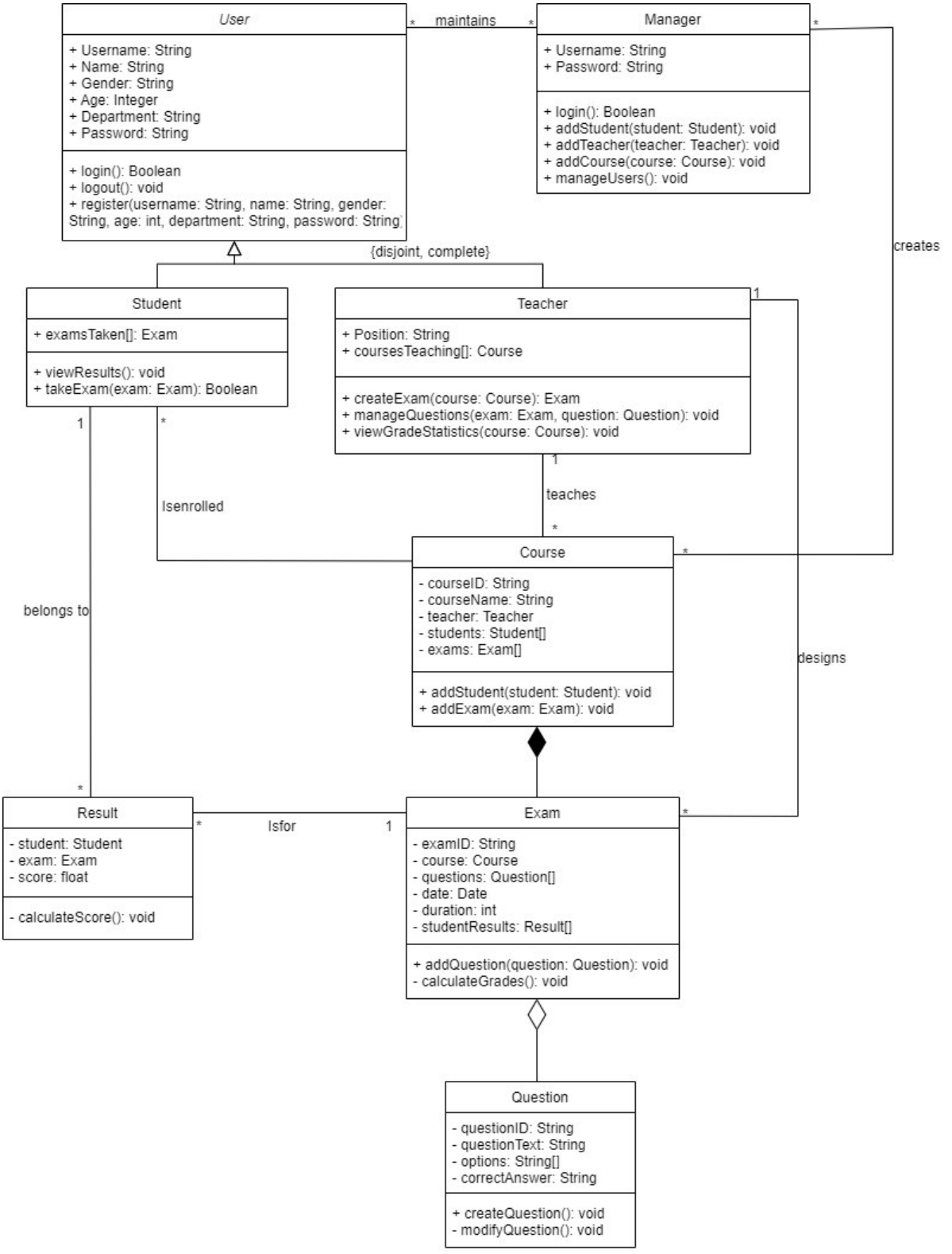
# Project Group Information

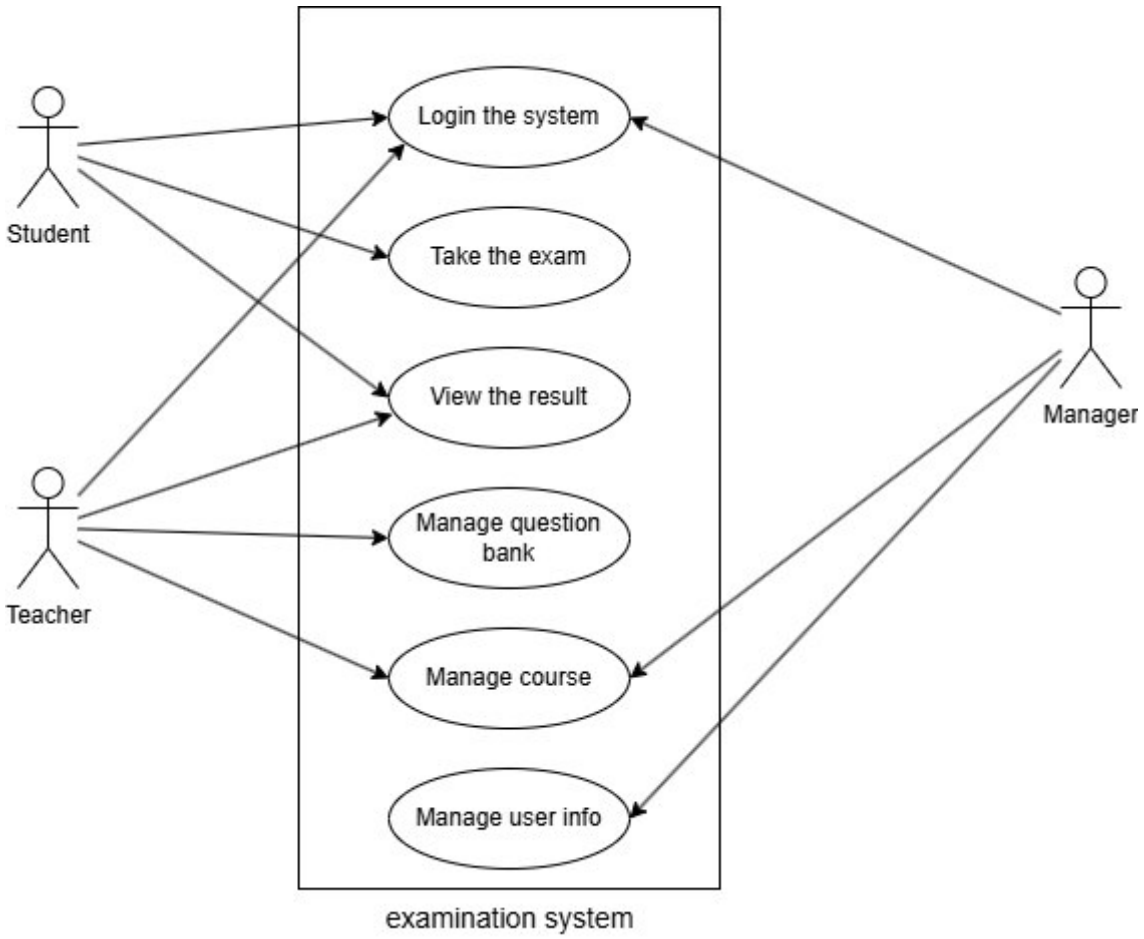
| Group No. | 13 |
|---|---|

| Name | GitHub User Id | UST Email | Is Team Leader? | Owner of Team Repo? | Roles/Task in Group Project | dev branch id |
|---|---|---|---|---|---|---|
| WANG Xinrui | wRtXmr | xwangjh@connect.ust.hk | No | No | Task 3 | Task 3 |
| SHI Juanquan | JessicaStOne | jshibh@connect.ust.hk | No | No | Task 2 | Task 2 |
| HUANG Ziyan | ZiyanHuang11 | zhuangfb@connect.ust.hk | Yes | Yes | Task 1 | Task 1 |

