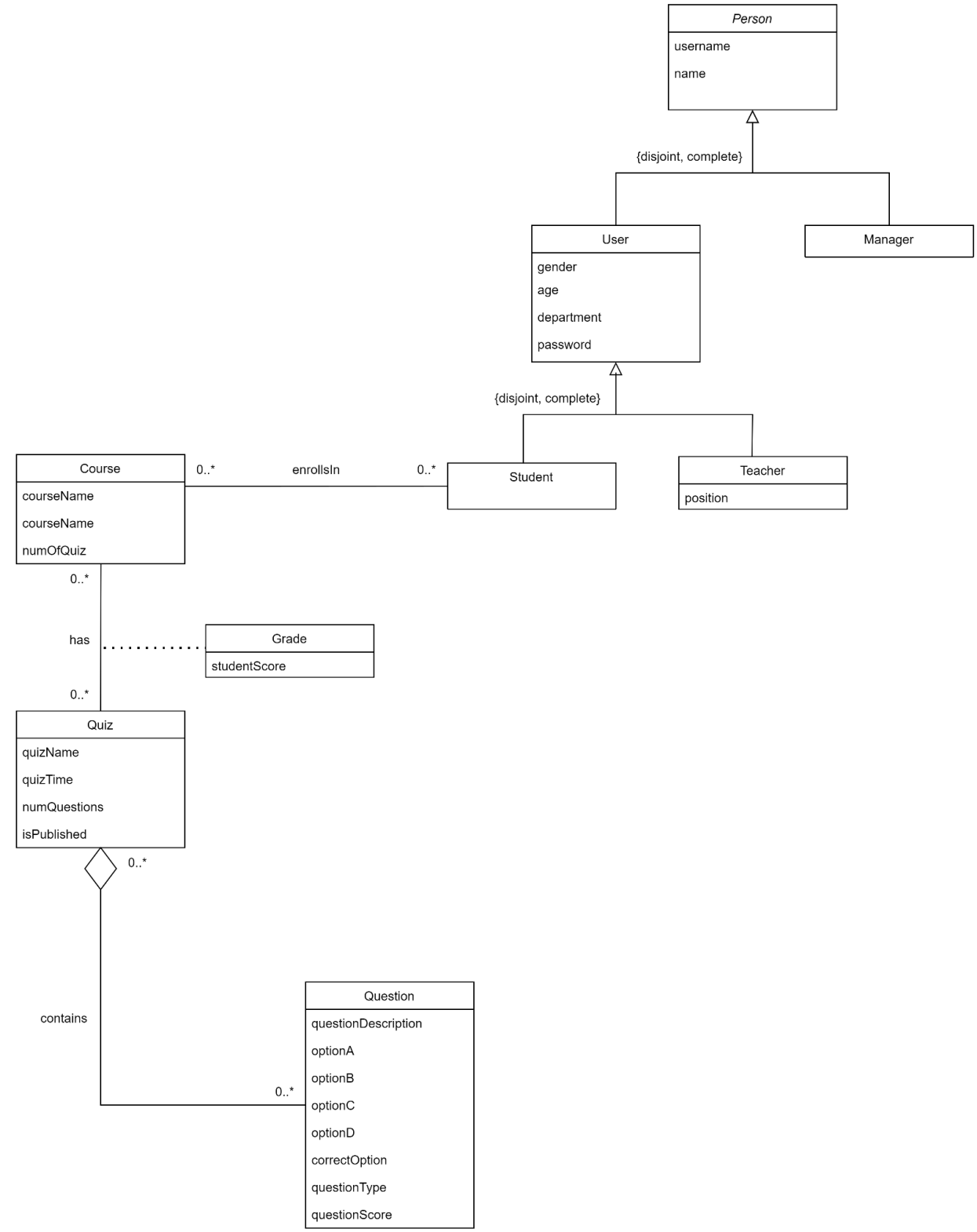
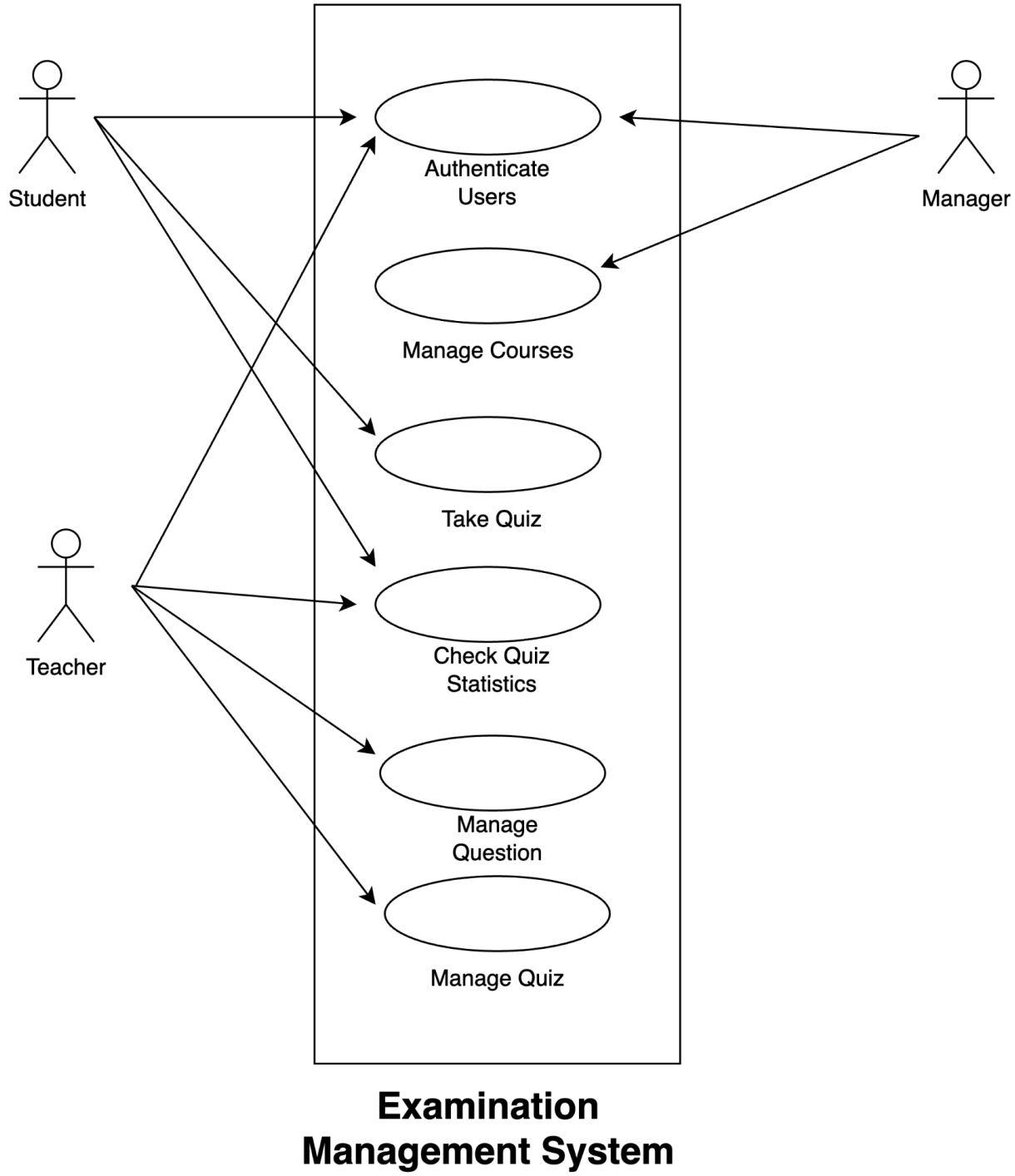
Class Diagram



