#### <IT> *Get list of killed processes.*



Figure 9 <IT> Get list of killed processes.

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_IT.01** | | | |
| **Use Case No.** | UC\_IT01 | **Use Case Version** | 2.0 |
| **Use Case Name** | Get list of killed processes. | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * IT   **Summary:**   * *Allows to get list of killed processes.* * *Help staff kill student’s processes real-time.* * *Kill unallowed process when student starts ExanTool up.*   **Goal:**   * *Staff can get list of killed processes to manage student’s killed processes.*   **Triggers:**   * *User send get command to server.*   **Preconditions:**   * *N/A*   **Post Conditions:**   * *Success: Display list of killed processes.* * *Fail: Display error messages.*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *IT goes to the tab “process must kill screen”* |  | | *2* |  | *The system will be displayed list process must kill with information:*   * Process name * Application name * Process path |   **Alternative Scenario:** *N/A*  **Relationship:** *N/A*  **Business Rules:**   * *Staff can get list killed process with following information:*   + *Process name*   + *Application name*   + *Process path* | | | |

Table 14 <IT> Get List Process Must Kill

#### <IT> Add process to list of killed processes.

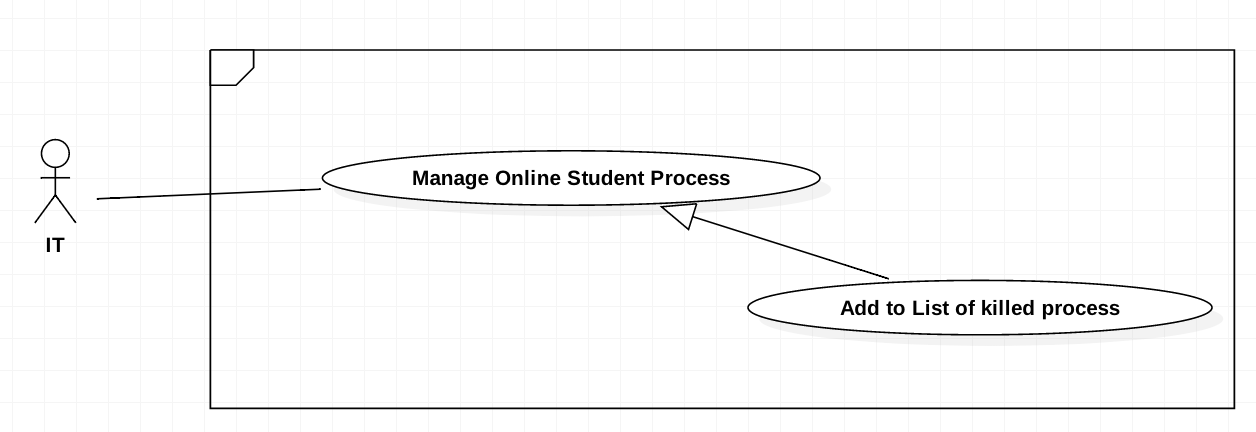


Figure 10 Add process to list of killed processes

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_IT.03** | | | |
| **Use Case No.** | UC\_IT03 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add process to list of killed processes | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * *IT*   **Summary:**   * *When staff forbids a process, staff can add process(es) to list of killed processes.*   **Goal:**   * *Staff can add process(es) to list of killed processes.*   **Trigger:**   * *Staff sends create command to add process(es) to killed processes list.*   **Preconditions:**   * *The process is not existed in killed processes.*   **Post Conditions:**   * *Success: Process is added to list of killed processes.* * *Fail: Missing process’s informations.*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *IT goes to list process real time which show student computer process real time.*  *IT sends get list killed process command.* | *The system will be displayed list process must kill with information:*   * *Process name* * *Application name* * *Process path* | | *2* | *Select the process row which IT want to Add* | *Parse process data to process information format* | | *3* | *IT send add to process must kill command* |  | | *4* |  | *Add new process to processes must killed. System check if process information is missing*  *Process will be addewd to list of killed process.*  *[Exception]* |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | *1* | *Process information is missing* | *Show message to IT client* |   **Relationships:**   * *N/A*   **Business Rules:**   * *Get processes must kill from system (A).* * *The system get current processes of student (B).* * *Get user selected processes from B (C).* * *Push C to A.* | | | |

