignore spaces in between. E.g., Good: john.doe@tcu.edu; f.smith@tcu.edu; tim.johnson@tcu.edu; lily.p.lee@tcu.edu Good: john.doe@tcu.edu;f.smith@tcu.edu Bad: john.doe@tcu.edu; f.smith@tcu.edu; Bad: john.doe@tcu.edu f.smith@tcu.edu Default email message: *Subject: Welcome to The Peer Evaluation Tool - Complete Your Registration* *Hello,* *[Name of the Admin] has invited you to join The Peer Evaluation Tool. To complete your registration, please use the link below:*		

	[Registration link] *If you have any questions or need assistance, feel free to contact [Admin's email] or our team directly.* *Please note: This email is not monitored, so do not reply directly to this message.* *Best regards,* *Peer Evaluation Tool Team* The invitation link shall be unique for each student. The Admin shall be able to cancel the use case at any time prior to submitting it.
Related Use Cases	The student needs to UC-25: Set up a student account after receiving the invitation email.
Assumptions:	
Open Issues:	

Use Case 12: The Admin assigns students to senior design teams

UC ID and Name:	UC-12: Assign students to senior design teams		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Student
Trigger:	The Admin indicates to assign Students to senior design teams.		
Description:	The Admin wants to assign Students to senior design teams, so that Students can start to submit WARs and evaluate teammates every week.		
Preconditions:	PRE-1. Teams are created. PRE-2. Students have set up their accounts. PRE-3. The Admin is logged into the System.		
Postconditions:	POST-1. Every Student is associated with one team.		
Main Success Scenario:	1. The Admin indicates to assign Students to senior design teams. 2. The System displays a list of teams and a list of Students. 3. The Admin selects a team and assigns a group of Students to it. The Admin repeats this step until she confirms that she has finished assigning Students to all the teams. 4. The System displays the team assignment information and asks the Admin to confirm the assignment. 5. The Admin confirms the assignment. 6. The System notifies relevant actors about the assignment according to the "Notification" defined in the Associated Information of this use case. 7. Use case ends.		
Extensions:	**4a. The Admin finds a wrong team assignment:** 4a1. The Admin removes a student from a team, reassign her to a new team, and returns to step 4 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Notification: ● The System notifies the Students about their team assignment. The Admin shall be able to cancel the process at any time prior to submitting it.		
Related Use Cases:			
Assumptions:			
Open Issues:			

Use Case 13: The Admin removes a student from a senior design team

UC ID and Name:	UC-13: Remove a student from a senior design team		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Student
Trigger:	The Admin indicates to remove a Student from a senior design team.		
Description:	The Admin wants to remove a Student from a senior design team, so that this Student can be assigned to a new team.		
Preconditions:	PRE-1. Teams are created. PRE-2. Students have set up their accounts. PRE-3. The Admin is logged into the System. PRE-4. Students have been assigned to teams.		
Postconditions:	POST-1. The Student is removed from a team.		
Main Success Scenario:	1. The Admin indicates to remove a Student from a senior design team. 2. The Admin views the details of the senior design team through UC-8: View a senior design team. 3. The Admin removes a Student from this team. 4. The System displays the new team assignments and asks the Admin to confirm the removal. 5. The Admin confirms the removal. 6. The System notifies relevant actors about the assignment according to the "Notification" defined in the Associated Information of this use case. 7. Use case ends.		
Extensions:	**5a. The Admin finds an wrong team member removal:** 5a1. The Admin corrects the wrong removal and returns to step 4 of the normal flow.		
Priority:	Low		
Frequency of Use:	Rare. 1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Notification: ● The System notifies the Student about her team removal. The Admin shall be able to cancel the process at any time prior to submitting it.		
Related Use Cases:	The Admin may immediately assign the Student to a new team.		
Assumptions:			
Open Issues:			

Use Case 14: The Admin deletes a senior design team

UC ID and Name:	UC-14: Delete a senior design team		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Student, Instructor
Trigger:	The Admin indicates to delete an existing senior design team.		
Description:	The Admin wants to delete an existing senior design team.		
Preconditions:	PRE-1. The Admin is logged into the System. PRE-2. There exists at least one senior design team.		
Postconditions:	POST-1. The senior design team is deleted from the System according to the "Deletion strategy" defined in the Associated Information of this use case.		
Main Success Scenario:	1. The Admin indicates to delete an existing senior design team. 2. The Admin views the details of this senior design team through UC-8: View a senior design team. 3. The Admin chooses to delete this senior design team. 4. The System alerts the Admin of the consequences of this deletion according to the "Data integrity and deletion rules" defined in the Associated Information of this use case, warns the Admin about the deletion, and asks the Admin to confirm. 5. The Admin confirms the deletion. 6. The System deletes the senior design team according to the "Deletion strategy" defined in the Associated Information of this use case and alerts the Admin that this senior design team has been deleted. 7. The System notifies relevant actors about the deletion of the senior design team according to the "Notification" defined in the Associated Information of this use case. 8. Use case ends.		
Extensions:			
Priority:	Low		
Frequency of Use:	Rare. 1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Data integrity and deletion rules: ● If a team already has students or instructors in it, deleting a team will automatically remove students and instructors from this team first. ● If a team already has WARs and peer evaluations, deleting a team will automatically delete the associated WARs and peer evaluations. Deletion strategy: ● Team deletion is a physical delete. In other words, this will permanently remove the team and the associated WARs and peer evaluations from the database (cannot be recovered). Notification: ● Students and instructors of the deleted team shall be notified. The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases:	The Admin may then choose to UC-12: Assign students to senior design teams and UC-19: Assign instructors to senior design teams.		
Assumptions:			
Open Issues:			