Use Case Model

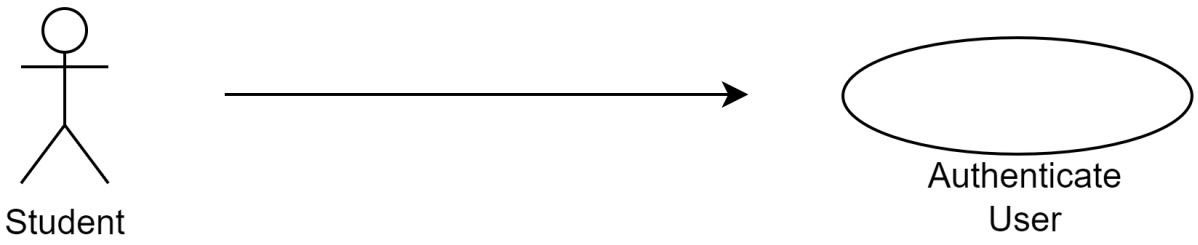


Task 1: Student Portal - YUEN, Bryan Qi Wen
Use case: Authenticate User

Brief description

- This use case describes how a user can log into or register for the examination management system

Use Case Diagram



Participating actors

- Student

Preconditions

- Student must be a member of the university.

Postconditions

- Registered Student is available in the database.

Basic Flow

1. The use case begins when the actor Student wants to log in or register into the system.
2. The system displays the interface for entering credentials.

{Choose Activity}

3.1 If the Student wants to LOG IN,

{Enter Credentials}

3.1.1 The Student clicks on the “login” button.

{Validate Credentials}

3.1.2. The system validates the credentials.

3.1.3. The system notifies the Student that the login is successful.

3.1.4. The Student clicks on the “ok” button.

3.2. If the Student wants to REGISTER,

3.2.1. The Student clicks on the “register” button.

{Begin Registration}

3.2.2. The system displays the register form.

3.2.3. The Student enters all credentials

3.2.4. The Student clicks on the “register” button.

{Registering Student}

3.2.5. The system creates a new account for the student.
{Complete Registration}
3.2.6. The Student is logged in and taken to the main Student Portal screen.
4. The use case ends.

Alternative Flows

A1: Invalid Credentials When Log In

At {Validate Credentials} if the credentials entered are incorrect,
1. The system notifies the Student that the credentials are incorrect.
2. The flow of events continues at {Enter Credentials}.

A2: Student Already Exists

At {Registering Student} if the Student already exists,
1. The system notifies the Student that the Student already exists.
2. The flow of events continues at {Begin Registration}.

A3: Password and Password Confirm Do Not Match

At {Registering Student} if the password and password confirm do not match,
1. The system notifies the Student that the password and password confirm do not match.
2. The flow of events continues at {Begin Registration}.

A4: Cancel Registration

At any point between {Begin Registration} and {Complete Registration},
1. The Student can click the “close” button to close the registration form.
2. The flow of events continues at {Choose Activity}.

A5: Required Fields Not Filled for Registering Student

At {Registering Student}, if the Student does not fill in all the required fields for registration,
1. The system displays a message indicating that all required fields must be filled in.
2. The flow of events continues at {Begin Registration}.

A6: Invalid Input Format for Registering Student

At {Registering Student}, if the Student enters invalid data (e.g., negative values for Age),
1. The system displays a message indicating that the input format is invalid.
2. The flow of events resumes at {Begin Registration}.

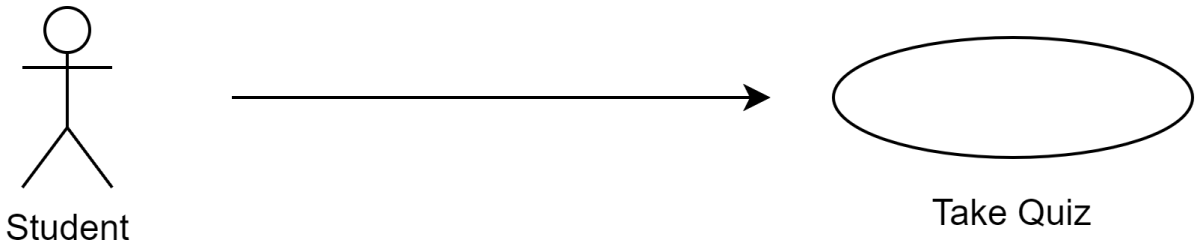
Use case detailed specification – Bryan

Use case: Take Quiz

Brief description

- This use case describes how a Student can take a quiz for a course that the Student is enrolled in

Use Case Diagram



Participating actor

- Student

Preconditions

- The Student must be enrolled in the course to which the quiz belongs.
- The Student must be logged into his account.
- The Quiz Questions will only be multiple-choice questions.