Table 15 Add Must Be Killed Process to List

#### <IT> Add process to list of exception processes

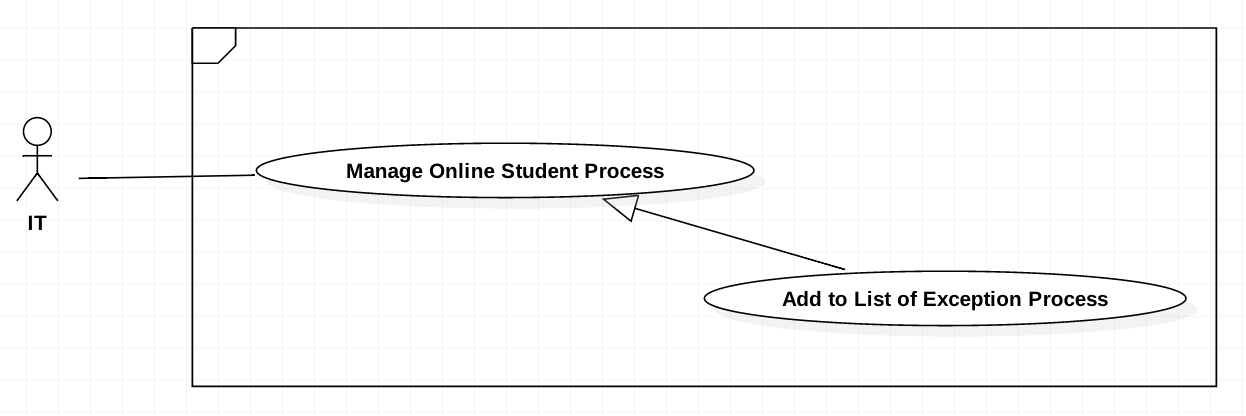


Figure 11 Add process to list of exception processes

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_IT.04** | | | |
| **Use Case No.** | UC\_IT04 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add process to list of exception processes | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * *IT*   ***Summary:***   * *IT add process from real time process to List Exception Process*   ***Goal:***   * *IT can add a new exception process to list of exception processes.*   **Trigger:**   * *IT send create command.*   **Preconditions:**   * *IT scan student computers‘s running processes.*   **Post Conditions:**   * *Success: added new records in list exception processes* * *Fail: Process information is missing*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *IT goes to list process real time which show student computer process real time.* | *The system will be displayed list process must kill with information:*   * *Process Name* * *Application Name* * *Process Path* | | *2* | *Select the process row which IT want to Add* | *Parse process data to process information format* | | *3* | *IT send add to process exception command* |  | | *4* |  | *Add new Process to Process must killed. System check if process information is missing*  *[Exception]* |   **Exceptions:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | *1* | *Process information is existed* | *Notify to IT Client* |   **Relationships:**   * *N/A.*   **Business Rules:**   * *Get exception processes from system (A).* * *The system get current processes of student (B).* * *Get user selected processes from B (C).* * *Exception C from A.* | | | |

Table 16 Add process to list of exception processes

#### <IT> Kill Process

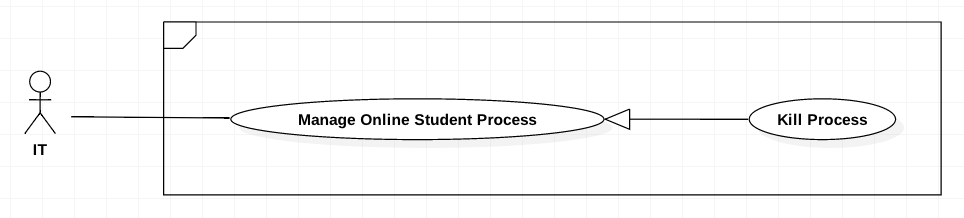


Figure 12 Kill Process

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_IT.05** | | | |
| **Use Case No.** | UC\_IT05 | **Use Case Version** | 2.0 |
| **Use Case Name** | Kill Process | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * *IT*   **Summary:**   * *Allows kill student’s processes real-time.*   **Goal:**   * *Kill student’s processes real-time.*   **Triggers:**   * *IT send kill command to server.*   **Preconditions:**   * *Need to scanned student’s processes.* * *Need to defined cheating processes.*   **Post Conditions:**   * *Success: Process is killed on student’s computer.* * *Fail: Display error messages.*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *IT goes to list process real time which show student computer process real time.* | *The system will be displayed list process must kill with information:*   * *Process Name* * *Application Name* * *Process Path* | | *2* | *Select the process row which IT want to Kill* | *Parse process data to process information format* | | *3* | *IT send Kill Process command* |  | | *4* |  | *Process selected will kill on all student computer if it existed. System check if process information is missing*  *[Exception]* |   **Relationships:**   * *N/A.*   **Business Rules:**   * *The system get current processes of student (A).* * *Get user selected processes from A (B).* * *Send request to kill B from A.* | | | |