Use Case 15: The Admin/Instructor finds students

UC ID and Name:	UC-15: Find students		
Created By:		Date Created:	
Primary Actor:	Admin, Instructor	Secondary Actors:	
Trigger:	The User indicates to find students.		
Description:	The User wants to find students which match specific criteria, so that she can decide what to do next.		
Preconditions:	PRE-1. The User is logged into the System.		
Postconditions:	POST-1. A list of matching students is returned and displayed to the User. It is possible that the list is empty.		
Main Success Scenario:	1. The User indicates to find students. 2. The System asks the User to enter search values according to the "Search criteria" defined in the Associated Information of this use case. 3. The User enters one or more search values and confirms that she has finished entering. 4. The System finds all students that match the provided search criteria values. 5. The System displays the matching students according to the "Search results display strategy" and the "Sort criteria" defined in the Associated Information of this use case. 6. Use case ends.		
Extensions:	**4a. No matching Students are found:** 4a1. The System alerts the User that no matching students are found. 4a2. The User either chooses to UC-11: Invite students to join a senior design section or chooses to terminate the use case or chooses to return to step 2 of the normal flow.		
Priority:	High		
Frequency of Use:	2 users, average of 2 usages per week.		
Business Rules:			
Associated Information:	Search criteria (aka search fields, search attributes/properties, search details, searchable qualities):		

Search property name	Data type	Validation rule	Security/access concerns	Reference to glossary
First name	String	Optional		
Last name	String	Optional		
Email	String	Optional		
Section name	String	Optional		
Team name	String	Optional		
Section Id	Integer	Optional		
Team Id	Integer	Optional		

Search results display strategy (specify which properties to display for each matching senior design section):

- First name, last name, team name

Sort criteria:

- First, section name in descending order
  - Then, student last name in ascending order

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 16: The Admin/Instructor views a student

UC ID and Name:	UC-16: View a student		
Created By:		Date Created:	
Primary Actor:	Admin, Instructor	Secondary Actors:	
Trigger:	The User indicates to view the details of a student.		
Description:	The User wants to view the details of a student, so that she can get a better idea of the student.		
Preconditions:	PRE-1. The User is logged into the System.		
Postconditions:	POST-1. The details of the specified student are displayed to the User.		
Main Success Scenario:	1. The User indicates to view the details of a student. 2. The User finds a list of students through UC-15: Find students. 3. The User views the list and chooses to view the details of one specific student. 4. The System retrieves and displays the details of this student according to the "Details" defined in the Associated Information and the "Security/access concerns" defined in the Business Rules of this use case. 5. The User views the details of this student. 6. Use case ends.		
Extensions:			
Priority:	High		
Frequency of Use:	1 user, average of 10 usages per week.		
Business Rules:			
Associated Information:	Details:		

Property name	Data type	Editability	Security/access concerns	Reference to glossary
First name				
Last name				
Section name				
Team name				
Peer evaluations				
WARs				

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 17: The Admin deletes a student

UC ID and Name:	UC-17: Delete a student		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Student
Trigger:	The Admin indicates to delete a Student.		
Description:	The Admin wants to delete a Student, because a Student may drop out of the senior design section.		
Preconditions:	PRE-1. The Admin is logged into the System. PRE-2. There exists at least one Student in the System.		
Postconditions:	POST-1. The Student is deleted from the System according to the “Deletion strategy” defined in the Associated Information of this use case.		
Main Success Scenario:	1. The Admin indicates to delete a Student . 2. The Admin views the details of this Student through UC-: View a student. 3. The Admin chooses to delete this Student. 4. The System alerts the Admin of the consequences of this deletion according to the “Data integrity and deletion rules” defined in the Associated Information of this use case, warns the Admin about the deletion, and asks the Admin to confirm. 5. The Admin confirms the deletion. 6. The System deletes the Student according to the “Deletion strategy” defined in the Associated Information of this use case and alerts the Admin that this Student has been deleted. 7. Use case ends.		
Extensions:			
Priority:	Low		
Frequency of Use:	Rare. 1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Data integrity and deletion rules: ● If a Student already submits WARs and peer evaluations, deleting a Student will automatically delete the associated WARs and peer evaluations. Deletion strategy: ● Student deletion is a physical delete. In other words, this will permanently remove the Student and the associated WARs and peer evaluations from the database (cannot be recovered). The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases:			
Assumptions:			
Open Issues:			

Use Case 18: The Admin invites instructors to register an account

UC ID and Name:	UC-18: Invite instructors to register an account		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Instructor
Trigger:	The Admin indicates to invite instructors to register an account.		
Description:	The Admin wants to invite instructors to register an account in the System, so that they can help supervise the senior design projects.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. Invitation emails are sent to instructors.		
Main Success Scenario:	1. The Admin indicates to invite instructors to register an account. 2. The System asks the Admin to provide instructors' emails. See the Associated Information section for format. 3. The Admin provides instructor emails and confirms that she has finished. 4. The System validates the Admin's inputs according to the emails format defined in the Associated Information of this use case. 5. The System displays the number of emails. 6. The System displays the email message. See the Associated Information section for the default message. 7. The Admin either confirms to send the invitation (continues the normal flow) or chooses to modify the details (return to step 3). 8. The System sends out an email to each email address. 9. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4a1. The System alerts the Admin that an input validation rule is violated and displays the nature and location of the error. 4a2. The Admin corrects the mistake and returns to step 4 of the normal flow. **6a. The Admin indicates to personalize the default email message:** 6a1. The Admin customizes the email content and confirms the message. 6a2. Returns to step 6 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Email format: emails shall be separated by semicolon and the System shall ignore spaces in between. E.g., Good: john.doe@tcu.edu; f.smith@tcu.edu; tim.johnson@tcu.edu; lily.p.lee@tcu.edu Good: john.doe@tcu.edu;f.smith@tcu.edu Bad: john.doe@tcu.edu; f.smith@tcu.edu; Bad: john.doe@tcu.edu f.smith@tcu.edu Default email message: *Subject: Welcome to The Peer Evaluation Tool - Complete Your Registration* *Hello,* *[Name of the Admin] has invited you to join The Peer Evaluation Tool. To complete your registration, please use the link below:*		