# Class Diagram



# User case Diagram

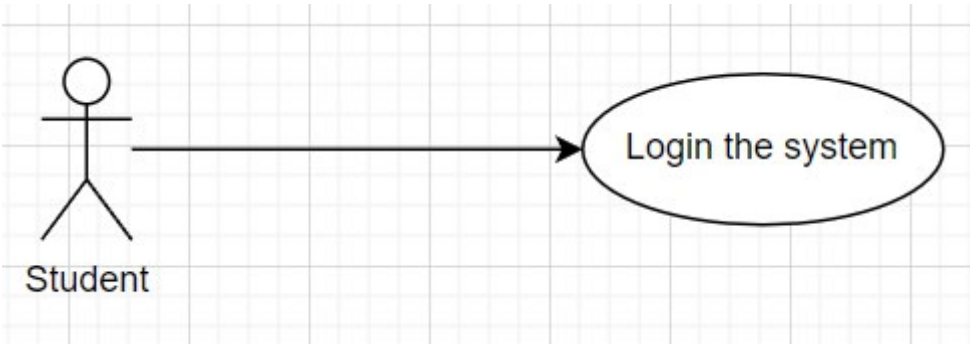


# User case specification

# Task # 1: Student Portal

HUANG Ziyan

## Use Case: Login the System




### Brief Description:

This use case describes how a student logs into the examination system, including both the login and registration processes, which provides access to functionalities such as taking exams and viewing results.

### Basic Flow:

1. The use case begins when the student actor selects "Student Login" from the welcome screen.
2. If the student wants to log in with an existing account:

   2.1 The system displays the student login interface, prompting the student to enter their username and password.

   {Enter credentials}

   2.2 The student inputs their credentials.

   {Begin verifying the credentials}

   - 2.2.1 If the credentials are valid, the system displays a "Login successful" message. After clicking OK, the system shows a welcome window with the title "Hi [username], Welcome to HKUST Examination System," with options to start an exam or view grade statistics.

3. If the student wants to register a new account:

   3.1 The system displays a registration form prompting the student to input details (Username, Name, Gender, Age, Department, Password, PasswordConfirm).

   {Enter information}

   3.2 If the student clicks Register:

   {Store entered information}

   - 3.2.1 The system stores the student's information at the backend and returns to the login screen.
   - 3.2.2 The student can then log in using the newly created credentials.

   3.3 If the student clicks Close:

   - 3.3.1 The system returns to the login screen.

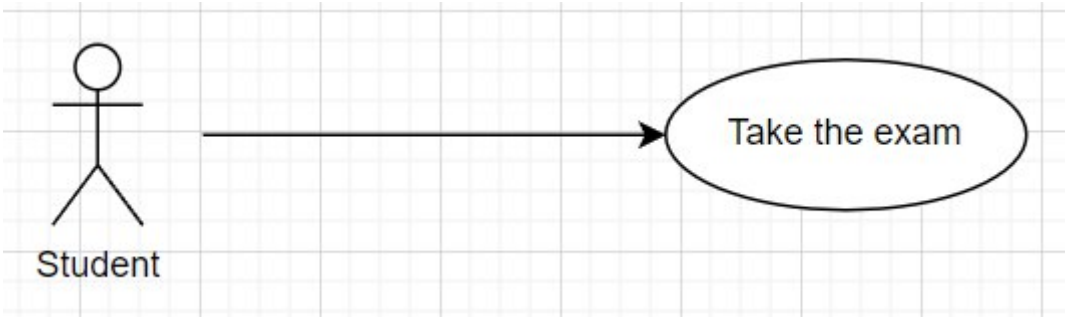
4. The use case ends when the student successfully logs into the system.

### Alternative Flows:

- **A1: Invalid Credentials**
  At {Begin verifying the credentials}, if the entered username or password is invalid:
  1. The system remains on the login interface.
  2. The flow of events is resumed at {Enter credentials}.
- **A2: Username Exists**
  At {Store entered information}, if the entered username already exists:
  1. The system informs the student that the account already exists.
  2. The flow of events is resumed at {Enter information}.
- **A3: Missing Required Fields**
  At {Enter credentials} or {Enter information}, if the student does not fill in required fields:
  1. The system prompts the student to fill in the missing fields.
  2. The flow of events is resumed at the relevant step.