Table 17 Kill Process

#### <Leader> Add chapter

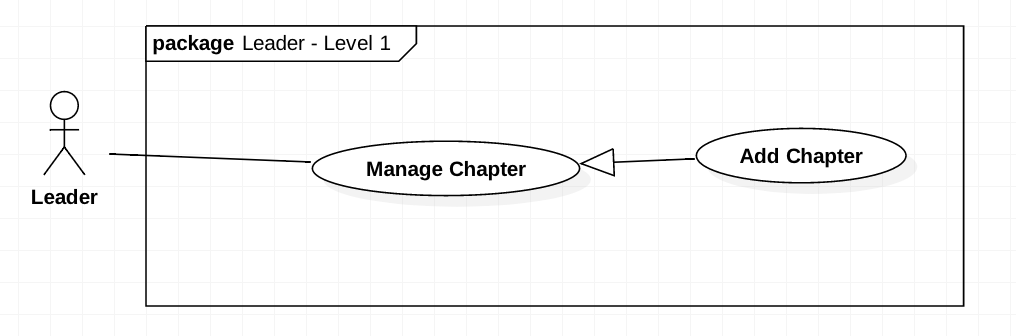


Figure 22 Add chapter

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_ Leader.151** | | | |
| **Use Case No.** | UC\_Leader 15 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Chapter | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * 1. *Leader*   **Summary:**   * 1. *Allow user to add chapters.*   2. *Support for organize question by chapters.*   3. *Support for generate question by chapters.*   **Goal:**   * 1. *Add new chapter to specify course.*   **Triggers:**   * 1. *Send an create command to server.*   **Preconditions:**  *N/A*  **Post Conditions:**   * + - *Success: Chapters will be added and display success message*     - *Fail: Display error messages.*  1. **Main Success Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Leader goes to add chapter view | The system require to fill information:   * + *Chapter*   + *Level*   + *Mark*   + *Chapter Content*   + *Options* | | 2 | Leader send a save command to save the chapters locally |  | | 3 |  | The system show response message and display new list chapter.  [Exception] |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | *1* | *The chapter info is missing* | *Show message to refill form to add new chapter.* |   **Alternative Scenario: N/A**  **Relationships: N/A**  **Business Rules:**   * + - *Get the user input for a new chapter(A).*     - *Get all chapter available base on course code in system(B).*     - *If A not existed in B, then insert A to system.* | | | |

Table 25 Add chapter

#### <Leader> Delete chapter

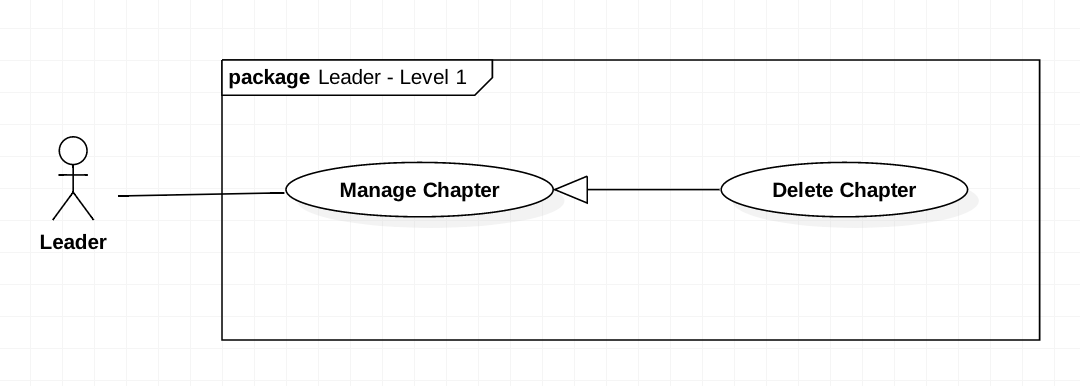


Figure 23 Delete chapter

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_ Leader.16** | | | |
| **Use Case No.** | UC\_Leader16 | **Use Case Version** | 2.0 |
| **Use Case Name** | Delete Chapter | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * + *Leader*   **Summary:**   * + *This use case allows user remove unnecessary chapters.*   + *Support for generate test exam more high quality.*   **Goal:**   * + *Remove the chapter from list chapter of this course*   **Triggers:**   * + *send delete chapter command.*   **Preconditions:**   * + *Login the system with Leader role.*   **Post Conditions:**   * + *Success: Chapters will be added.*   + *Fail: Chapters will not be added and show error message*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  | *Leader goes to list chapter* | *The system will display list chapter with information:*   * + *Chapter*   + *Level*   + *Mark*   + *Chapter Content*   + *Options* | |  | *Leader select chapter want to remove* |  | | *3* | *Leader send a delete chapter command.* | *The system will show a popup notify if that chapter has been deleted successfully* |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships: N/A**  **Business Rules:**   * *Get the user input for a chapter(A).* * *Get all chapter available base on course code in system(B).* * *If A existed in B, then remove A from system.* | | | |