	[Registration link] *If you have any questions or need assistance, feel free to contact [Admin's email] or our team directly.* *Please note: This email is not monitored, so do not reply directly to this message.* *Best regards,* *Peer Evaluation Tool Team* The invitation link shall be unique for each instructor. The Admin shall be able to cancel the use case at any time prior to submitting it.
Related Use Cases	The instructor needs to UC-30: Set up an instructor account after receiving the invitation email.
Assumptions:	
Open Issues:	

Use Case 19: The Admin assigns instructors to senior design teams

UC ID and Name:	UC-19: Assign instructors to senior design teams		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Instructor
Trigger:	The Admin indicates to assign instructors to senior design teams.		
Description:	The Admin wants to assign instructors to senior design teams, so that instructors can start to supervise teams assigned to them.		
Preconditions:	PRE-1. Teams are created. PRE-2. Instructors have set up their accounts. PRE-3. The Admin is logged into the System.		
Postconditions:	POST-1. Instructors are associated with teams.		
Main Success Scenario:	1. The Admin indicates to assign instructors to senior design teams. 2. The System displays a list of teams and a list of instructors . 3. The Admin selects a team and assigns one or more instructors to it. The Admin repeats this step until she confirms that she has finished assigning instructors to all the teams. 4. The System displays the team assignment information and asks the Admin to confirm the assignment. 5. The Admin confirms the assignment. 6. The System notifies relevant actors about the assignment according to the "Notification" defined in the Associated Information of this use case. 7. Use case ends.		
Extensions:	**4a. The Admin finds an wrong team assignment:** 4a1. The Admin removes an instructor from a team, reassign her to a new team, and returns to step 4 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, 1 usage per year.		
Business Rules:	BR-1		
Associated Information:	Notification: ● The System notifies the instructors about their team assignment. The Admin shall be able to cancel the process at any time prior to submitting it.		
Related Use Cases:			
Assumptions:	The instructor must be assigned to the section of the team first. TODO		
Open Issues:			

Use Case 20: The Admin removes an instructor from a senior design team

UC ID and Name:	UC-20: Remove an instructor from a senior design team		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	Instructor
Trigger:	The Admin indicates to remove an instructor from a senior design team.		
Description:	The Admin wants to remove an instructor from a senior design team, so that this instructor on longer supervises this team.		
Preconditions:	PRE-1. Teams are created. PRE-2. Instructors have set up their accounts. PRE-3. The Admin is logged into the System. PRE-4. Instructors have been assigned to teams.		
Postconditions:	POST-1. The instructor is removed from a team.		
Main Success Scenario:	1. The Admin indicates to remove an instructor from a senior design team. 2. The Admin views the details of the senior design team through UC-8: View a senior design team. 3. The Admin removes an instructor from this team. 4. The System displays the new team assignments and asks the Admin to confirm the removal. 5. The Admin confirms the removal. 6. The System notifies relevant actors about the assignment according to the "Notification" defined in the Associated Information of this use case. 7. Use case ends.		
Extensions:	**5a. The Admin finds an wrong team member removal:** 5a1. The Admin corrects the wrong removal and returns to step 4 of the normal flow.		
Priority:	Low		
Frequency of Use:	Rare. 1 user, 1 usage per year.		
Business Rules:	BR-1		
Associated Information:	Notification: ● The System notifies the instructor about her team removal. The Admin shall be able to cancel the process at any time prior to submitting it.		
Related Use Cases:	The Admin may immediately assign the instructor to a new team.		
Assumptions:			
Open Issues:			

Use Case 21: The Admin finds instructors

UC ID and Name:	UC-21: Find instructors		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to find instructors.		
Description:	The Admin wants to find instructors which match specific criteria, so that she can decide what to do next.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. A list of matching instructors is returned and displayed to the Admin. It is possible that the list is empty.		
Main Success Scenario:	1. The Admin indicates to find instructors. 2. The System asks the Admin to enter search values according to the "Search criteria" defined in the Associated Information of this use case. 3. The Admin enters one or more search values and confirms that she has finished entering. 4. The System finds all instructors that match the provided search criteria values. 5. The System displays the matching instructors according to the "Search results display strategy" and the "Sort criteria" defined in the Associated Information of this use case. 6. Use case ends.		
Extensions:	**4a. No matching instructors are found:** 4a1. The System alerts the Admin that no matching instructors are found. 4a2. The Admin either chooses to UC-18: Invite instructors to join a senior design section or chooses to terminate the use case or chooses to return to step 2 of the normal flow.		
Priority:	High		
Frequency of Use:	1 user, 3 usages per year.		
Business Rules:			
Associated Information:	Search criteria (aka search fields, search attributes/properties, search details, searchable qualities):		

Search property name	Data type	Validation rule	Security/access concerns	Reference to glossary
First name	String	Optional		
Last name	String	Optional		
Team name	String	Optional		
Status	Active or Deactivated	Optional		

Search results display strategy (specify which properties to display for each matching senior design section):

- First name, last name, team name, status

Sort criteria:

- First, academic year in reverse chronological order
  - Then, instructor last name in ascending order

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 22: The Admin views an instructor

UC ID and Name:	UC-22: View an instructor		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to view the details of an instructor.		
Description:	The Admin wants to view the details of an instructor, so that she can get a better idea of the instructor.		
Preconditions:	PRE-1. The Admin is logged into the System.		
Postconditions:	POST-1. The details of the specified instructor are displayed to the Admin.		
Main Success Scenario:	1. The Admin indicates to view the details of an instructor. 2. The Admin finds a list of instructors through UC-21: Find instructors. 3. The Admin views the list and chooses to view the details of one specific instructor. 4. The System retrieves and displays the details of this instructor according to the "Details" defined in the Associated Information and the "Security/access concerns" defined in the Business Rules of this use case. 5. The Admin views the details of this instructor. 6. Use case ends.		
Extensions:			
Priority:	Medium		
Frequency of Use:	1 user, 5 usages per year.		
Business Rules:			
Associated Information:	Details:		

Property name	Data type	Editability	Security/access concerns	Reference to glossary
First name				
Last name				
Supervised Teams				
Status	Active or Deactivated			

Supervised teams shall be organized by section names.