Postconditions

- Student can see the grade of the quiz taken.
- Grade of quiz taken inserted database.

Basic Flow

1. The use case begins when the actor Student wishes to take a quiz for a course the Student has enrolled in.
2. The system displays the Quiz Selection Screen.
3. The Student clicks on the “Start” button.
4. The system displays a separate window to show the question.

{Begin Taking Quiz}

5. While the student still has questions to answer
    - 5.1. Student chooses one of the options A, B, C, D
    - 5.2. If the Student wants to go to the NEXT question,
        - 5.2.1. The Student clicks on the “next” button
        - 5.2.2. Display the next question on the screen.
    - 5.3. If the Student wants to go to the PREVIOUS question,

5.3.1. The Student clicks on the “previous” button.
5.3.2. Display the previous question on the screen.

5.4. If the Student wants to SUBMIT his quiz,
5.4.1. The Student clicks on the “submit” button.
5.4.2. The system calculates the student's score.
5.4.3. The system records the student’s score in the database.
{Display Results}
5.4.4. The system displays the Student’s results for the quiz.
5.4.5. The Student clicks on the “OK” button.

5.5. If the Remain Time reaches 0,
5.5.1. The system calculates the student's score.
5.5.2. The system records the student’s score in the database.
{Display Results}
5.5.3. The system displays the Student’s results for the quiz.
5.5.4. The Student clicks on the “OK” button.

{Complete Taking Quiz}

6. The use case ends.

Alternative Flows

A1: Closing Quiz Window

At any point between {Begin Taking Quiz} and {Complete Taking Quiz},

1. The Student can close the window.
2. The flow of events is returned to {Display Results}.

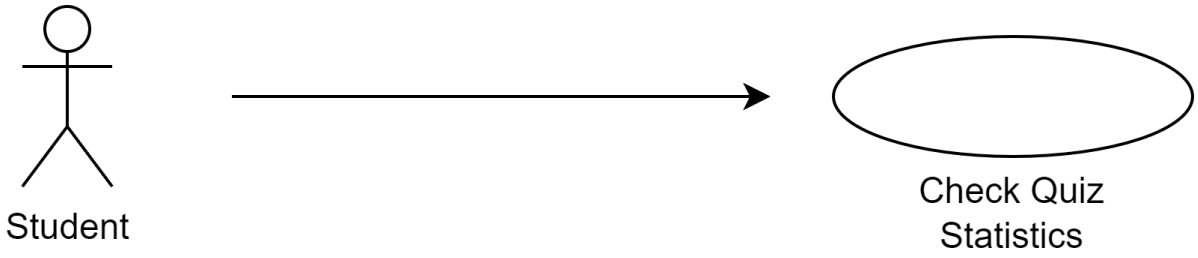
Use case detailed specification – Bryan

Use case: Check Quiz Statistics

Brief description

- This use case describes how a Student can check his grade statistics

Use Case Diagram



Participating actor

- Student

Basic Flow

1. The use case begins when the actor Student wishes to check his grade statistics for quizzes that the Student has taken.

{Display All Grade Statistics}

2. The system displays the grade statistics for all the quizzes the Student has taken.
3. While the Student is still checking the statistics
    - 3.1. If the Student wants to FILTER the course,
        - 3.1.1. Student can select courses from a drop-down list.
        - 3.1.2. Student clicks on “Filter” button.

        {Checking Filter}
        - 3.1.3. The system validates the filter request.
        - 3.1.4. The system displays the quiz statistics of the chosen course.
    - 3.2. If the Student wants to RESET the filters,
        - 3.2.1. The student clicks on the “Reset” button.
        - 3.2.2. The system resets all filters.
        - 3.2.3. The system displays the grade statistics for all the quizzes the Student has taken.
    - 3.4. If the Student wants to REFRESH the display,
        - 3.4.1. The system removes all previous filters.
        - 3.4.2. The system refreshes.
        - 3.4.2. The system displays new grade statistics if any.
4. The use case ends.

Alternative Flows

A1: No Course Selected When Filtering

At {Check Filter} if no course has been selected from the drop-down list,

1. The system notifies the student that no course has been selected.
2. The flow of events continues at {Display All Grade Statistics}.

Task 2: Teacher Portal - TAN, Juin
Use case name: Authenticate User

Brief description: This use case describes how a Teacher can log in or register into the system

Use Case Diagram

Participating actor
- Teacher

Preconditions
- Teacher must be a member of the university

Basic Flow
1. The use case begins when the actor Teacher wants to log in or register into the system.
2. The system displays the interface for entering credentials.
{Choose activity}
3.1 If the Teacher chooses to LOG IN,
    {Enter Credentials}
    3.1.1 The Teacher clicks on the “login” button.
    {Validate Credentials}
    3.1.2. The system validates the credentials.
    3.1.3. The system notifies the Teacher that the login is successful.
    3.1.4. The Teacher clicks on the “ok” button.
3.2. If the Teacher chooses to REGISTER,
    3.2.1. The Teacher clicks on the “register” button.
    {Begin Registration}
    3.2.2. The system displays the register form.
    3.2.3. The Teacher enters all credentials
    3.2.4. The Teacher clicks on the “register” button.
    {Registering Teacher}
    3.2.5. The system creates a new account for the teacher.
    {Complete Registration}
    3.2.6. The Teacher is logged in and taken to the main Teacher Portal screen.
4. The use case ends.

Alternative Flows

A1: Invalid Credentials When Log In

At {Validate Credentials}, if the credentials entered are incorrect,

1. The system notifies the Teacher that the credentials are incorrect.
2. The flow of events continues at {Enter Credentials}.

A2: Existing Username

At {Registering Teacher}, if the Teacher username already registered,

1. The system informs the Teacher that the Username is already registered.
2. The flow of events continues at {Begin Registration}.

A3: Password and Password Confirm Do Not Match

At {Registering Teacher}, if the password and password confirm do not match,

1. The system notifies the Teacher that the password and password confirm do not match.
2. The flow of events continues at {Begin Registration}.