Table 26 Delete chapter

#### <Leader> Edit chapter

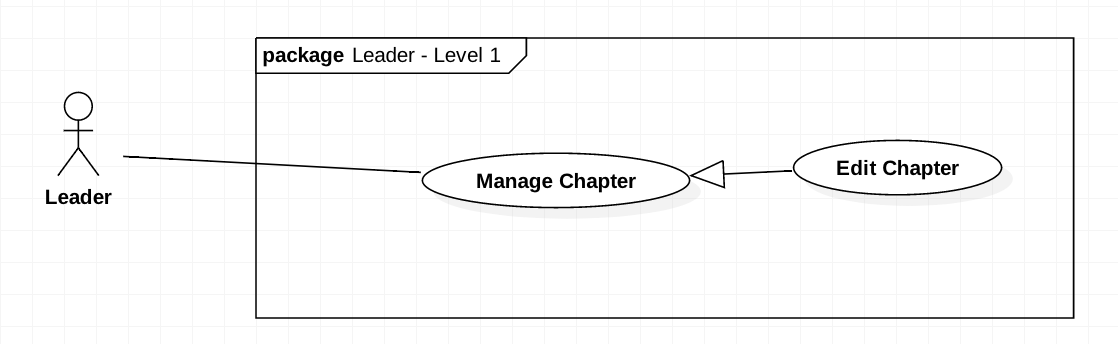


Figure 24 <Leader> Edit chapter

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_ Leader.18** | | | |
| **Use Case No.** | UC\_Leader18 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Chapter | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * + - *Leader*   **Summary:**   * + - *This use case allows user to edit chapter name and chapter order*.     - *Support organize question and course more complete.*   **Goal:**   * + - *Edit chapter with new name.*   **Triggers:**   * + - *User send an edit chapter command.*     - *User input new chapter Name.*   **Preconditions:**   * + - *Login the system with Leader role.*     - *Change chapter to new information (chapter name, chapter order)*   **Post Conditions:**   * + - *Success: Chapters will be edited show success message*     - *Fail: Chapters will not be edit and show message error*  1. **Main Success Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Leader goes to List chapter view* | *The system will display list chapter with information:*   * + *Chapter*   + *Level*   + *Mark*   + *Chapter Content*   + *Options* | | *2* | *Leader fill the form* | *The system requires Leader to input:*   * + *Chapter*   + *Level*   + *Mark*   + *Chapter Content*   + *Options* | | *3* | *Leader send a save command to save the chapters locally* | *[Exception]* | | *4* | *Leader send save command* | *The system will run save change command to save all changed* |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | *1* | *The chapter info is missing* | *Show message to fill full form,* |   **Alternative Scenario: N/A**  **Relationships: Manage Chapter**  **Business Rules:**   * + - *Get the chapter base on chapter’s id (A).*     - *Update A by user input (name, order, …).*     - *Save A to the system.* | | | |

Table 27 <Leader> Edit chapter

#### <Teacher> Search Question

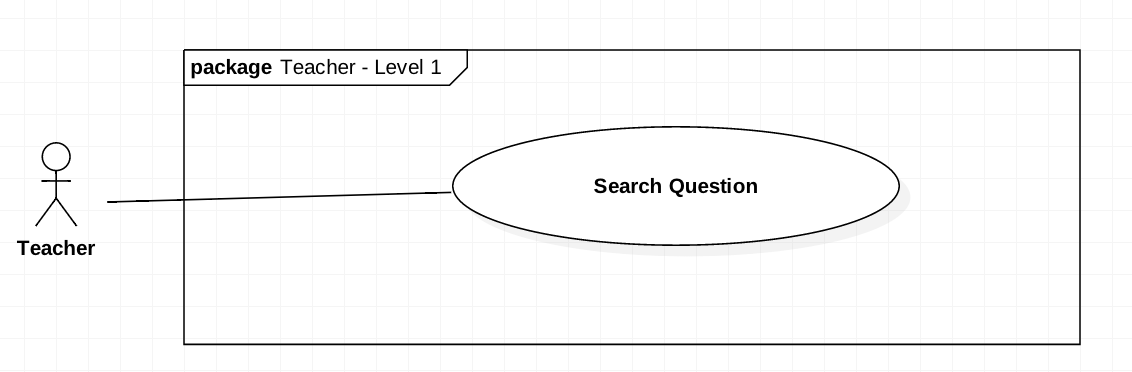


Figure 25 <Teacher> Search Question

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Teacher.19** | | | |
| **Use Case No.** | UC\_Teacher 19 | **Use Case Version** | 2.0 |
| **Use Case Name** | Search Question | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * *Teacher*   **Summary:**   * *This use case allows teacher search questions by question’s code and content* * *Support teacher can find question to edit question or remove question from question bank.*   **Goal:**   * *Teacher can finding the question with matching code and question content.*   **Triggers:**   * *Teacher send search command*   **Preconditions:**   * *Login the system with teacher role.* * *User input question information (question content, question code)*   **Post Conditions:**   * *Success: Questions will be found or not be found* * *Fail: Display error messages*  1. **Main Success Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher send command get list question* | *The system will display list question information at specify page:*   * + *Code*   + *Course*   + *Mark*   + *Level* | | *1* | *Teacher input name or code in the search field* |  | | *2* | *Teacher sends a command to get questions to follow name or code in the search field* | *The system run command and show questions have been searching on the question list:*   * + *Code*   + *Course*   + *Level*   + *Mark* |  1. **Alternative Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher sends a command to get questions to follow name or code in the search field* | *Show result not found on the screens* |   **Exceptions: N/A.**  **Relationships: N/A.**  **Business Rules:**   * *Get question list base on course code (A).* * *Filter all questions has content contain search value in A (B).* * *System send back B to client.* | | | |