---

## Use Case: Take the Exam




### Brief Description:

This use case describes how a student selects and takes an exam in the system after successfully logging in.

### Basic Flow:

1. The use case begins when the student logs into the system and chooses to take an exam.

   {Display available quiz}
2. The system displays the quiz selection options screen with available exams.
3. The student selects an exam and clicks Start.

   {Timer Begin Countdown}
4. The system displays the exam interface, including the quiz name, number of questions, a timer, a list of questions (left side), multiple-choice options, and navigation buttons (Next, Previous, Submit).

   {Student takes the quiz}
   - 4.1 The student clicks Next to move forward through the questions.
   - 4.2 The student clicks Previous to review or change previous answers.

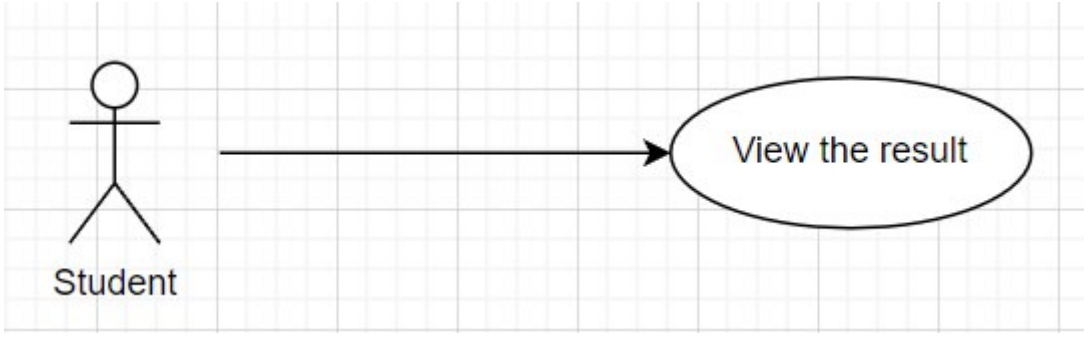
5. The student completes the quiz and clicks Submit.
6. The system stores the quiz results and displays the results for this quiz.
7. The use case ends.

### Alternative Flows:

- **A1: No Available Exams**
  At {Display available quiz}, if there are no available exams:
  1. The system displays a message indicating that no exams are available.
  2. The use case ends.
- **A2: Incomplete Exam**
  At {Student takes the quiz}, if the student does not complete the exam before the timer expires:
  1. The system automatically submits the exam.
  2. The use case ends when the exam is submitted.
- **A3: Quiz Interrupted**
  At {Timer Begin Countdown}, if the student's connection is lost or the system crashes during the exam:
  1. The system saves the student's progress, and the student is prompted to resume the exam after reconnecting.
  2. The flow of events resumes from {Student takes the quiz}.

---

## Use Case: View the Result




### Brief Description:

This use case describes how a student views the results of previously taken exams.

### Basic Flow:

1. The use case begins when the student logs into the system and chooses to view exam results by clicking Grade Statistics.

   {Display Course list options}
2. The system displays a list of courses with taken exams.
3. The student filters results by course using a Filter button.

   {Loading Course results}
4. The system retrieves and displays the results for the selected course, showing:

    - A table with columns Course, Exam, Score, Full Score, Time.
    - A dynamically generated bar chart representing the statistics of the selected exam.
5. If the student click Refresh button, the displayed results update.
6. If the student click Reset button， the results reset and student can select another course to display.
7. The use case ends when the student finishes viewing the results.

## Alternative Flows:

- **A1: Result Not Available**
  At {Loading Course results}, if the result for the selected course is not available:
    1. The system displays a message indicating the result is pending.
    2. The flow resumes at {Display Course list options}.
- **A2: Data Retrieval Error**
  At {Loading Course results}, if the system cannot retrieve the exam results due to a backend issue:
    1. The system displays an error message indicating that the results cannot be loaded at the moment.
    2. The flow resumes at {Display Course list options}.