A4: Cancel Registration

At any point between {Begin Registration} and {Complete Registration},

1. The Teacher can click the “close” button to close the registration form.
2. The flow of events continues at {Choose Activity}.

A5: Required Fields Not Filled for Registering Teacher

At {Registering Teacher}, if the Teacher does not fill in all the required fields for registration,

1. The system displays a message indicating that all required fields must be filled in.
2. The flow of events continues at {Begin Registration}.

A6: Invalid Input Format for Registering Teacher

At {Registering Teacher}, if the Teacher enters invalid data (e.g., negative values for Age),

1. The system displays a message indicating that the input format is invalid.
2. The flow of events resumes at {Begin Registration}.

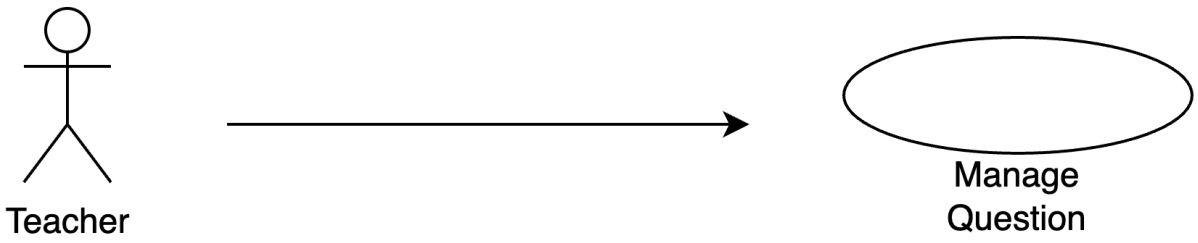
Use case detailed specification – Juin

Use Case: Manage Question

Brief description

- This use case describes how a teacher can manage the question bank by creating questions, updating questions, and deleting questions from the question bank.

Use-case Diagram



Participating actor

- Teacher

Preconditions

- Teacher must be a member of the university.

Postconditions

- Questions added must be multiple choice questions and must have four options.
- If there is any question added to the question bank, then the question should be shown in the question bank.
- If any question is deleted from the question bank, then the question should not be shown in the question bank.

Basic Flow

1. The use case begins when the Teacher chooses to access the Question Bank Management feature by clicking the “Question Bank Management” button.
2. The system displays the Question Bank Management screen.

{Select Activity}

3. While the Teacher has an activity to perform
    - 3.1. If the Teacher wants to RESET the filters
        - 3.1.1. The Teacher clicks on the “Reset” button.
        - 3.1.2. The system removes all the existing filters, if any.
        - 3.1.3. The system fetches all the questions from the database.
        - 3.1.4. The system displays all the questions in the questions bank.
    - 3.2. If the Teacher wants to FILTER the questions

3.2.1. The Teacher inputs filtering conditions, including Question, Type, or Score.
3.2.2. The Teacher clicks on the “Filter” button.
{Filter Questions}
3.2.3. The system fetches all the questions from the database, and filters them based on Question, Type, and Score.
3.2.4. The system displays all the filtered questions in the question list.

3.3. If the Teacher wants to DELETE the questions
3.3.1. The Teacher chooses the question that wants to be deleted.
3.3.2. The Teacher clicks on the “Delete” button.
{Delete Question}
3.3.3. The system deletes the selected question(s) from the database.
3.3.4. The system fetches all the questions from the database.
3.3.5. The system displays all the questions that fulfill the filtering conditions if there is any in the question list.

3.4. If the Teacher wants to REFRESH the questions
3.4.1. The Teacher clicks on the “Refresh” button.
3.4.2. The system removes all the existing filters, if any.
3.4.3. The system clears the data in the add/update question fields, if any.
3.4.4. The system fetches all the questions from the database.
3.4.5. The system displays them in the question list.

3.5. If the Teacher wants to ADD questions
{Begin Adding Question}
3.5.1. The Teacher fills in all the required fields.
3.5.2. The Teacher clicks on the "Add" button.
{Add Question}
3.5.3. The system adds the newly created question to the database.
{Complete Adding Question}
3.5.4. The system fetches all the questions from the database.
3.5.5. The system displays all the questions that fulfill the filtering conditions if there is any in the question list.
3.5.6. The system clears the data in the add/update question fields.

3.6. If the Teacher wants to UPDATE questions
3.6.1. The Teacher selects the question that wants to be updated.
{Begin Update Question}
3.6.2. The Teacher updates the question’s information.
3.6.3.The Teacher clicks on the "Update" button.
{Update Question}
3.6.4. The system updates the question in the database.
{Complete Update Question}
3.6.5. The system fetches all the questions from the database.
3.6.6. The system all the questions fulfill the filtering conditions if there is any in the question list.
3.6.7. The system also clears the data in the add/update question fields.

4. The use case ends.

Alternative Flows

A1: No Questions, Types, and Scores for Filtering

At {Filter Questions}, if there are no questions, types, and scores available for filtering,
1. The system displays a message indicating that there must be at least one condition for filtering the questions.
2. The flow of events resumes at step {Select Activity}.

A2: Required Fields Not Filled for Adding a Question

At {Add Question}, if the Teacher does not fill in all the required fields for adding a new question,
1. The system displays a message indicating that all required fields must be filled in.
2. The flow of events resumes at step {Begin Adding Question}.

A3: Required Fields Not Filled for Updating a Question

At {Update Question}, if the Teacher does not fill in all the required fields for updating a question,
1. The system displays a message indicating that all required fields must be filled in.
2. The flow of events resumes at step {Begin Update Question}.

A4: No Question Selected for Updating

At {Begin Update Question}, if the Teacher does not select a question to update,
1. The system displays a message indicating that a question must be selected to update.
2. The flow of events resumes at step {Select Activity}.

***A5: No Question Selected for* Deleting**

At {Delete Question}, if the Teacher does not select a question to be deleted,
1. The system displays a message indicating that a question must be selected to be deleted.
2. The flow of events resumes at step {Select Activity}.

***A6: Cancel Ac*tivity**

At any point between {Begin Adding Question} and {Complete Adding Question} or {Begin Update Question} and {Complete Update Question},
1. The Teacher can cancel the activity.
2. The flow of events resumes at step {Select Activity}.

