**Use Case Template: Comment on the Lesson**

**Use Case Name**: Comment on a Lesson   
**Description**: Allows a user to post a comment on a lesson uploaded by another user.

**Actor**: teacher, DB

**Preconditions**:

- The user must be logged in.

- The lesson on which to comment must exist in the database.

**Main Flow**:

1. The user navigates to a lesson they wish to comment on.

2. The system queries the database to retrieve and display the lesson along with existing comments.

3. The user enters their comment in the provided text area.

4. The user submits the comment.

5. The system validates the comment (e.g., checks for inappropriate content).

6. The system saves the comment to the database associated with the lesson.

7. The system updates the display to include the new comment under the lesson.  
8. The system triggers a notification to inform the owner of the lesson about the new comment.

**Alternative Flows**:

-**A5a. If the comment contains inappropriate content**: The system displays an error message, and the comment is not posted.

-**A3a. If the text area is left empty and submit is pressed**: The system prompts the user to enter text before submitting.

**Exception Flows**:

-**E6a. If the database update fails**: The system displays an error message indicating that the comment cannot be saved at this time.

**Postconditions**:

- The comment is saved in the database and displayed under the lesson.  
-The user(who uploaded) receives notifications about comments on their uploaded lessons.

**Use Case Template: Search Lessons**

**Use Case Name**: Search Lessons   
**Description**: Allows users to search for lessons using keywords.

**Actor**: teacher, DB

**Preconditions**:

- The user has access to the search functionality on the platform.

**Main Flow**:

1. The user enters one or more keywords in the search bar.

2. The user submits the search query.

3. The system processes the query and retrieves matching lessons from the database.

4. The system displays the results to the user.

**Alternative Flows**:

-**A3a. No results found**: The system displays a message indicating no lessons match the search criteria.

**Exception Flows**:

-**E3a. If the database search fails (e.g., server error)**: The system displays an error message and asks the user to try again later.

**Postconditions**:

- The system displays lessons matching the search criteria.

**Use Case Template: Filter Lessons**

**Use Case Name**: Filter Lessons   
**Description**: Allows users to filter lessons based on specific criteria such as category, difficulty level, or rating.

**Actor**: teacher, DB

**Preconditions**:

- Lessons are categorized with filterable attributes in the database.

**Main Flow**:

1. The user selects filter options (e.g., category, difficulty, rating).

2. The user applies the filter.

3. The system queries the database to retrieve and display lessons that match the filter criteria.

4. The system displays the filtered results.

**Alternative Flows**:

-**A3a. No lessons meet the filtering criteria**: The system displays a message indicating there are no lessons that match the filters.

**Exception Flows**:

-**E3a. If the database query fails (e.g., server error)**: The system displays an error message and asks the user to try again later.

**Postconditions**:

- The system displays only the lessons that meet the selected criteria.

**Use Case Template: Manage Users**

**Use Case Name**: Manage Users   
**Description**: Allows the administrator to manage user accounts, including creating new accounts, updating existing accounts, and deleting accounts.

**Actor**: Administrator, DB

**Preconditions**:

- The administrator must be logged in with administrative privileges.

**Main Flow**:

1. **Creating a New User Account**:

- The administrator selects the option to create a new user account.

- The system displays a form for entering new user details (e.g., username, password, email).

- The administrator enters the details and submits the form.

- The system validates the entered information.

- The system saves the new user account to the database.

- The system confirms account creation to the administrator.

2. **Updating an Existing User Account**:

- The administrator searches for a user account using search/filter criteria.

- The system displays matching user accounts.

- The administrator selects a user account to update.

- The system displays the current details of the user account.

- The administrator modifies the required information and submits the changes.

- The system validates the changes.

- The system updates the user information in the database.

- The system confirms the update to the administrator.

3. **Deleting a User Account**:

- The administrator searches for a user account to delete.

- The system displays matching user accounts.

- The administrator selects a user account to delete and confirms the decision.

- The system deletes the user account from the database.

- The system confirms the deletion to the administrator.

**Alternative Flows**:

-**A1a. If any field in the new user form is invalid**: The system displays an error message and requests the correct information.

-**A2a. If the administrator cancels the update**: No changes are saved, and the system returns to the previous screen.