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 23: The Admin deactivate an instructor

UC ID and Name:	UC-23: Deactivate an instructor		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to deactivate an instructor.		
Description:	The Admin wants to deactivate an instructor, so that this instructor no longer has access to the System.		
Preconditions:	PRE-1. The Admin is logged into the System. PRE-2. There exists at least one instructor in the System.		
Postconditions:	POST-1. The insturctor's account is deactivated.		
Main Success Scenario:	1. The Admin indicates to deactivate an instructor. 2. The Admin views the details of this instructor through UC-22: View an instructor. 3. The Admin chooses to deactivate this instructor and enters a reason. 4. The System alerts the Admin of the consequences of this deactivationdefined in the Associated Information of this use case, warns the Admin about the deactivation, and asks the Admin to confirm. 5. The Admin confirms the deactivation. 6. The System deactivates the instructor and alerts the Admin that this instructor has been deactivated. 7. Use case ends.		
Extensions:			
Priority:	Low		
Frequency of Use:	Rare. 1 user, 1 usage per year.		
Business Rules:			
Associated Information:	Consequence of the deactivation: The instructor will no longer have access to the System. But the instructor's information is kept in the System. Deactivation: ● Deactivation will NOT remove the instructor from the System and the instructor's account can be recovered in the future. The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases:			
Assumptions:			
Open Issues:			

Use Case 24: The Admin reactivate an instructor

UC ID and Name:	UC-23: Deactivate an instructor		
Created By:		Date Created:	
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin indicates to reactivate a deactivated instructor.		
Description:	The Admin wants to reactivate a deactivated instructor, so that this instructor has access to the System.		
Preconditions:	PRE-1. The Admin is logged into the System. PRE-2. There exists at least one deactivated instructor in the System.		
Postconditions:	POST-1. The Insturctor's account is reactivated.		
Main Success Scenario:	1. The Admin indicates to reactivate a deactivated instructor. 2. The Admin views the details of this instructor through UC-22: View an instructor. 3. The Admin chooses to reactivate this instructor. 4. The System asks the Admin to confirm. 5. The Admin confirms the reactivation. 6. The System reactivates the instructor and notifies this instructor that her account has been reactivated. 7. Use case ends.		
Extensions:			
Priority:	Low		
Frequency of Use:	Rare. 1 user, 1 usage per year.		
Business Rules:			
Associated Information:	The Admin shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases:			
Assumptions:			
Open Issues:			

Use Case 25: The Student sets up a student account

UC ID and Name:	UC-25: Set up a student account		
Created By:		Date Created:	
Primary Actor:	Student	Secondary Actors:	
Trigger:	The Student clicks the registration link in the invitation email.		
Description:	The Student wants to set up an account, so that she can join a senior design section and submit WARs and peer evaluations.		
Preconditions:	PRE-1. An invitation email is sent to the Student.		
Postconditions:	POST-1. The Student account is set up.		
Main Success Scenario:	1. The Student clicks the registration link in the invitation email. 2. The System opens a new page and asks the Student to enter the details of this new account according to the "Details" defined in the Associated Information of this use case. 3. The Student enters the details of this new account and confirms that she has finished. 4. The System validates the Student's inputs according to the "Details" defined in the Associated Information of this use case. 5. The System displays the details of the new account and asks the Student to confirm the registration. 6. The Student either confirms the registration (continues the normal flow) or chooses to modify the details (return to step 3). 7. The System saves the information about the new account and informs the Student that this account has been created. 8. The System redirects the Student to the login page. 9. Use case ends.		
Extensions:	**2a. The Student has already set up the account:** 2a1. The System alerts the Student that she has already set up her account and shall log in. 2a2. The System redirects the Student to the login page. 2a3. Use case ends. **4a. Input validation rule violation:** 4a1. The System alerts the Student that an input validation rule is violated and displays the nature and location of the error. 4a2. The Student corrects the mistake and returns to step 4 of the normal flow.		
Priority:	High		
Frequency of Use:	Approximately 35-40 users, 1 usage per year.		
Business Rules:			
Associated Information:	Details: ● First name ● Last name ● Email ● Password The Student shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases			
Assumptions:			
Open Issues:			