# Task # 2: Teacher Portal

SHI Juanquan

## Use Case: Teacher Login/Registration

Brief Description:

This use case describes how a teacher logs into the system or registers a new account.

Basic Flow:

1. The use case begins when the teacher navigates to the login screen.

2. The system displays the login screen with two options:

Login: For teachers with existing accounts.

Register: For teachers who need to create a new account.

{Select Activity}

3. If the teacher selects the Login option:

The system prompts the teacher to enter their username and password.

The teacher inputs their credentials and clicks Login.

{Enter Login Credentials}

4. The system validates the credentials.

If valid, the system grants access to the teacher's portal.

If invalid, an error message is displayed, and the teacher is prompted to retry.

{Login Validated}

5. If the teacher selects the **Register** option:

- The system displays the registration form.

- The teacher enters the required information (full name, username, password, email, etc.) and clicks "Submit."

{Enter Registration Information}

6. The system validates the registration details.

If valid, the system stores the new teacher's information and confirms successful registration.

If invalid, the system prompts the teacher to correct the information.

{Registration Validated}

Alternative Flows:

A1: Invalid Login Credentials

At {Login Validated}, if the login credentials are incorrect:

1. The system informs the teacher that the credentials are invalid.
2. The flow of events resumes at {Enter Login Credentials}.

A2: Registration Error

At {Registration Validated}, if the registration data is invalid (e.g., missing fields or duplicate username):

1. The system informs the teacher about the error.
2. The flow of events resumes at {Enter Registration Information}.

Post-condition:

The teacher is either logged in successfully or registered in the system and can now log in.

Exception:

If the system encounters an error (e.g., network issue or server failure), the teacher is notified to retry later.

## Use Case: Manage Question Bank

Brief Description:

This use case describes how a teacher manages the question bank, including adding, editing, deleting, and filtering questions.

Basic Flow:

1. The use case begins when the teacher selects the **Question Bank Management** option from the main portal.
2. The system displays the question bank management interface, including fields for filtering, adding, editing, and deleting questions.

{Display Question Bank Interface}

3. The teacher can perform the following actions:

Add Question:

1. The teacher enters the new question details (e.g., question text, options, type, and score).
2. The teacher clicks "Add."
3. The system saves the new question in the question bank.

{Question Added}

Edit Question:

1. The teacher selects an existing question to edit.
2. The teacher modifies the question details and clicks "Update."
3. The system updates the question in the question bank.

{Question Updated}

Delete Question:

1. The teacher selects a question to delete.
2. The teacher clicks "Delete."
3. The system removes the question from the question bank.

{Question Deleted}

4. The system updates the question bank and refreshes the list of questions.

{Update Question Bank}

Alternative Flows:

A1: Invalid Question Dat

At {Question Added} or {Question Updated}, if the input data is invalid (e.g., missing fields or invalid format):

1. The system informs the teacher about the error.
2. The flow of events resumes at {Display Question Bank Interface}.

Post-condition:

The question bank is successfully updated with the added, edited, or deleted questions.

Exception:

If the system encounters an error while saving the changes (e.g., database connection issue), the teacher is notified, and the system prompts them to retry later.

## Use Case: Manage Course

Brief Description:

This use case describes how a teacher manages the courses they are assigned, including updating course details and reviewing student performance.

Basic Flow:

1. The use case begins when the teacher selects the **Course Management** option from the main portal.
2. The system displays a list of courses the teacher is responsible for.

{Display Course List}

3. The teacher selects a course to manage.

{Select Course}

4. The system displays the details for the selected course, including student performance data.

{Display Course Details}

5. The teacher can:

Update Course Details:

1. The teacher modifies course information (e.g., syllabus, schedule, or resources).
2. The teacher clicks "Save."
3. The system updates the course details in the database.

{Course Updated}

Review Student Performance:

1. The teacher reviews the student performance data displayed for the course.

{Performance Reviewed}

Alternative Flows:

A1: Invalid Course Selection

- At {Select Course}, if the teacher attempts to select a course they are not assigned to:

1. The system denies access and displays an error message.
2. The flow of events resumes at {Display Course List}.

Post-condition:

- The course details are successfully updated, or the teacher has reviewed the student performance data.