-**A3a. If the administrator decides not to delete the user after selection**: The deletion process is canceled, and no action is taken.

**Exception Flows**:

-**E1a, E2a, E3a. If the database operation fails (create, update, delete)**: The system displays an error message stating that the operation could not be completed and suggests trying again later.

**Postconditions**:

- User accounts are created, updated, or deleted as per the administrator's actions.

**Use Case Template: Manage Lessons**

**Use Case Name**: Manage Lessons

**Actor**: Administrator, teacher, DB

**Description**: Allows the administrator to manage lessons, including creating, updating, and deleting lessons.

**Preconditions**:

- The administrator must be logged in with administrative privileges.

**Main Flow**:

1. **Creating a New Lesson**:

- The administrator selects the option to create a new lesson.

- The system displays a form for entering lesson details (e.g., title, description, category, video link, materials).

- The administrator enters the details and submits the form.

- The system validates the entered information.

- The system saves the new lesson to the database.

- The system confirms lesson creation to the administrator.

2. **Updating an Existing Lesson**:

- The administrator searches for a lesson using search/filter criteria.

- The system displays matching lessons.

- The administrator selects a lesson to update.

- The system displays the current details of the lesson.

- The administrator modifies the required information and submits the changes.

- The system validates the changes.

- The system updates the lesson information in the database.

- The system confirms the update to the administrator.

3. **Deleting a Lesson**:

- The administrator searches for a lesson to delete.

- The system displays matching lessons.

- The administrator selects a lesson to delete and confirms the decision.

- The system deletes the lesson from the database.

- The system confirms the deletion to the administrator.

**Alternative Flows**:

-**A1a. If any field in the new lesson form is invalid**: The system displays an error message and requests the correct information.

-**A2a. If the administrator cancels the update**: No changes are saved, and the system returns to the previous screen.

-**A3a. If the administrator decides not to delete the lesson after selection**: The deletion process is canceled, and no action is taken.

**Exception Flows**:

-**E1a, E2a, E3a. If the database operation fails (create, update, delete)**: The system displays an error message stating that the operation could not be completed and suggests trying again later.

**Postconditions**:

- Lessons are created, updated, or deleted based on the administrator's actions.

**Use Case Template: Sign-Up**

**Name:** Sign-Up

**Description:**

This use case describes the steps a student takes to create a new account on the website.

**Actors:** teacher, DB

**Preconditions:**

- The student has access to a device with internet connectivity.

- The website is accessible and functional.

**Basic Flow:**

1. The student navigates to the website's homepage.

2. The student clicks on the "Sign-Up" button.

3. The website presents a registration form to the student.

4. The student fills in the required fields in the registration form, including:

- Username

- Email address

- Password

- Confirm password

5. The student submits the registration form.

6. The website validates the information provided by the student:

- Checks if the username is unique.

- Validates the email address format.

- Ensures the password meets security requirements.

7. If the information is valid, the website creates a new user account for the student and logs them in automatically.

8. The website redirects the student to their dashboard or the homepage as a logged-in user.

9. The student can now access and contribute to lesson plans on the website.

**Alternative Flows:**

- If the username provided by the student is not unique, the website prompts the student to choose a different username.

- If the email address provided by the student is already associated with an existing account, the website prompts the student to log in instead or use a different email address.

- If any of the required fields are missing or invalid, the website displays error messages indicating the issues and prompts the student to correct them.

**Postconditions:**

- The student successfully registers as a user on the website and can access the features available to registered users.

**Use Case Template: Log-in**

**Name:** Log-In

**Description:**

This use case describes the steps a student takes to log in to their account on the website.

**Actors:** teacher, DB

**Preconditions:**

- The student is registered on the website.

- The student has access to a device with internet connectivity.

- The website is accessible and functional.

**Basic Flow:**

1. The student navigates to the website's homepage.

2. The student clicks on the "LogIn" button.

3. The website presents a login form to the student.

4. The student enters their username or email address and password into the respective fields.

5. The student submits the login form.

6. The website verifies the provided credentials against the stored user data.

7. If the credentials are valid, the website logs the student into their account and redirects them to their dashboard or the homepage as a logged-in user.