Table 29 <Teacher> Search Question

#### <Teacher> Add Question by Form

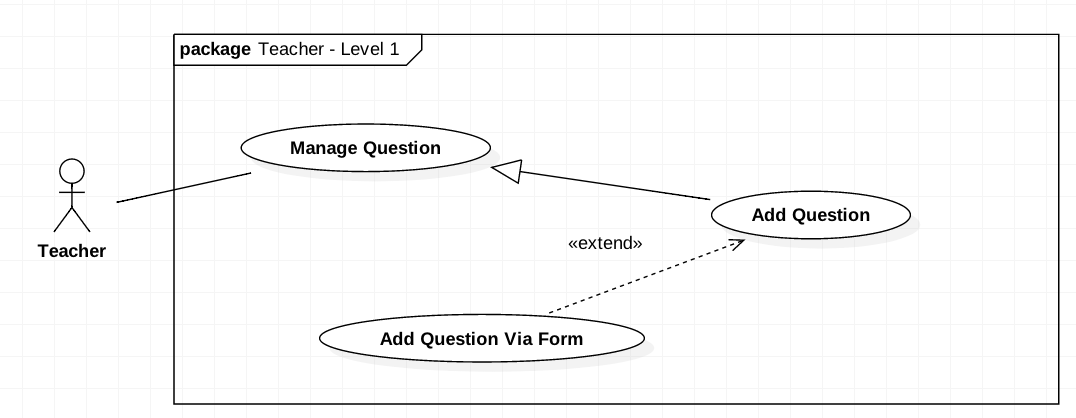


Figure 26 <Teacher> Add Question by Form

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Teacher.20** | | | |
| **Use Case No.** | UC\_Teacher 20 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Question by Form | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * 1. Teacher   **Summary:**   * 1. *This use case allows teacher to add questions by form.*   **Goal:**   * 1. *Teacher can add question by form.*   **Triggers:**   * 1. *Teacher send an create command.*   **Preconditions:**   * 1. *Login the system with teacher role.*   **Post Conditions:**   * 1. *Success: Questions will be added.*   2. *Fail: Questions will not be added.*  1. **Main Success Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher fill the form* | *The system requires teacher to input:*   * + *Chapter*   + *Level*   + *Mark*   + *Question Content*   + *Options* | | *2* | *Teacher send a save command to save the questions locally* |  | | *3* | *Teacher send a send command add question.* | *The system will run command and question and comeback to question list.*  *[Exception]* |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | *1* | *Teacher not fill full question info* | *Show message to fill full form.* | | *2* | *Teacher not choose write answer for question.* | *Show message require answer* |  1. **Alternative Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher add question with content matching >80% question already in database* | *Show warning may question already in database and confirm continuous to add or cancel action.* |   **Relationships: Add Question**  **Business Rules:**   * + - *Teacher fill question form (A).*     - *A has format:*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Content | Options | Mark | Level | Chapters | Learning outcomes |  * + - *Add A to question list.* | | | |

Table 30 <TEACHER> ADD QUESTION BY FORM

#### <Teacher> Add Question by Import File

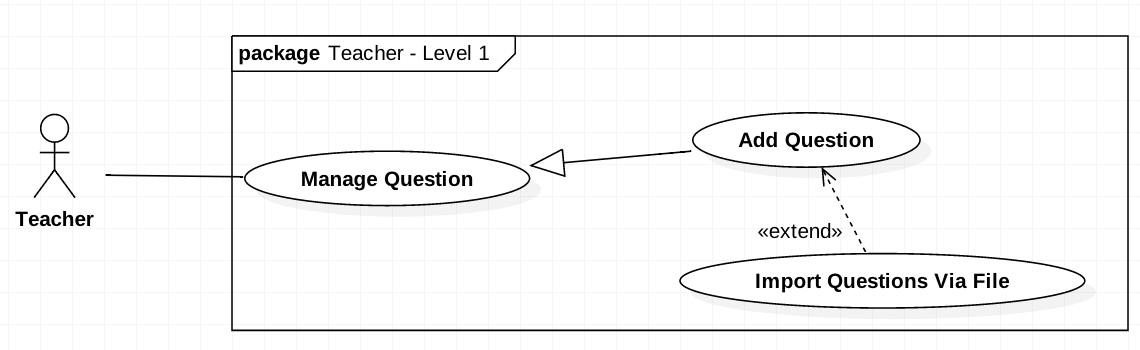


Figure 27 <Teacher> Add Question by Import File

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Teacher.21** | | | |
| **Use Case No.** | UC\_Teacher 21 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Question by Import File | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * *Teacher*   **Summary:**   * *This use case allows teacher to add questions by uploading file.*   **Goal:**   * *Upload multiple question list exported by Moodle (old system)*   **Triggers:**   * *Teacher send an create command.*   **Preconditions:**   * *Login the system with teacher role.*   **Post Conditions:**   * *Success: Questions will be added.* * *Fail: Questions will not be added.*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | |  |  |  | | *1* | *Teacher upload the file* | *The system requires teacher to upload a file with GIFT format or file extensions: .doc,.xml* | |  |  | *System run command to validate and check duplicate question.* | | *2* | *Teacher review the uploaded questions* | *Show question format status:*   * + *Success*   + *Warning*   + *Error* | | *3* | *Teacher send a submit command to save the questions to the database* | *System run add questions to add question list* |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships: Add Question**  **Business Rules:**   * + - *Uploaded a file (A).*     - *Validate A by checking file type (.xml, .docx, .txt) and file format (GIFT format)(B)*     - *If B valid, parse B to question list.*     - *Question information has format:*  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Content | Options | Mark | Level | Chapters | Learning outcomes | | | | |