Use Case 26: The Student edits an account

UC ID and Name:	UC-26: Edit an account		
Created By:		Date Created:	
Primary Actor:	Student	Secondary Actors:	
Trigger:	The Student indicates to change the details of her account.		
Description:	The Student wants to change the details of her account, so that she can correct mistakes made during registration or change the password.		
Preconditions:	PRE-1. The Student is logged into the System.		
Postconditions:	POST-1. Changes made to the account are stored in the System.		
Main Success Scenario:	1. The Student indicates to change the details of her account. 2. The System displays the details of her account. 3. The Student chooses to change the details of this account. 4. The System asks the Student to make changes to this account where allowed according to the "Details" defined in the Associated Information and the "Security/access concerns" defined in the Business Rules of this use case. 5. The Student makes changes to this account until she confirms that she has finished changing. 6. The System validates the Student's changes and alerts warning messages according to the "Details" defined in the Associated Information of this use case. 7. The Student acknowledges the warnings and chooses to continue. 8. The System displays the updated details of this account and alerts the Student to confirm the change. 9. The Student either confirms the change (continues the normal flow) or chooses to continue to change the details (return to step 5). 10. The System saves the changes, carries out the effect of change according to the "Details" defined in the Associated Information of this use case, and informs the Student that this account has been changed. 11. Use case ends.		
Extensions:	**6a. Input validation rule violation:** 6a1. The System alerts the Student that an input validation rule is violated and displays the nature and location of the error. 6a2. The Student corrects the mistake and returns to step 6 of the normal flow.		
Priority:	High		
Frequency of Use:	Rare. Approximately 35-40 users, 1 usage per year.		
Business Rules:			
Associated Information:	Details: **Property name** / **Data type** / **Editability** / **Validation rule** / **Effect of change** / **Warning** / **Reference to glossary** First name / String / Yes Last name / String / Yes Email / String / Yes The Student shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases:			
Assumptions:			
Open Issues:			

Use Case 27: The Student manages activities in a Weekly Activity Report (WAR)

UC ID and Name:	UC-27: Manage activities in a weekly activity report		
Created By:		Date Created:	
Primary Actor:	Student	Secondary Actors:	
Trigger:	The Student indicates to manage activities in a WAR.		
Description:	The Student wants to manage activities in a WAR, so that she can add/edit/delete an activity in a WAR.		
Preconditions:	PRE-1. The Student is logged into the System.		
Postconditions:	POST-1. A new activity is added to the WAR for that week. or POST-2. An existing activity is edited. or POST-3. An existing activity is deleted.		
Main Success Scenario:	1. The Student indicates to manage activities in a WAR. 2. The System asks the Student to select an active week. 3. The Student specifies the active week (cannot select a future active week). 4. The System displays the activities already added by this Student in the WAR and asks the Student to select operations: ● Add a new activity (step 6-10) ● Edit an existing activity (step 11-16) ● Delete an existing activity (step 17-20) 5. The Student selects one out of the three operations. Based on the Student's selection in step 5, the flow goes through either step 6-10, or step 11-16, or step 17-20. 6. The Student selects to add a new activity to this WAR. See the "Details" defined in the Associated Information of this use case. 7. The Student enters the details of the activity and confirms that she has finished. 8. The System validates the Student's inputs according to the "Details" defined in the Associated Information of this use case. 9. The Student either confirms the creation of the activity (continues the normal flow) or chooses to modify the details (return to step 7). 10. The System adds this activity to this WAR and informs the Student that this WAR has been updated. 11. The Student selects to edit an existing activity in this WAR. 12. The Student edits the activity. 13. The System validates the change. 14. The System displays the details of the updated activity and asks the Student to confirm the change. 15. The Student either confirms the change (continues the normal flow) or chooses to modify the details (return to step 12). 16. The System saves the change and informs the Student that this WAR has been updated. 17. The Student selects to delete an existing event. 18. The System asks the Student to confirm the deletion. 19. The Student confirms the deletion. 20. The System deletes the activity and informs the Student that this WAR has been updated. 21. Use case ends.		

Extensions:	**8a. Input validation rule violation:** 8a1. The System alerts the Student that an input validation rule is violated and displays the nature and location of the error. 8a2. The Student corrects the mistake and returns to step 8 of the normal flow. **13a. Input validation rule violation:** 13a1. The System alerts the Student that an input validation rule is violated and displays the nature and location of the error. 13a2. The Student corrects the mistake and returns to step 8 of the normal flow.
Priority:	High
Frequency of Use:	Approximately 35-40 users, average of 3 usages per week.
Business Rules:	
Associated Information:	Details: The Student can add activities to a WAR. For each activity, the Student shall provide the following: ● Activity category: DEVELOPMENT, TESTING, BUGFIX, COMMUNICATION, DOCUMENTATION, DESIGN, PLANNING, LEARNING, DEPLOYMENT, SUPPORT, MISCELLANEOUS ● Activity ● Description ● Planned hours ● Actual hours ● Status: In progress, Under testing, Done. The above properties are editable. The Student shall be able to cancel the use case at any time prior to submitting it.
Related Use Cases	
Assumptions:	
Open Issues:	

Use Case 28: The Student submits a peer evaluation for the previous week

UC ID and Name:	UC-28: Submit a peer evaluation for the previous week		
Created By:		Date Created:	
Primary Actor:	Student	Secondary Actors:	
Trigger:	The Student indicates to submit a peer evaluation for the previous week.		
Description:	The Student wants to submit a peer evaluation for the previous week, so that she can provide feedback and assessment for every team member.		
Preconditions:	PRE-1. The Student is logged into the System.		
Postconditions:	POST-1. The peer evaluation is stored in the System.		
Main Success Scenario:	1. The Student indicates to submit a peer evaluation for the previous week. 2. The System asks the Student to provide peer evaluations for every team member. See the "Details" defined in the Associated Information of this use case. 3. The Student evaluates each team member (self included) and confirms that she has finished. 4. The System validates the Student's inputs according to the "Details" defined in the Associated Information of this use case. 5. The System displays the details of the peer evaluation and asks the Student to confirm the evaluation and submission. Peer evaluations can be edited after submission. 6. The Student either confirms the evaluation and submission (continues the normal flow) or chooses to modify the details (return to step 3). 7. The System saves the peer evaluation and informs the Student that this peer evaluation has been submitted. 8. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4a1. The System alerts the Student that an input validation rule is violated and displays the nature and location of the error. 4a2. The Student corrects the mistake and returns to step 4 of the normal flow.		
Priority:	High		
Frequency of Use:	Approximately 35-40 users, 1 usage per week.		
Business Rules:	BR-3, BR-4		
Associated Information:	Details: Every team member MUST be evaluated. Scores MUST be integers.		