8. The student can now access the features available to registered users.

**Alternative Flows:**

- If the provided credentials are incorrect (either the username/email or password), the website displays a login error message.

- Extend: Display Log-In Error

1. The website displays an error message indicating that the provided credentials are incorrect.

2. The student is prompted to re-enter their username/email and password.

3. The flow returns to step 4 of the Basic Flow.

**Postconditions:**

- If the provided credentials are correct, the student successfully logs into their account on the website and can access the features available to registered users. If the provided credentials are incorrect, the student remains on the login page with an error message displayed.

**Use Case Template: Rate Lesson**

**Name:** Rate Lesson

**Description:**

This use case allows teachers to provide feedback on the quality and effectiveness of the lesson plans.

**Actors:** Teacher, Administrator.

**Preconditions:**

- Users must be registered and logged into the website.

- The lesson plan to be rated must be accessible and viewable by the user.

**Basic Flow:**

1. User navigates to the desired lesson plan they wish to rate.

2. User selects the "Rate Lesson" option.

3. User provides a rating on a scale (1 to 5 stars).

4.User submits the rating and comments.

5.The administrator records the rating and associated feedback.

**Alternative Flows:**

* If the teacher encounters technical difficulties while accessing the rating interface - he will contact the administrator for assistance.
* Rating Adjustment: If the user wishes to update their rating or comments, they can edit their submission before finalizing.

**Exception Flows**:

* Unauthorized Access: If a user attempts to rate a lesson plan without being logged in, they will be prompted to log in or register.
* Technical Issues: If there are technical issues preventing the submission of the rating, the system should display an error message and allow the user to retry.

**Postconditions:**

* The lesson plan's profile is updated to reflect the average rating and any comments provided by users.
* Users can continue to access and rate lesson plans, contributing to the collective feedback on the platform.

**Use Case Template: Download Lesson**

**Name:** Download Lesson

**Description:**

This use case enables registered users to download lesson plans from the website for offline use or reference.

**Actors:** Teacher, Administrator.

**Preconditions:**

- Users must be registered and logged into the website.

- The lesson plan must be available for download and accessible to the user.

**Basic Flow:**

1. User navigates to the desired lesson plan they wish to download.

2. User selects the "Download Lesson" option.

3. The system prompts the user to confirm the download action.

4. User confirms the download.

5. The administrator initiates the download process, providing the user with the lesson plan file.

6. Users save the downloaded lesson plan to their local device.

**Alternative Flows:**

* Preview before Download: Users may have the option to preview the lesson plan content before proceeding with the download.
* Downloading Multiple Lessons: Users may have the ability to download multiple lesson plans in a single action, such as selecting multiple lessons and choosing a "Download Selected" option.

**Exception Flows**:

* Unauthorized Access: If a user attempts to download a lesson plan without being logged in, they will be prompted to log in or register.
* Access Restrictions: If the lesson plan is restricted or requires special permissions to access, the system will notify the user accordingly.

**Postconditions:**

* The user successfully downloads the lesson plan file to their local device.
* The lessons are saved on the device from which the teacher connected.

**Use Case Template: Add to Favorites**

**Name:** Add to Favorites

**Description:**

This use case enables registered users to bookmark and save specific lesson plans for quick access and reference.

**Actors:** Teacher, Administrator.

**Preconditions:**

- Users must be registered and logged into the website.

- The lesson plan to be added to favorites must be accessible and viewable by the user.

**Basic Flow:**

1. User navigates to the desired lesson plan they wish to add to their favorites.

2. User selects the "Add to Favorites" option.

3. The system adds the selected lesson plan to the user's favorites list.

4. User confirms the addition to favorites.

5. The administrator updates the user's favorites list to include the newly added lesson plan.

**Alternative Flows:**

* Removing from Favorites: Users may have the option to remove lesson plans from their favorites list if they no longer wish to keep them bookmarked.
* Managing Favorites: Users may have the ability to organize their favorites into categories or folders for better organization and retrieval.

**Exception Flows**:

If the selected lesson is already in the teacher's favorites list - the system notifies the teacher and will not duplicate the lesson.

**Postconditions:**

* The selected lesson plan is successfully added to the user's favorites list.
* Users can easily access their favorite lesson plans from their account dashboard or a dedicated favorites section.