Table 31 <Teacher> Add Question by Import File

#### <Teacher> Remove Question

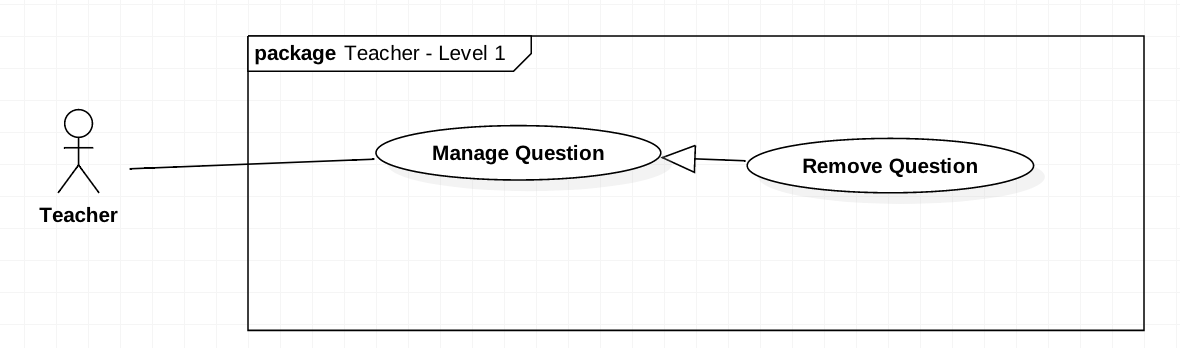


Figure 28 <Teacher> Remove Question

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Teacher.22** | | | |
| **Use Case No.** | UC\_Teacher 22 | **Use Case Version** | 2.0 |
| **Use Case Name** | Remove Question | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * + *Teacher*   **Summary:**   * + *Remove question from question bank.*   **Goal:**   * + *Remove question from question bank.*   **Triggers:**   * + *Teacher send delete command.*   **Preconditions:**   * + *Login the system with teacher role.*   **Post Conditions:**   * + *Success: Question will be removed.*   + *Fail: Notify error when application cannot connect to server.*  1. **Main Success Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher goes to list question list* | *The system will display list question information:*   * + *Course*   + *Mark*   + *Level* | | *2* | *Teacher choose delete question* | *The system will show a warning popup to ensure delete action again.* | | *3* | *Teacher confirm remove question in popup.* |  | | *4* | *Teacher send a delete question command.* | *The system will show a popup notify that question has been deleted successfully* |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships: N/A**  **Business Rules:**  *- Get all questions in system (A).*  *- Filter question in A base on question’s code (B).*  *- If B existed in A, remove B from system.* | | | |