A7:Unsaved Changes Alert for Adding a Question

At any point between {Begin Adding Question} and {Complete Adding Question}, if the Teacher attempts to navigate away without saving changes,
1. The system displays a message asking if they want to cancel the activity of adding a question and leave.
2. If the Teacher chooses yes, the flow of events resumes at step {Select Activity}. If the Teacher chooses no, the flow of events resumes at {Begin Adding Question}.

A8:Unsaved Changes Alert for Updating a Question
At any point between {Begin Update Question} and {Complete Update Question}, if the Teacher attempts to navigate away without saving changes,
1. The system displays a message asking if they want to cancel the activity of updating the question and leave.
2. If the Teacher chooses yes, the flow of events resumes at step {Select Activity}. If the Teacher chooses no, the flow of events resumes at {Begin Update Question}.

A9: Invalid Input Format for Adding a Question
At {Add Question}, if the Teacher enters invalid data (e.g., negative values for Score or Answer is neither of A, B, C, nor D),
1. The system displays a message indicating that the input format is invalid.
2. The flow of events resumes at {Begin Adding Question}.

A10: Invalid Input Format for Updating a Question
At {Update Question}, if the Teacher enters invalid data (e.g., negative values for Score or Answer is neither of A, B, C, nor D),
1. The system displays a message indicating that the input format is invalid.
2. The flow of events resumes at {Begin Update Question}.

A11: Selected Question for Update not found in database
At {Update Question}, if the selected question to be updated is not found in the database,
1. The system displays a message indicating that the selected question is not found in the database.
2. The flow of events resumes at {Select Activity}.

A12: Selected Questions for Deletion not found in database
At {Delete Question}, if the selected question(s) to be deleted is not found in the database,
1. The system displays a message indicating that the selected question(s) is not found in the database.
2. The flow of events resumes at {Select Activity}.

Use case detailed specification – Juin

Use Case: Manage Quiz

Brief description

- This use case describes how a teacher can create and manage exams using the questions in question bank.

Use-case Diagram

Participating actor

- Teacher

Preconditions

- Teacher must be a member of the university
- There is at least one course which is registered in the Examination Management System.

Postconditions

- Exams created should be under a course that is registered in the Examination Management System.
- If there is any newly created exam, then the exam should be shown in the exam list.
- If any exam is deleted from the exam list, then the exam should not be shown in the exam list.

Basic Flow

1. The use case begins when the Teacher chooses to access the Exam Management feature by clicking the “Exam Management” button.
2. The system displays the Exam Management screen.

{Select Activity}

3. While the Teacher has an activity to perform
    3.1. If the Teacher wants to RESET EXAM FILTERS
        3.1.1. The Teacher clicks on the “Reset” button on the left-hand side.
        3.1.2. The system removes all the existing exam filters, if any.

3.1.3. The system fetches all the exams and questions from the database.
3.1.4. The system displays all the exams in the exam list and the questions that fulfill the question filtering conditions if there is any in the question list.

3.2. If the Teacher wants to RESET QUESTION FILTERS
3.2.1. The Teacher clicks on the “Reset” button on the right-hand side.
3.2.2. The system removes all the existing question filters, if any.
3.2.3. The system fetches all the exams and questions from the database.
3.2.4. The system displays all the questions in the questions list and the exams that fulfilled the exam filtering conditions if there are any in the exam list.

3.3. If the Teacher wants to FILTER EXAM
3.3.1. The Teacher inputs filtering conditions for exams, including Exam Name, Course ID, or Publish Status.
3.3.2. The Teacher clicks on the “Filter” button on the left-hand side.
{Filter Exams}
3.3.3. The system fetches all the exams from the database, and filter them based on Exam Name, Course ID, and Publish Status.
3.3.4. If there is no input for any of the conditions, we just ignore that filtering condition.
3.3.5. The system displays all the filtered exams in the exam list.

3.4. If the Teacher wants to FILTER QUESTION
3.4.1. The Teacher inputs filtering conditions for questions, including Question, Type, or Score.
3.4.2. The Teacher clicks on the “Filter” button on the right-hand side.
{Filter Questions}
3.4.3. The system fetches all the questions from the database, and filters them based on Question, Type, or Score.
3.4.4. If there is no input for any of the conditions, we just ignore that filtering condition.
3.4.5. The system displays all the filtered questions in the question list.

3.5. If the Teacher wants to DELETE EXAM
3.5.1. The Teacher chooses the exam that wants to be deleted.
3.5.2. The Teacher clicks on the “Delete” button.
{Delete Exam}
3.5.3. The system deletes the selected exam(s) from the database.
3.5.4. The system fetches all the exams from the database.
3.5.5. The system displays all the exams fulfil the exam filtering conditions if there is any in the exam list.

3.6. If the Teacher wants to REFRESH the exams and questions
3.6.1. The Teacher clicks on the “Refresh” button.
3.6.2. The system removes all the existing exam filters and question filters, if any.
3.6.3. The system clears the data in the add/update exam fields, if any.
3.6.4. The system fetches all the exams and questions from the database.
3.6.5. The system displays them in the exam and question list.

3.7. If the Teacher wants to ADD exams

{Begin Adding Exam}

3.7.1. The Teacher fills out the required fields for the new exam, including Exam Name, Exam Time, Course ID, and Publish Status.