Exception:

- If the system encounters an issue saving the course details or retrieving the performance data, the teacher is notified, and the system prompts them to retry later.

# Task # 3: Teacher and Manager Portal

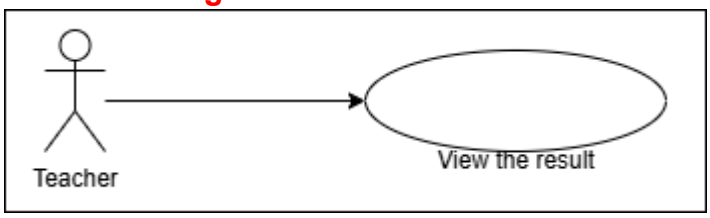
WANG Xinrui

## Use Case: View the Result

### Brief Description

This use case describes how a teacher views the grade statistics for students and courses/exams.

### Use-case Diagram




### Basic Flow

1. The Teacher clicks "Grade Statistics" button on the main dashboard.

**{Display Grade Statistics Screen}**

2. The system displays the Grade Statistics screen.
3. The Teacher can filter results based on:

- Course
- Exam
- Students

**{Select Filters}**

4. The Teacher selects the desired filters and clicks the "Filter" button.

**{Display Filtered Results}**

5. The system updates and displays the list of results for all students based on the selected filters.
6. The system provides graphical representations of results.
7. The Teacher can interact with the following buttons
   - 7.1. If the Teacher clicks the “Reset” button,
     **{Reset Filters}**
     - 7.1.1. The system clears all selected filters.
     - 7.1.2. The system displays the default result.
   - 7.2. If the Teacher clicks the “Refresh” button,
     **{Refresh Results}**
     - 7.2.1. The system re-fetches the current results based on the selected filters.
     - 7.2.2. The system updates and displays the latest results.
   - 7.3. If the Teacher clicks the “Filter” button,
     **{Apply Filters}**
     - 7.3.1. If the Teacher has selected valid filters, the system updates and displays the list of results based on the selected filters.
     - 7.3.2. If no filters are selected, the system informs the Teacher that at least one filter must be selected.
   - 7.4. If the Teacher changes any filter options,
     **{Update Filters}**
     - 7.4.1. The system allows the Teacher to reapply the filters.
     - 7.4.2. The flow of events continues at **{Press Filter Button}**.
8. The use case ends.

## Alternative Flows

### *A1: No Results Found*

At **{Display Filtered Results}** if no results match the selected filters,

1. The system informs the Teacher that no results were found.
2. The system prompts the Teacher to adjust the filters.
3. The flow of events is resumed at **{Select Filters}**.

### *A2: No Exam Selected*

At **{Press Filter Button}** if the Teacher has selected a course but has not selected an exam,

1. The system informs the Teacher that an exam must be selected for the chosen course.
2. The flow of events is resumed at **{Select Filters}**.

### *A3: No Students Selected*

At **{Press Filter Button}** if the Teacher has selected an exam but has not selected any students,

1. The system informs the Teacher that at least one student must be selected for the chosen exam.
2. The flow of events is resumed at **{Select Filters}**.

### *A4: Cancel Activity*

At any point before **{Press Filter Button}**,

1. The Teacher can cancel the activity.
2. The system asks for confirmation to cancel the activity.
   **{Confirm Cancellation}**
   2.1 If the Teacher confirms, the flow of events is resumed at the main dashboard.
   2.2 If the Teacher chooses not to cancel, the flow continues at the current screen.

### *A5: Course Without Exam*

At **{Display Filtered Results}** if the selected course does not have an associated exam,

1. The system informs the Teacher that the selected course has no exam available for results.
2. The system prompts the Teacher to select a different course or adjust filters.
3. The flow of events is resumed at **{Select Filters}**.