Table 32 <Teacher> Remove Questions

#### <Teacher> Edit Questions

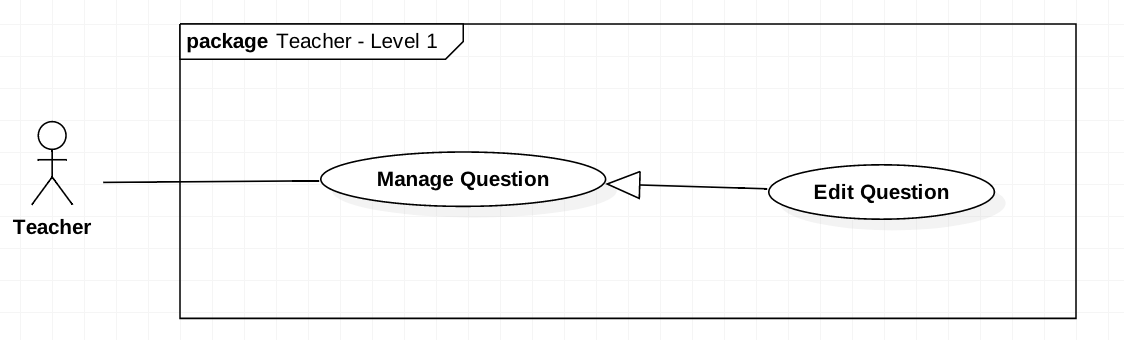


Figure 29 <Teacher> Edit Questions

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Teacher.23** | | | |
| **Use Case No.** | UC\_Teacher 23 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Questions | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * *Teacher*   **Summary:**   * *Update question information.*   **Goal:**   * *Update question information.*   **Triggers:**   * *Teacher send an update command.*   **Preconditions:**   * *Login the system with teacher role.*   **Post Conditions:**   * *Success: Questions will be edited.* * *Fail: Notify error when user does not full fill information or application cannot connect to server.*  1. **Main Success Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher goes to list question view* | *The system will display list question information:*   * + *Course*   + *Mark*   *Level* | | *2* | *Teacher choose to view detail question* | *The system will display detail information of this question:*   * + *Question code*   + *Question content*   + *Option*   + *Learning Outcome*   + *Chapter*   + *Mark*   + *Percent*   + *Level*   + *Right answer* | | *3* | *Teacher edit question information* | *The system require fill full info except learning outcome.* | | *4* | *Teacher send a submit command to save the questions.* | *The system run save question command to save and comeback to list question.* |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships: Manage Question**  **Business Rules:**  *- Get the question base on question’s code (A)*  *- Update A by user input as format :*   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Content | Options | Mark | Learning outcomes | Chapters | Level |   *- Save A to system.* | | | |

Table 33 <Teacher> Edit Questions

#### <Teacher> Approve Test Exam

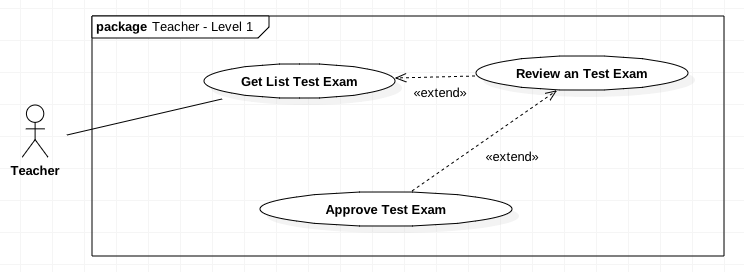


Figure 31 <Teacher> Approve Test Exam

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Teacher.25** | | | |
| **Use Case No.** | UC\_Teacher 25 | **Use Case Version** | 2.0 |
| **Use Case Name** | Approve Test Exam | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * *Teacher*   **Summary:**   * *User approve test exam.* * *Test exam which status is “Approve” can be release to create exam.*   **Goal:**   * *Approve test exam.*   **Triggers:**   * *Teacher send an update command.*   **Preconditions:**   * *Login the system with teacher role.*   **Post Conditions:**   * *Success: Status of test exam will be changed to “Approved”.* * *Fail: Notify error when application cannot connect to server.*  1. **Main Success Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher goes to Approve Test Exam View* | *The system will display list test exam information assigned to that teacher:*   * + *Course*   + *Number of questions*   + *Duration* | |  | *Teacher send command to view detail Test Exam* | *The system will display list question and require teacher take that exam.* | | *3* | *Teacher take an exam* |  | | *4* | *Teacher submit test exam* | *The system will calculate mark and change Test Exam status to taken and display Test Exam result to interface with information:*   * + *Mark*   + *Test Exam Result* | | *5* | *Teacher send command approve/ reject test exam.*  *[Alternative Scenario 1]* | *The system will change Test Exam to approved or rejected and come back display list exam assigned to approve.* |  1. **Alternative Scenario:**  |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Teacher send command edit questions* | *The system will require teacher to retake exam* |   **Exceptions: N/A**  **Relationships: Extend “Review an test exam”**  **Business Rules:**   * *Get test exam base on code (A).*   + *If A’s status is “Pending”, teacher take an exam. After A finished, A’s status change to “Taken”.*   + *If A’s status is “Taken:*     - *If teacher update A, A’s status change to “Edited”.*     - *If teacher approve. A’s status change to “Approved”.*     - *If teacher reject, A’s status change to “Rejected”.*   + *If A’s status is “Edited”, teacher available to retake and A’s status change to “Pending”.* | | | |