Student	**Name: Quality of work Description: How do you rate the quality of this teammate's work? (1-10)**	**…**	**Public comments**	**Private comments**
John Doe	8	…	…	…
Lily Fisher	10	…	…	…
Tim Smith	9	…	…	…
…				

	Private comments are for the instructor only. Public comments will be sent to the student under assessment. The Student shall be able to cancel the use case at any time prior to submitting it.
Related Use Cases	
Assumptions:	
Open Issues:	

Use Case 29: The Student views her own peer evaluation report

UC ID and Name:	UC-29: View her own peer evaluation report		
Created By:		Date Created:	
Primary Actor:	Student	Secondary Actors:	
Trigger:	The Student indicates to view her own peer evaluation report (on demand).		
Description:	The Student wants to run a peer evaluation report, so that she can better understand how she is assessed by her teammates.		
Preconditions:	PRE-1. The Student is logged into the System.		
Postconditions:	POST-1. The details of the report are returned and displayed to the Student.		
Main Success Scenario:	1. The Student indicates to generate a peer evaluation report. 2. The System asks the Student to provide configurable report generating parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 3. The Student enters the required parameters and confirms that she has finished entering. 4. The System validates the input parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 5. The System generates the peer evaluation report according to the "Report generating algorithm" defined in the Associated Information of this use case and displays to the Student according to the "Report generating parameters" defined in the Associated Information of this use case. 6. The System delivers the generated report according to the specified report disposition in the specified format in the "Report generating parameters" defined in the Associated Information of this use case. 7. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4b1. The System alerts the Student that an input validation rule is violated and displays the nature and location of the error. 4b2. The Student corrects the mistake and returns to step 4 of the normal flow. **5a. No data is returned:** 5a1. The System alerts the Student that no data is available in the generated report. 5a2. The Student either chooses to return to step 3 of the normal flow or chooses to terminate the use case.		
Priority:	High		
Frequency of Use:	Approximately 35-40 users, average of 1 usage per week.		
Business Rules:	BR-5		
Associated Information:	Report generating parameters: ● Active week: Each peer evaluation report is associated with a week. The instructor shall first indicate for which active week she wants to generate a peer evaluation. E.g., "02-12-2024 to 02-18-2024", by default, it shall be the previous week. ● Columns to include: Student name, average rubric criterion scores, public comments, average total grade. See the example below. ● Pagination: Not needed. ● Format of the generated report: HTML.		

An example of the generated report:
Details:

Student	**Name: Quality of work Description: How do you rate the quality of this teammate's work? (1-10)**	**…**	**Public comments**	**Grade**
John Doe	8.5	…	Good work. Need to work harder. …	54/60

Attention, a student shall never see the private comments and the evaulaters.

Report generating algorithm:

For each individual criterion score (e.g., Quality of work), the System shall consider the scores provided by all teammates and compute an average. For the overall grade, see the algorithm in UC-31:Generate a peer evaluation report of the entire senior design section.

Related Use Cases:	UC-31:Generate a peer evaluation report of the entire senior design section
Assumptions:	
Open Issues:	

Use Case 30: The Instructor sets up an instructor account

UC ID and Name:	UC-30: Set up an instructor account		
Created By:		Date Created:	
Primary Actor:	Instructor	Secondary Actors:	
Trigger:	The Instructor clicks the registration link in the invitation email.		
Description:	The Instructor wants to set up an account, so that she can supervise senior design projects in a senior design section.		
Preconditions:	PRE-1. An invitation email is sent to the Instructor.		
Postconditions:	POST-1. The Instructor account is set up.		
Main Success Scenario:	1. The Instructor clicks the registration link in the invitation email. 2. The System opens a new page and asks the Instructor to enter the details of this new account according to the "Details" defined in the Associated Information of this use case. 3. The Instructor enters the details of this new account and confirms that she has finished. 4. The System validates the Instructor's inputs according to the "Details" defined in the Associated Information of this use case. 5. The System displays the details of the new account and asks the Instructor to confirm the registration. 6. The Instructor either confirms the registration (continues the normal flow) or chooses to modify the details (return to step 3). 7. The System saves the information about the new account and informs the Instructor that this account has been created. 8. The System redirects the Instructor to the login page. 9. Use case ends.		
Extensions:	**2a. The Instructor has already set up the account:** 2a1. The System alerts the Instructor that she has already set up her account and shall log in. 2a2. The System redirects the Instructor to the login page. 2a3. Use case ends. **4a. Input validation rule violation:** 4a1. The System alerts the Instructor that an input validation rule is violated and displays the nature and location of the error. 4a2. The Instructor corrects the mistake and returns to step 4 of the normal flow.		
Priority:	High		
Frequency of Use:	Approximately 2 users, average of 1 usage per year.		
Business Rules:			
Associated Information:	Details: ● First name ● Middle initial ● Last name ● Password ● Reenter password: must be the same as password. The Instructor shall be able to cancel the use case at any time prior to submitting it.		
Related Use Cases			
Assumptions:			
Open Issues:			

Use Case 31: The Instructor generates a peer evaluation report of the entire senior design section

