

Requirement & Design Specification

**Online Course Management System (OCMS)**

**Version: 1.0**

– Hanoi, September 2024 –

# Record of Changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| V1.0 | 18/9/2024 | A | manhpthe172481 | Database design |
| V1.0 | 19/9/2024 | A | manhpthe172481 | 2.1 Actor  2.2.1 UC for login |
| V1.0 | 21/9/2024 | A | huypqhe171509 | 2.3 Descriptions  1.5 Register |
| V1.0 | 22/9/2024 | A | manhpthe172481 | 1.2 User Login  1.3 Log out |
| V1.0 | 22/9/2024 | A | Tuanvmhe176448 | 1.4 Forgot password  1.8 Change password |
| V2.0 | 3/10/2024 | A | Manhpthe172481 |  |
| V2.0 |  |  |  |  |
| V2.0 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Contents

[Record of Changes 2](#_Toc179751901)

[I. Overview 5](#_Toc179751902)

[1. System Context 5](#_Toc179751903)

[2. User Requirements 5](#_Toc179751904)

[2.1 Actors 5](#_Toc179751905)

[2.2 Diagrams 6](#_Toc179751906)

[2.3 Descriptions 7](#_Toc179751907)

[II. Functional Requirements 7](#_Toc179751908)

[1. Authentication 7](#_Toc179751909)

[1.2 User Login 7](#_Toc179751910)

[1.3 Log out 8](#_Toc179751911)

[1.4 Forgot password 9](#_Toc179751912)

[1.5 Register 12](#_Toc179751913)

[1.6 View Profile 15](#_Toc179751914)

[1.7 Edit Profile 16](#_Toc179751915)

[1.8 Change password 18](#_Toc179751916)

[2. Instructor 20](#_Toc179751917)

[2.1 Course 20](#_Toc179751918)

[2.2 Lesson 25](#_Toc179751919)

[2.3 Assignment 31](#_Toc179751920)

[3. Student 36](#_Toc179751921)

[3.1 Course 36](#_Toc179751922)

[III. System Design 41](#_Toc179751923)

[1. Database Design 41](#_Toc179751924)

[1.1 Database Schema 42](#_Toc179751925)

[1.2 Table Descriptions 42](#_Toc179751926)

[2. Code Packages 43](#_Toc179751927)

[2.1 Package Diagram 43](#_Toc179751928)

[2.2 Package Descriptions 43](#_Toc179751929)

[IV. Details Design 44](#_Toc179751930)

[1. Add course 44](#_Toc179751931)

[2. Edit course 45](#_Toc179751932)

[3. View course details 46](#_Toc179751933)

[4. Create lesson 47](#_Toc179751934)

[5. View lesson details 48](#_Toc179751935)

[6. Update lesson 49](#_Toc179751936)

[7. Delete lesson 50](#_Toc179751937)

[8. Create assignment 51](#_Toc179751938)

[9. Edit assignment 52](#_Toc179751939)

[10. Delete assignment 53](#_Toc179751940)

[11. Enroll course 55](#_Toc179751941)

[12. Accessing course material 56](#_Toc179751942)

[13. Submit assignment 57](#_Toc179751943)

[13. Track progress 58](#_Toc179751944)

# I. Overview

## 1. System Context

The **Online Course Management System (OCMS)** is designed to help universities manage courses, student enrolment, and academic progress. It allows students to enroll in courses, access course materials, and track their progress. Instructors can manage course content, assignments, and assessments, while administrators oversee the overall course offerings and user management. The OCMS interacts with several external entities, including students, instructors, administrators, a payment system, and notification services like email or SMS. Students enroll in courses, submit assignments, and receive course materials and grades. Instructors create courses, provide learning materials, and grade students. Administrators manage user accounts, monitor enrolment, and generate system reports. The payment system handles transactions related to course enrollment, and notification services send important alerts to users regarding updates or deadlines. These interactions ensure a seamless flow of information between all parties involved, supporting the entire lifecycle of course management.