{Manage Questions Activity}

3.7.2. While the Teacher has activity to manage question

3.7.2.1. If DELETE FROM LEFT is selected

3.7.2.1.1. The Teacher selects the question in the exam to be removed from the exam.

3.7.2.1.2. The Teacher clicks the "Delete from left" button to remove the selected question(s) from the exam.

{Remove Questions}

3.7.2.1.3. The system removes the selected question(s) from the exam.

3.7.2.2. If ADD TO LEFT is selected

3.7.2.2.1. The Teacher selects the question in the question list to be added to the exam.

3.7.2.2.2. The Teacher clicks the "Add to left" button to add the selected question(s) to the exam.

{Add Questions}

3.7.2.2.3. The system adds the selected question(s) to the exam.

3.7.3. The Teacher clicks the "Add" button to create the new exam.

{Add Exam}

3.7.4. The system adds the newly created exam to the database.

{Complete Adding Exam}

3.7.5. The system fetches all the exams from the database.

3.7.6. The system displays all the exams that fulfill the exam filtering conditions if there is any in the exam list.

3.7.7. The system clears the data in the add/update exam fields and display of the questions in the exam.

3.8. If the Teacher wants to UPDATE exams

3.8.1. The Teacher selects the exam that wants to be updated from the exam list.

{Begin Update Exam}

3.8.2. The Teacher updates the exam's information.

{Manage Questions Activity}

3.8.3. While the Teacher has activity to manage question

3.8.3.1. If DELETE FROM LEFT is selected

3.8.3.1.1. The Teacher selects the question in the exam to be removed from the exam.

3.8.3.1.2. The Teacher clicks the "Delete from left" button to remove the selected question(s) from the exam.

{Remove Questions}

3.8.3.1.3. The system removes the selected question(s) from the exam.

3.8.3.2. If ADD TO LEFT is selected

3.8.3.2.1. The Teacher selects the question in the question list to be added to the exam.
3.8.3.2.2. The Teacher clicks the "Add to left" button to add the selected question(s) to the exam.
{Add Questions}
3.8.3.2.3. The system adds the selected question(s) to the exam.

3.8.4. The Teacher clicks on the "Update" button.
{Update Exam}
3.8.5. The system updates the exam in the database.
{Complete Update Exam}
3.8.6. The system fetches all the exams from the database.
3.8.7. The system displays all the exams that fulfill the exam filtering conditions if there is any in the exam list.
3.8.8. The system clears the data in the add/update exam fields and the questions in the exam.

4. The use case ends.

Alternative Flows

A1: No Question Selected for Creating an Exam

At {Add Exam}, if the Teacher does not select any questions for the new exam,

1. The system displays a message indicating that at least one question must be selected to create an exam.
2. The flow of events resumes at step {Manage Questions Activity}.

A2: No Questions in Exam When Updating an Exam

At {Update Exam}, if the exam has no questions in it,

1. The system displays a message indicating that at least one question must be added to the exam.
2. The flow of events resumes at step {Manage Questions Activity}.

A3: Required Fields Not Filled for Creating an Exam

At {Add Exam}, if the Teacher does not fill in all the required fields for creating a new exam,

1. The system displays a message indicating that all required fields must be filled in.
2. The flow of events resumes at step {Begin Adding Exam}.

A4: Required Fields Not Filled for Updating an Exam

At {Update Exam}, if the Teacher does not fill in all the required fields for updating an exam,

1. The system displays a message indicating that all required fields must be filled in.
2. The flow of events resumes at step {Begin Update Exam}.

A5: Exam Name with the same Course ID already existed

At {Add Exam}, if the Exam Name with the same Course ID already existed in the database,

1. The system displays a message indicating that an exam with the same Exam Name and Course ID already existed in the database.
2. The flow of events resumes at step {Begin Adding Exam}.

A6: No Filtering Conditions Provided for Exams

At {Filter Exams}, if the Teacher does not provide any filtering conditions for exams,

1. The system displays a message indicating that at least one filtering condition for exam must be provided.
2. The flow of events resumes at step {Select Activity}.

A7: No Filtering Conditions Provided for Questions

At {Filter Questions}, if the Teacher does not provide any filtering conditions for questions,

1. The system displays a message indicating that at least one filtering condition for question must be provided.
2. The flow of events resumes at step {Select Activity}.

A8: No Exam Selected for Deleting

At {Delete Exam}, if the Teacher does not select an exam to delete,

1. The system displays a message indicating that an exam must be selected for deletion.
2. The flow of events resumes at step {Select Activity}.

A9: Cancel Activity

At any point between {Begin Adding Exam} and {Complete Adding Exam} or {Begin Update Exam} and {Complete Update Exam},

1. The Teacher can cancel the activity.
2. The flow of events resumes at step {Select Activity}.

A10: Invalid Input Format for Creating an Exam

At {Add Exam}, if the Teacher enters invalid data (e.g., negative values for Exam Time),

1. The system displays a message indicating that the input format is invalid.
2. The flow of events resumes at {Begin Adding Exam}.

A11: Invalid Input Format for Updating an Exam

At {Update Exam}, if the Teacher enters invalid data (e.g., negative values for Exam Time),

1. The system displays a message indicating that the input format is invalid.
2. The flow of events resumes at {Begin Update Exam}.

A12: Selected Exam for Update not found in database

At {Update Exam}, if the selected exam to be updated is not found in the database,

1. The system displays a message indicating that the selected exam is not found in the database.
2. The flow of events resumes at {Select Activity}.

A13: Selected Exams for Deletion not found in database

At {Delete Exam}, if the selected exam(s) to be deleted is not found in the database,

1. The system displays a message indicating that the selected exam(s) is not found in the database.
2. The flow of events resumes at {Select Activity}.

A14:Unsaved Changes Alert for Creating an Exam

At any point between {Begin Adding Exam} and {Complete Adding Exam}, if the Teacher attempts to navigate away without saving changes,

1. The system displays a message asking if they want to cancel the activity of creating an exam and leave.
2. If the Teacher chooses yes, the flow of events resumes at step {Select Activity}. If the Teacher chooses no, the flow of events resumes at {Begin Adding Exam}.

A15:Unsaved Changes Alert for Updating an Exam

At any point between {Begin Update Exam} and {Complete Update Exam}, if the Teacher attempts to navigate away without saving changes,

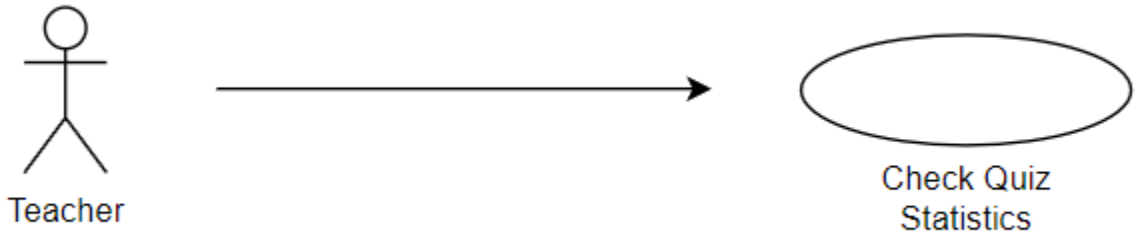
1. The system displays a message asking if they want to cancel the activity of updating the exam and leave.
2. If the Teacher chooses yes, the flow of events resumes at step {Select Activity}. If the Teacher chooses no, the flow of events resumes at {Begin Update Exam}.