UC ID and Name:	UC-31: Generate a peer evaluation report of the entire senior design section		
Created By:		Date Created:	
Primary Actor:	Instructor	Secondary Actors:	
Trigger:	The Instructor indicates to generate a peer evaluation report of the entire senior design section.		
Description:	The Instructor wants to run a peer evaluation report, so that she can better understand students' performance within a team environment.		
Preconditions:	PRE-1. The Instructor is logged into the System.		
Postconditions:	POST-1. The details of the report are returned and displayed to the Instructor.		
Main Success Scenario:	1. The Instructor indicates to generate a peer evaluation report of the entire senior design section. 2. The System asks the Instructor to provide configurable report generating parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 3. The Instructor enters the required parameters and confirms that she has finished entering. 4. The System validates the input parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 5. The System generates the peer evaluation report according to the "Report generating algorithm" defined in the Associated Information of this use case and displays to the Instructor according to the "Report generating parameters" defined in the Associated Information of this use case. 6. The System delivers the generated report according to the specified report disposition in the specified format in the "Report generating parameters" defined in the Associated Information of this use case. 7. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4b1. The System alerts the Instructor that an input validation rule is violated and displays the nature and location of the error. 4b2. The Instructor corrects the mistake and returns to step 4 of the normal flow. **5a. No data is returned:** 5a1. The System alerts the Instructor that no data is available in the generated report. 5a2. The Instructor either chooses to return to step 3 of the normal flow or chooses to terminate the use case.		
Priority:	High		
Frequency of Use:	Approximately 2 users, average of 1 usage per week.		
Business Rules:			
Associated Information:	Report generating parameters: ● Active week: Each peer evaluation report is associated with a week. The instructor shall first indicate for which active week she wants to generate a peer evaluation. E.g., "02-12-2024 to 02-18-2024", by default, it shall be the previous week. ● Columns to include: Student name, grade, comments. See the example below. ● Sorting criteria: by default, sort by last name in ascending order. ● Pagination: Not needed. ● Format of the generated report: HTML.		

An example of the generated report:

Student	**Grade**	**Commented by**	**Public comments**	**Private comments**
John Doe	54/60	Tim Smith	Good work.	Nothing.
		Lily Fisher	Need to work harder.	Dr. Wei, I need to talk more about John.
Lily Fisher	…	…	…	…

The report shall show who did not turn in the peer evaluation for that week.

Report generating algorithm:

How to compute the peer evaluation grade for a student?
Each student receives multiple peer evaluations from her teammates every week. First, obtain the peer evaluations received by a student for that week. For each peer evaluation, compute a total score by adding up the individual criterion scores. Then compute the average of the total scores across the peer evaluations.

For example, John Doe receives two peer evaluations from Tim Smith and Lily Fisher, respectively. Tim Smith gives the following scores based on the rubric: 10, 9, 10, 9, 10, 10. So the total score given by Tim is 58. Lily Fisher gives the following scores based on the rubric: 5, 5, 10, 10, 10, 10. So the total score given by Tim is 50. The grade that John Doe receives that week is (58 + 50) / 2 = 54.

Details of a peer evaluation:
The Instructor may choose to see more details of one student's peer evaluation. For example, this table below shows the scores given by each evaluator in the same team to John Doe.

Evaluator of John Doe	**Name: Quality of work Description: How do you rate the quality of this teammate's work? (1-10)**	**…**	**Public comments**	**Private comments**
John Doe	10	…	…	…
Lily Fisher	6	…	…	…
Tim Smith	9	…	…	…
…				

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 32: The Instructor/Student generates a WAR report of a senior design team

UC ID and Name:	UC-32: Generate a WAR report of a senior design team		
Created By:		Date Created:	
Primary Actor:	Instructor, Student	Secondary Actors:	
Trigger:	The Instructor/Student indicates to generate a WAR report of a senior design team.		
Description:	The User wants to run a WAR report, so that she can better understand how students contribute to the project in a week.		
Preconditions:	PRE-1. The User is logged into the System.		
Postconditions:	POST-1. The details of the report are returned and displayed to the User.		
Main Success Scenario:	1. The User indicates to generate a WAR report. 2. The System asks the User to provide configurable report generating parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 3. The User enters the required parameters and confirms that she has finished entering. 4. The System validates the input parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 5. The System generates the WAR report according to the "Report generating algorithm" defined in the Associated Information of this use case and displays to the User according to the "Report generating parameters" defined in the Associated Information of this use case. 6. The System delivers the generated report according to the specified report disposition in the specified format in the "Report generating parameters" defined in the Associated Information of this use case. 7. Use case ends.		
Extensions:	**4a. Input validation rule violation:** 4b1. The System alerts the User that an input validation rule is violated and displays the nature and location of the error. 4b2. The User corrects the mistake and returns to step 4 of the normal flow. **5a. No data is returned:** 5a1. The System alerts the User that no data is available in the generated report. 5a2. The User either chooses to return to step 3 of the normal flow or chooses to terminate the use case.		
Priority:	High		
Frequency of Use:	Approximately 37 users, average of 1 usage per week.		
Business Rules:			
Associated Information:	Report generating parameters: ● Active week: Each WAR report is associated with a week. The instructor shall first indicate for which active week she wants to generate a WAR. E.g., "02-12-2024 to 02-18-2024", by default, it shall be the previous week. ● Columns to include: Student name, Activity category, Planned activity, Description, Planned hours, Actual hours, Status. See the example below. ● Sorting criteria: by default, sort by last name in ascending order. ● Pagination: Not needed. ● Format of the generated report: HTML.		

An example of the generated report:

Student	Activity category	Planned activity	Description	Planned hours	Actual hours	Status
John Doe	Bug fixing	Activity 1	Fix the login bug……	4	5	Done
	Documentation	Activity 2	Write three new use cases. They are ……	5		In Progress

The report shall show who did not turn in the WAR for that week.