## 2. User Requirements

### 2.1 Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Student | The student actor represents all users enrolled in the system to access course materials, enroll in courses, submit assignments, and track academic progress. They provide input data such as enrolment requests, assignment submissions, and receive outputs like course materials, grades, and notifications. |
| 2 | Instructor | The instructor actor represents all users responsible for creating and managing course content, grading assignments, and interacting with students. Instructors input course materials and grades into the system, and the system returns student submissions and grades for review. |
| 3 | Administrator | The administrator actor oversees system management, including user account administration, course catalog management, and enrollment monitoring. Administrators provide input for system configurations and user management, receiving reports and system notifications in return. |

### 2.2 Diagrams

#### 2.2.1 UCs for Student

A diagram of a network

Description automatically generated

#### 2.2.2 UCs for Administrator

A diagram of a network

Description automatically generated with medium confidence

### 2.3 Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| 01 | Log in | Student, Admin | Actor provides valid credentials (username and password) to authenticate and gain access to the system. |
| 02 | Log out | Student, Admin | Actor terminates the session, effectively logging out of the system. |
| 03 | Forgot password | Student, Admin | Actor provides their email or username to request a password reset link, which will be sent to their registered email address. |
| 04 | Register | Student | Student provides required information (name, email, password, etc.) to create a new account in the system. |
| 05 | View Profile | Student, Admin | Actor views their personal profile information, including name, contact details, and any other relevant account data. |
| 06 | Edit Profile | Student, Admin | Actor updates their personal profile information, such as contact details or preferences. |
| 07 | Change password | Student, Admin | Actor updates their password by providing the current password and setting a new one. |
| 08 | View accounts | Admin | Admin views a list of all user accounts registered in the system, including students and other admins. |
| 09 | Deactivate account | Admin | Admin deactivates a user account, which prevents the user from logging into the system or accessing any services. |
| 10 | Edit account | Admin | Admin modifies account details for a specific user, such as updating roles, permissions, or personal information. |
| 11 | Create Lesson | Instructor | - Instructor choose a course - click on create new lesson - Upload file or video - click save |
| 12 | View Lesson | Student, Admin, Instructor | - Click on a course - click on a lesson link - View Lesson Content ( student need to enroll) |
| 13 | Update Lesson | Instructor | - Instructor choose course setting - Instructor choose a lesson - Instructor click Update button lesson content - Instructor click Save |
| 14 | Delete Lesson | Instructor | - Instructor choose course setting - Instructor choose a lesson - Instructor click Delete - Prompt to ensure instructor want to delete |
| 15 | Create Course | Instructor | - Instructor click create course button - Instructor enter course name and course title |
| 16 | View Course | Student, Admin, Instructor | - Click on course name to see lessons |
| 17 | Update Course | Instructor | - Instructor click the "Update Course" button - Instructors can modify course details,  such as the name, description, or content |
| 18 | Delete Course | Instructor | - Instructor choose the course want to delete - Instructor can click "Delete Course " button  - Prompt to ensure instructor want to delete |
| 19 | Create Assignment | Instructor | - Instructor choose a course  - Instructor add a Assignment in file  and due date |
| 20 | View Assigment | Student, Admin, Instructor | - Student( after enrollment) can see the Assignment in course dashboard |
| 21 | Update Assignment | Instructor | - Instructor click on assignment link in the course - Instructor click Edit button - Fill the new information like duedate, title, file material  - click Save button |
| 22 | Delete Assignment | Instructor | - Instructor choose the course - Click on assignment link - Instructor can click "Delete Assignment" button  - Prompt to ensure instructor want to delete |
| 23 | Student enroll course | Student | - Student choose a course on list courses - Student choose a course on list  - Student click Enroll button |
| 24 | Access Course Material | Student, Admin, Instructor | - Click on course link to view list of lessons available - click on an arbitrary lesson link to see detail |
| 25 | Submit Assignment | Student | - Student choose the course - Student click on Assignment link - Student click on Upload file and submit |
| 26 | Track Progress | Student, Admin, Instructor | - Student choose a course ( after enrolment) - Student can see the progress about completion percentage. |

# II. Functional Requirements

## 1. Authentication

### 1.2 User Login

This is the Login screen for the Online Course Management System (OCMS). The purpose of this screen is to authenticate users (Students, Admins, or Instructors) and allow them to access their respective dashboards. Users can either log in using their email and password or opt for a Google login for easier access