Task 3: Teacher Grade Statistics System & Manager Portal - LOH, Angus Han Jern
Use Case: Check Grade Statistics

Brief Description

- This use case describes how a Teacher retrieves grade statistics from the system and filters the results accordingly

Use-case Diagram



Participating Actors

- Teacher

Precondition

- Teacher must be logged into the system

Basic Flow

1. The use case begins when the Teacher wants to check the grade statistics.
2. The system displays the interface for selecting between “Question Bank Management”, “Exam Management” and “Grade Statistics”.
3. The Teacher clicks “Grade Statistics”
4. While the Teacher is looking at the grade statistics
   {Display Course Statistics}
   4.1. The system displays the grade statistics for all courses, exams, and students.
   4.2. If the Teacher wants to filter the results
      4.2.1. The Teacher selects the Course, Exam or Student information
      4.2.2. The Teacher clicks “Filter”
      {Filter Course Check}
      4.2.3. The system validates the filter request
      4.2.4. The system applies the filter
   4.3. If the Teacher wants to reset the filter
      4.3.1. The Teacher clicks “Reset”
      4.3.2. The system resets the filters

Alternative Flow

A1: No Filter Selected

At {Filter Course Check} if no items are selected from the drop-down boxes,

1. The system informs the Manager to apply filter conditions.
2. The flow of events is resumed at {Display Course Statistics}.

Use case detailed specification – Angus

Use Case: Manage Users

Brief Description

- This use case describes how the Manager manages the information of Students and Teachers, which are categorized as Users.

Use-case Diagram

Participating Actors

- Manager

Precondition

- The Manager must be logged into the system.

Postcondition

- The information of Student or Teacher will be updated accordingly in the database.

Basic Flow

1. The use case begins when the Manager wants to manage the Users.
2. The system displays the interface for selecting between “Student Management”, “Teacher Management” and “Course Management”.
3. If the Manager wants to manage Students
   3.1. The Manager clicks “Student Management”
   3.2. While the Manager is managing students
   {Display Student Information}
   3.2.1. The system displays information of all existing Students on the screen.
   3.2.2. If the Manager wants to filter the student
   3.2.2.1. The Manager enters Student information (username, name, department).
   3.2.2.2. The Manager clicks “Filter”.
   {Filter Student Check}
   3.2.2.3. The system validates the Student information.
   3.2.2.4. The system retrieves and displays the information of specific Students.
   3.2.3. If the Manager wants to reset the filter
   3.2.3.1. The Manager clicks the “Reset” button.

3.2.3.2. The system clears the filters.

3.2.4. If the Manager wants to add Students into the system

3.2.4.1. The manager enters Student information (username, name, age, gender, department, password).

3.2.4.2. The Manager clicks “Add”.

{Add Student Check}

3.2.4.3. The system validates Student information.

3.2.4.4. The system adds Student information into the system.

3.2.5. If the Manager wants to update the information of a Student

3.2.5.1. The Manager enters Student information to be updated (username, name, age, gender, department, password).

3.2.5.2. The Manager clicks “Update”.

{Update Student Check}

3.2.5.3. The system validates Student information.

3.2.5.4. The system adds the Student.

3.2.6. If the Manager wants to delete a Student from the system

3.2.6.1. The Manager enters the “username” of the Student to be deleted.

3.2.6.2. The Manager clicks “Delete”.

{Delete Student Check}

3.2.6.3. The system validates Student information.

3.2.6.4. The system deletes the Student.

3.2.7. If the Manager wants to refresh the screen

3.2.7.1. The Manager clicks “Refresh”.

3.2.7.2. The system refreshes the screen.

4. If the Manager wants to manage Teachers

4.1. The Manage clicks “Teacher Management”

4.2. While the Manager is managing teachers

{Display Teacher Information}

4.2.1. The system displays the information of all existing Teachers.

4.2.2. If the Manager wants to filter the Teacher

4.2.2.1. The Manager enters the Teacher information to be filtered (username, name, department).

4.2.2.2. The Manager clicks “Filter”.

{Filter Teacher Check}

4.2.2.3. The system validates the filter request.

4.2.2.4. The system retrieves and displays the information of specific Teachers.

4.2.3. If the Manager wants to reset the filter

4.2.3.1. The Manager clicks “Reset”.

4.2.3.2. The system clears the filters.

4.2.4. If the Manager wants to add a Teacher into the system

4.2.4.1. The Manager enters Teacher information (username, name, age, gender, position, department, password).

4.2.4.2. The Manager clicks “Add”.

4.2.4.3. The system validates Teacher information.
{Add Teacher Check}
4.2.4.4. The system adds the Teacher.
4.2.5. If the Manager wants to update the information of a Teacher
4.2.5.1. The Manager enters Teacher information to be updated (username, name, age, gender, position, department, password).
4.2.5.2. The Manager clicks the “Update” button.
{Update Teacher Check}
4.2.5.3. The system validates Teacher information.
4.2.5.4. The system updates the Teacher information.
4.2.6. If the Manager wants to delete a Teacher from the system
4.2.6.1. The Manager enters the “username” of the Teacher to be deleted.
4.2.6.2. The Manager clicks “Delete”.
{Delete Teacher Check}
4.2.6.3. The system validates Teacher information.
4.2.6.4. The system deletes the Teacher.
4.2.7. If the Manager wants to refresh the screen
4.2.7.1. The Manager clicks “Refresh”.
4.2.7.2. The system refreshes the screen.

5. The use case ends.

Alternative Flows

A1: Student Filter Fail

At {Filter Student Check} if the username, name or department of the Student does not exist,

3. The system informs the Manager that the username/name/department does not exist.
4. The flow of events is resumed at {Display Student Information}.

A2: Student Username Already Exists in Student Add

At {Add Student Check} if the username already exists,

1. The system informs the Manager the username already exists.
2. The flow of events is resumed at {Display Student Information}.