Report generating algorithm: N/A

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 33: The Instructor generates a peer evaluation report of a student

UC ID and Name:	UC-33: Generate a peer evaluation report of a student		
Created By:		Date Created:	
Primary Actor:	Instructor	Secondary Actors:	
Trigger:	The Instructor indicates to generate a peer evaluation report of a student.		
Description:	The Instructor wants to run a peer evaluation report of a student, so that she can better understand this student's performance during a period of time.		
Preconditions:	PRE-1. The Instructor is logged into the System.		
Postconditions:	POST-1. The details of the report are returned and displayed to the Instructor.		
Main Success Scenario:	1. The Instructor indicates to generate a peer evaluation report of a student. 2. The Instructor views the details of this student through UC-16: View a student. 3. The Instructor chooses to generate a peer evaluation report of this student. 4. The System asks the Instructor to provide configurable report generating parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 5. The Instructor enters the required parameters and confirms that she has finished entering. 6. The System validates the input parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 7. The System generates the peer evaluation report according to the "Report generating algorithm" defined in the Associated Information of this use case and displays to the Instructor according to the "Report generating parameters" defined in the Associated Information of this use case. 8. The System delivers the generated report according to the specified report disposition in the specified format in the "Report generating parameters" defined in the Associated Information of this use case. 9. Use case ends.		
Extensions:	**6a. Input validation rule violation:** 6b1. The System alerts the Instructor that an input validation rule is violated and displays the nature and location of the error. 6b2. The Instructor corrects the mistake and returns to step 6 of the normal flow. **7a. No data is returned:** 7a1. The System alerts the Instructor that no data is available in the generated report. 7a2. The Instructor either chooses to return to step 5 of the normal flow or chooses to terminate the use case.		
Priority:	High		
Frequency of Use:	2 users, average of 10 usage per week.		
Business Rules:			
Associated Information:	Report generating parameters: ● Period: Start active week and end active week. ● Columns to include: Week, grade, comments. See the example below. ● Sorting criteria: by default, sort by week in chronological order. ● Pagination: Not needed. ● Format of the generated report: HTML.		

An example of the generated report:

Week	Grade	Commented by	Public comments	Private comments
02-12-2024 - 02-18-2024	54/60	Tim Smith	Good work.	Nothing.
		Lily Fisher	Need to work harder.	Dr. Wei, I need to talk more about John.
02-19-2024 - 02-25-2024	55/60	…	…	…

Report generating algorithm:
Refer to the algorithm defined in UC-31: Generate a peer evaluation report of the entire senior design section

Details of a peer evaluation:
The Instructor may choose to see more details of one student's peer evaluation. Refer to the algorithm defined in UC-31: Generate a peer evaluation report of the entire senior design section.

Related Use Cases:	
Assumptions:	
Open Issues:	

Use Case 34: The Instructor generates a WAR report of the student

UC ID and Name:	UC-34: Generate a WAR report of the student		
Created By:		Date Created:	
Primary Actor:	Instructor	Secondary Actors:	
Trigger:	The Instructor indicates to generate a WAR report of a student.		
Description:	The Instructor wants to run a WAR report of a student, so that she can better understand how this student contributes to the project during a period of time.		
Preconditions:	PRE-1. The Instructor is logged into the System.		
Postconditions:	POST-1. The details of the report are returned and displayed to the Instructor.		
Main Success Scenario:	1. The Instructor indicates to generate a WAR report of a student. 2. The Instructor views the details of this student through UC-16: View a student. 3. The Instructor chooses to generate a WAR report of this student. 4. The System asks the Instructor to provide configurable report generating parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 5. The Instructor enters the required parameters and confirms that she has finished entering. 6. The System validates the input parameters according to the "Report generating parameters" defined in the Associated Information of this use case. 7. The System generates the WAR report according to the "Report generating algorithm" defined in the Associated Information of this use case and displays to the User according to the "Report generating parameters" defined in the Associated Information of this use case. 8. The System delivers the generated report according to the specified report disposition in the specified format in the "Report generating parameters" defined in the Associated Information of this use case. 9. Use case ends.		
Extensions:	**6a. Input validation rule violation:** 6b1. The System alerts the Instructor that an input validation rule is violated and displays the nature and location of the error. 6b2. The Instructor corrects the mistake and returns to step 6 of the normal flow. **7a. No data is returned:** 7a1. The System alerts the Instructor that no data is available in the generated report. 7a2. The Instructor either chooses to return to step 5 of the normal flow or chooses to terminate the use case.		
Priority:	High		
Frequency of Use:	2 users, average of 10 usage per week.		
Business Rules:			
Associated Information:	Report generating parameters: ● Period: Start active week and end active week. ● Columns to include: Activity category, Planned activity, Description, Planned hours, Actual hours, Status. See the example below. ● Sorting criteria: by default, sort by active weeks in chronological order. ● Pagination: Not needed. ● Format of the generated report: HTML.		

An example of the generated report:

Active week: 02-12-2024 to 02-18-2024

Activity category	**Planned activity**	**Description**	**Planned hours**	**Actual hours**	**Status**
Bug fixing	Activity 1	Fix the login bug……	4	5	Done
Documentation	Activity 2	Write three new use cases. They are ……	5		In Progress

Active week: 02-19-2024 to 02-25-2024

Activity category	**Planned activity**	**Description**	**Planned hours**	**Actual hours**	**Status**
New feature dev	Activity 3	Fix the login bug……	10	9	Done
Documentation	Activity 2	Write three new use cases. They are ……	5	10	Done

……

Report generating algorithm: N/A

Related Use Cases:	
Assumptions:	
Open Issues:	

Business Rules

BR-1: Every senior design team must be assigned at least one instructor. It is common that there are two instructors in one team: one is a real TCU instructor, the other one is the client. An instructor can be assigned to multiple teams.
BR-2: For the fall semester, the active weeks are usually from the 5th week to the 15th week. Winter holidays are inactive weeks. For the spring semester, the active weeks are from the 1st week to the 15th week. Students are only allowed to submit peer evaluations during active weeks. However, they can submit weekly activities outside active weeks.
BR-3: Peer evaluation cannot be edited once completed. (TODO)
BR-4: A student can only submit a peer evaluation for the previous week. A student has one week to complete a peer evaluation for the previous week. If a student fails to complete a peer evaluation, she cannot make it up.
BR-5: For the peer evaluation, a student can only see her rubric criterion scores, public comments, and the overall grade.