Table 34 <Teacher> Approve Test Exam

#### <Training Department Staff> Generate Test Exam

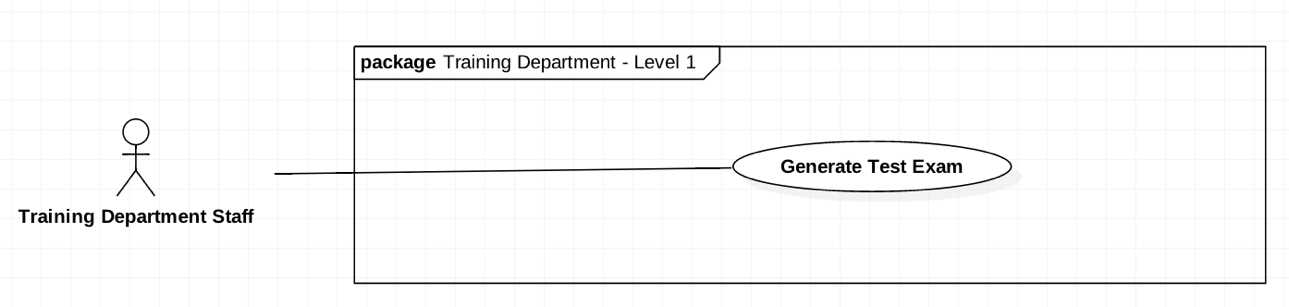


Figure 32 <Teacher> generate Test Exam

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Training Department Staff.25** | | | |
| **Use Case No.** | UC\_ Training Department Staff 25 | **Use Case Version** | 2.0 |
| **Use Case Name** | Generate Test Exam | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * + *Training department staff*   **Summary:**   * + *User can generate test exam by manufacturing or random.*   **Goal:**   * + *Generate new test exam to test exam bank.*   **Triggers:**   * + *Training Department Staff send create test exam command.*   **Preconditions:**   * + *Login the system with “Training department staff” role.*   **Post Conditions:**   * + *Success: Generate new test exam.*   + *Fail: Notify error when question bank is empty or application cannot connect to server.*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Manager click the Approve/Reject Exam button in the menu* |  | | *2* | *Manager click the Approve/Reject Exam tab* |  | | *3* | *Teacher select the semester* |  | | *4* | *Teacher send a command to get list of exam* | *Show list of exam with:*  *- Course*  *- Start Date*  *- End Date*  *- Duration*  *- Status*  *- Actions* | | *5* | *Teacher click “More Detail” button of an exam in the Actions column* |  | | *6* | *Teacher click the “Submit” button* |  | | *7* | *Teacher click the “Approve” button* |  | | *8* | *Teacher send an approve exam command* |  |   **Alternative Scenario: N/A**  **Exceptions: N/A.**  **Relationships: N/A.**  **Business Rules:**   * 1. *Get user’s input as format (A):* * *CourseCode* * *Chapters* * *Questions* * *LearningOutcomes* * *NumberOfTest* * *NumberOfQuestions* * *Duration* * *Type*   1. *If A’s type is “Manufacturing”, system create test exam base on A’s questions.*   2. *If A’s type is “LearningOutcomes”, system generate test exam base on A’s LearningOutcomes.*   3. *If A’s type is “Chapters”, system generate test exam base on A’s Chapters.* | | | |

Table 35 <Training department staff> Generate Test Exam

#### <Training Department Staff> Assign Approver

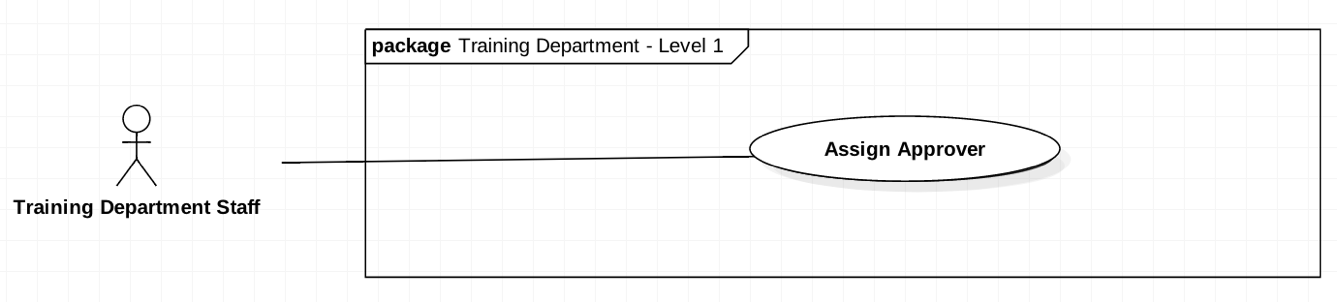


Figure 33 < Training Department Staff > Approve Test Exam

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE-UC\_Training Department Staff.25** | | | |
| **Use Case No.** | UC\_ Training Department Staff 25 | **Use Case Version** | 2.0 |
| **Use Case Name** | Assign Approver | | |
| **Author** | HieuCT | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:**   * + *Training department staff*   **Summary:**   * + *Assign test exam to a teacher.*   + *Support teacher approve test exam.*   **Goal:**   * + *Assign test exam to a teacher.*   **Triggers:**   * + *User need to select the test exam to assign.*   **Preconditions:**   * + *Login the system with “Training department staff” role.*   **Post Conditions:**   * + *Success: Assign test exam to a teacher.*   + *Fail: Notify error when user do not full fill information.*   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | *1* | *Training Department Staff goes to Assign Approver view* | *System require complete information:*   * + *Course*   + *Test Exam*   + *Start Date to approve*   + *End Date to take Exam*   + *Approver* | | *2* | *Training Department Staff full fill information* |  | | *3* | *Teacher run Assign Approver command* | *The system run Assign Approver command*  *[Exception]* |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | *1* | *Staff not full fill form* | *Show error message* |   **Alternative Scenario: N/A**  **Relationships: N/A.**  **Business Rules:**   * 1. *Get test exam base on code (A).*   2. *Update A by user’s input as format:*  |  |  |  | | --- | --- | --- | | Approver | StartDate | EndDate |  * 1. *System save A.* | | | |

Table 36 <Training department staff> Assign Approver