A3: No Additional Information in Student Add

At {Add Student Check} if there are missing fields,

1. The system informs the Manager to input required fields.
2. The flow of events is resumed at {Display Student Information}.

A4: Student Does Not Exist in Update

At {Update Student Check} if the username of the Student cannot be found,

1. The system informs the Manager that the Student cannot be found.
2. The flow of events is resumed at {Display Student Information}.

A5: No Additional Information in Student Update

At {Update Student Check} if no additional fields are provided along with “username”,

1. The system informs the Manager to input fields.
2. The flow of events is resumed at {Display Student Information}.

A6: Student Does Not Exist in Delete

At {Delete Student Check} if the username of the Student cannot be found,

1. The system informs the Manager that the Student cannot be found.
2. The flow of events is resumed at {Display Student Information}.

A7: Teacher Filter Fail

At {Filter Teacher Check} if the username, name or department of the Teacher does not exist,

1. The system informs the Manager that the username/name/department does not exist.
2. The flow of events is resumed at {Display Teacher Information}.

A8: Teacher Username Already Exists in Teacher Add

At {Add Teacher Check} if the username already exists,

1. The system informs the Manager that the username already exists.
2. The flow of events is resumed at {Display Teacher Information}.

A9: No Additional Information in Teacher Add

At {Add Teacher Check} if there are missing fields,

1. The system informs the Manager to input required fields.
2. The flow of events is resumed at {Display Teacher Information}.

A10: Teacher Does Not Exist in Teacher Update

At {Update Teacher Check} if the username of the Teacher cannot be found,

1. The system informs the Manager that the Teacher cannot be found.
2. The flow of events is resumed at {Display Teacher Information}.

A11: No Additional Information in Teacher Update

At {Update Teacher Check} if no additional fields are provided along with “username”,

3. The system informs the Manager to input fields.
4. The flow of events is resumed at {Display Teacher Information}.

A12: Teacher Does Not Exist in Delete

At {Delete Teacher Check} if the username of the Teacher cannot be found,

1. The system informs the Manager that the Teacher cannot be found.
2. The flow of events is resumed at {Display Teacher Information}.

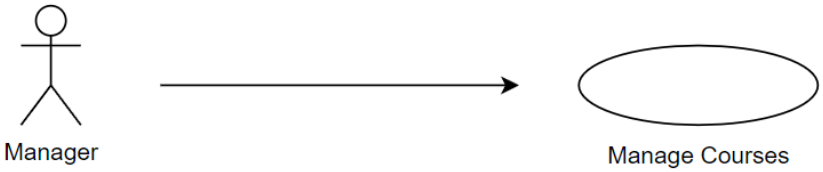
Use case detailed specification – Angus

Use Case: Manage Courses

Brief Description

This use case describes how the Manager manages course information.

Use-case Diagram



Participating Actors

- Manager

Preconditions

- The Manager must be logged into the system.

Postconditions

- The course information will be updated accordingly in the database.

Basic Flow

1. The use case begins when the Manager wants to manage courses.
2. The system displays the interface for selecting between “Student Management”, “Teacher Management” and “Course Management”.
3. If the Manager wants to manage courses
   3.1 The Manager clicks “Course Management”.
   3.2. While the Manager is managing courses
      {Display Course Information}
      3.2.1. The system displays the information of all existing courses.
      3.2.2. If the Manager wants to filter a course
         3.2.2.1. The Manager enters either information needed to filter (Course ID, Course Name or Department).
         3.2.2.2. The Manager clicks “Filter”.
         {Course Filter Check}
         3.2.2.3. The system validates the filter.
         3.2.2.4. The system applies the filter.
         3.2.2.5. The system retrieves the information according to the filter and displays the courses.
      3.2.3. If the Manager wants to reset the filter
         3.2.3.1 The Manager clicks “Reset”.
         3.2.3.2. The system removes all filters.
      3.2.4. If the Manager wants to add a new course

3.2.4.1. The Manager enters the Course information (Course ID, Course Name, Department).
3.2.4.2. The Manager clicks “Add”.
{Course Add Check}
3.2.4.3. The system validates the add request.
3.2.4.4. The system adds the new course.
3.2.5. If the Manager wants to delete a course
3.2.5.1. The Manager enters the Course ID of the course to be deleted.
{Course Delete Check}
3.2.5.2. The system validates the delete request.
3.2.5.3. The system deletes the course.
3.2.6. If the Manager wants to modify a course
3.2.6.1. The Manager enters the Course information (Course ID AND (Course Name OR Department)) of the course to be changed.
{Course Modify Check}
3.2.6.2. The system validates the course modification request.
3.2.6.3. The system modifies the existing course.
3.2.7. If the Manager wants to refresh the screen

4. The use case ends.

Alternative Flows

A1: Course Filter Check Fail

At {Course Filter Check} if the Course ID, Course Name or Department of the course does not exist,

1. The system informs the Manager that the Course ID/ Course Name/ Department does not exist.
2. The flow of events is resumed at {Display Course Information}.

A2: Course Already Exists in Add

At {Course Add Check} if the Course ID of the course already exists,

1. The system informs the Manager that the Course ID of the course already exists.
2. The flow of events is resumed at {Display Course Information}.

A3: No Additional Information in Course Add

At {Course Add Check} if the Course ID, Course Name or Department of the course are not filled,

1. The system informs the Manager that the Course ID/ Course Name/ Department are required fields.
2. The flow of events is resumed at {Display Course Information}.

A4: Course Does Not Exist in Delete Course

At {Course Delete Check} if the Course ID of the course does not exist,

1. The system informs the Manager that the Course ID of the course does not exist.
3. The flow of events is resumed at {Display Course Information}.

A5: Course Does Not Exists in Modify Course

At {Course Modify Check} if the Course ID of the course does not exist,

1. The system informs the Manager that the Course ID of the course does not exist.
2. The flow of events is resumed at {Display Course Information}.

A6: No Additional Information in Course Modify

At {Course Modify Check} if the Course ID AND either Course Name OR Department of the course are not filled,

1. The system informs the Manager that the Course ID is a required field, and at least one of Course Name OR Department are required.
2. The flow of events is resumed at {Display Course Information}.