### *A6: Student Without Exam*

At **{Display Filtered Results}** if the selected student does not have any associated exam results,

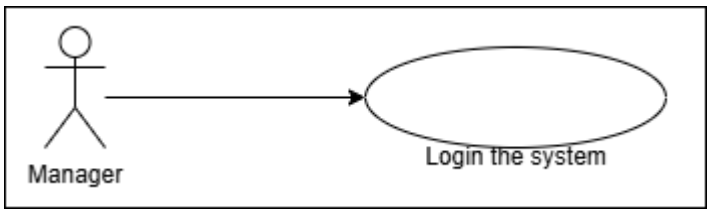
1. The system informs the Teacher that the selected student has no exam results available.
2. The system prompts the Teacher to select a different student or adjust filters.
3. The flow of events is resumed at **{Select Filters}**.

## Use case: Login the system

## Brief Description

This use case describes how a manager logs into the system.

## Use-case Diagram




## Basic Flow

1. The use case begins when the Manager actor clicks the "Manager Login" button on the main screen.
2. The system displays the login interface for the manager.

**{Enter Username and Password}**

3. The Manager enters their username and password.

**{Press Login Button}**

4. The Manager presses the "Login" button.

**{Display Login Message}**

5. The system displays a message indicating the login attempt status.
6. The Manager clicks "OK" on the message.

**{Display Main Dashboard}**

7. The system displays the main dashboard for the manager.
8. The use case ends.

## Alternative Flows

### *A1: Invalid Credentials*

At **{Press Login Button}** if the entered username or password is invalid,

1. The system informs the Manager that the credentials are invalid.
2. The flow of events is resumed at **{Enter Username and Password}**.

### *A2: Cancel Login*

At any point before **{Press Login Button}**,

1. The Manager can cancel the login attempt.
2. The flow of events is resumed at the main screen.

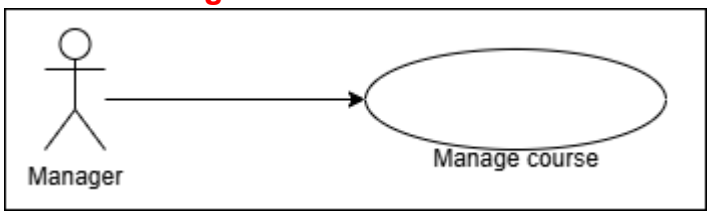
# Use case: Manage course

## Brief Description

This use case describes how a manager manages courses registered in the system.

Use-case Diagram

## Use-case Diagram




## Basic Flow

1. The Manager clicks the "Course Management" button on the main dashboard.

**{Display Course Management Screen}**

2. The system displays the Course Management screen.
3. The Manager can filter courses based on:

- Course ID
- Course Name
- Department

**{Select Filters}**

4. The Manager enters the desired filter criteria and clicks the "Filter" button.

**{Display Filtered Results}**

5. The system updates and displays the list of courses matching the filter criteria.
6. The Manager can interact with the following buttons:

6.1. If the Manager clicks the “Reset” button,

**{Reset Filters}**

6.1.1. The system clears all selected filters.

6.1.2. The system displays the default course list.

6.2. If the Manager clicks the “Refresh” button,

**{Refresh Results}**

6.2.1. The system re-fetches the current list of courses based on the selected filters.

6.2.2. The system updates and displays the latest course list.

6.3. If the Manager clicks the “Add” button,

**{Add New Course}**

6.3.1. The Manager fills out the form to add a new course, including:

- Course ID
- Course Name
- Department

6.3.2. The system validates the input and adds the new course to the database.

6.3.3. The system notifies the Manager that the course has been added successfully.

6.4. If the Manager clicks the “Modify/Update” button,

**{Update Course Information}**

6.4.1. The Manager selects a course from the list to update its information.

6.4.2. The system displays the selected course's information in the form.

6.4.3. The Manager makes the necessary changes and clicks the "Update" button.

6.4.4. The system validates the input and updates the course information in the database.

6.4.5. The system notifies the Manager that the course information has been updated successfully.

6.5. If the Manager clicks the “Delete” button,

**{Confirm Deletion}**

6.5.1. The Manager selects a course from the list to delete.

6.5.2. The system asks for confirmation to delete the selected course.

6.5.3. If the Manager confirms, the system removes the course from the database.

6.5.4. The system notifies the Manager that the course has been deleted successfully.

7. The use case ends Alternative Flows.

## Alternative Flows

### *A1: No Courses Found*

At **{Display Filtered Results}** if no courses match the selected filters,

1. The system informs the Manager that no courses were found.
2. The system prompts the Manager to adjust the filters.
3. The flow of events is resumed at **{Select Filters}**.

### *A2: Invalid Input When Adding a Course(include no input)*

At **{Press Add Button}** if the input data for adding a course is invalid,

1. The system informs the Manager which fields are invalid.
2. The flow of events is resumed at **{Add New Course Form}**.

### *A3: Cancel Activity*

At any point before **{Press Add Button}**,

1. The Manager can cancel the activity.
2. The system asks for confirmation to cancel the activity.

   **{Confirm Cancellation}**

   2.1. If the Manager confirms, the flow of events is resumed at the main dashboard.

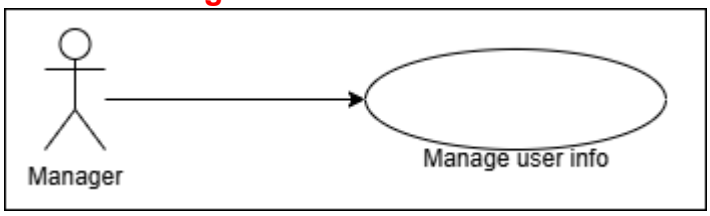
   2.2. If the Manager chooses not to cancel, the flow continues at the current screen.

## Use case: Manage user info

### Brief Description

This use case describes how a manager manages students and teachers' information in the system.

### Use-case Diagram




### Basic Flow

1. The Manager clicks either the "Student Management" or "Teacher Management" button on the main dashboard.

**{Display User Management Screen}**

2. The system displays the appropriate User Management screen (Student or Teacher).
3. The Manager can filter users based on:

- Username
- Name
- Department

**{Select Filters}**

4. The Manager enters the desired filter criteria and clicks the "Filter" button.

**{Display Filtered Results}**

5. The system updates and displays the list of users matching the filter criteria.
6. The Manager can interact with the following buttons:

   6.1. If the Manager clicks the "Reset" button,

   **{Reset Filters}**

   6.1.1. The system clears all selected filters.

   6.1.2. The system displays the default user list.

6.2. If the Manager clicks the “Refresh” button,

**{Refresh Results}**

6.2.1. The system re-fetches the current list of users based on the selected filters.

6.2.2. The system updates and displays the latest user list.

6.3. If the Manager clicks the “Add” button,

**{Add New User}**

6.3.1. The system validates the input and adds the new user to the database.

6.3.2. The system notifies the Manager that the user has been added successfully.

6.4. If the Manager clicks the “Update” button,

**{Update User Information}**

6.4.1. The system validates the input and updates the user's information in the database.

6.4.2. The system notifies the Manager that the user's information has been updated successfully.

6.5. If the Manager clicks the “Delete” button,

**{Confirm Deletion}**

6.5.1. The system asks for confirmation to delete the selected user.

6.5.2. If the Manager confirms, the system removes the user from the database.

6.5.3. The system notifies the Manager that the user has been deleted successfully.

7. The use case ends.

## Alternative Flows

### *A1: No Users Found*

At **{Display Filtered Results}** if no users match the selected filters,

1. The system informs the Manager that no users were found.
2. The system prompts the Manager to adjust the filters.
3. The flow of events is resumed at **{Select Filters}**.

### *A2: Invalid Input When Adding a User(include no input)*

At **{Press Add Button}** if the input data for adding a user is invalid,

1. The system informs the Manager which fields are invalid.
2. The flow of events is resumed at **{Add New User Form}**.

### *A3: Cancel Activity*

At any point before **{Press Add Button}**,

1. The Manager can cancel the activity.
2. The system asks for confirmation to cancel the activity.

**{Confirm Cancellation}**

2.1. If the Manager confirms, the flow of events is resumed at the main dashboard.

2.2. If the Manager chooses not to cancel, the flow continues at the current screen.