Screen Specifications

1. **Field Initializations**:
   * **Email**: Empty by default, expects the user to input a valid email address.
   * **Password**: Empty by default, expects the user to input their password.
   * **Remember Me**: Unchecked by default. If checked, the system will store the session and automatically log the user in the next time they visit.
2. **Button Behaviors**:
   * **Continue with Google**: Redirects to Google OAuth for authentication. If successful, the user is redirected back to their dashboard.
   * **Sign In**: Verifies the email and password entered. If valid, the user is directed to their respective dashboard. If invalid, an error message is displayed.
   * **Forgot Password**: Redirects the user to a password reset form.
   * **Sign Up**: Redirects the user to the account registration page.
3. **Business Rules**:
   * Users must provide valid credentials (correct email format, non-empty password) to log in.
   * The system should handle failed login attempts with appropriate error messages (e.g., "Invalid email or password").
   * If the user logs in successfully, the system should redirect to the appropriate dashboard based on their role (Student, Admin, or Instructor).
4. **Normal Flow**:
   * User enters valid **email** and **password**.
   * Clicks **Sign In**.
   * The system authenticates the user and redirects to their dashboard.
5. **Alternative Flow**:
   * **Forgot Password Flow**: If the user clicks "Forgot Password," they are taken to a screen where they can request a password reset link to be sent to their email.
   * **Google Login Flow**: If the user chooses **Google login**, they are redirected to Google OAuth, where they must approve access. Upon successful authentication, they are directed back to their dashboard.

A screenshot of a login form

Description automatically generated

### 1.3 Log out

This screen represents the header for the user after they have logged into the Online Course Management System (OCMS). Its primary function is to give the user access to navigation options and to allow them to log out of the system when desired. The user can click on the **Log out** button to safely end their session and return to the login page or homepage.

Screen Specifications

1. **Field Initializations**:
   * **Navigation links**: All links (Home, Courses, Dashboard) are visible and initialized based on the user's login status. After login, they are fully accessible.
   * **Favorites and Cart icons**: Both icons are initialized with a count of items (number of favorites and items in the cart).
   * **Log out button**: This button is visible only when the user is logged in. If clicked, it logs the user out and redirects them to the login or home page.
2. **Button Behaviors**:
   * **Log out button**: When clicked, it terminates the session, clears user session data, and redirects the user to the login screen. The button is only visible when the user is authenticated.
3. **Business Rules**:
   * Users must be logged in to access this screen (navigation options and log out button).
   * Clicking the **Log out** button will:
     + Invalidate the user’s session.
     + Clear any session-based data (e.g., authentication tokens, cached personal information).
     + Redirect the user to the homepage or login screen depending on the system configuration.
4. **Normal Flow**:
   * User clicks on the **Log out** button.
   * The system logs the user out, invalidates the session, and redirects the user to the login page.
5. **Alternative Flow**:
   * If the user navigates away from the screen without logging out, their session will remain active until they manually log out or their session times out.



### 1.4 Forgot password

The **Forgot Password** functionality is a multi-step process that allows users to reset their password in case they have forgotten it. The process is divided into three key screens:

1. **Forgot Password Screen**: This screen prompts the user to input their email address to receive a password reset link.
2. **Enter OTP Code Screen**: After receiving the OTP via email, the user must input the correct OTP code to proceed.
3. **Reset Password Screen**: This screen allows the user to enter a new password and confirm it to reset their account.

Screen Specifications

1. **Forgot Password Screen**
   * **Field Initializations**:
     + **Email Address Field**: Empty by default. The user must input a valid registered email address.
     + **Reset Password Button**: Enabled when a valid email is entered.
   * **Business Rules**:
     + The email field must be validated (check for valid format, and that the email is registered in the system).
     + If the email is valid, the system sends a password reset link or OTP code to the provided email address.
     + If the email is not found, an error message appears prompting the user to enter a valid email.
   * **Normal Flow**:
     + User enters their email and clicks **Reset Password**.
     + The system sends a password reset email with an OTP or link to the user’s email.
   * **Alternative Flow**:
     + If the email address is invalid or not found in the system, an error message will display ("Invalid email address").
2. **Enter OTP Code Screen**
   * **Field Initializations**:
     + **OTP Code Field**: Empty by default, expects the user to enter the OTP sent to their email.
     + **Verify OTP Button**: Enabled once an OTP is entered.
   * **Business Rules**:
     + The OTP code is validated against the system's records. If the code matches, the user proceeds to reset their password.
     + If the OTP is incorrect, the user can try again or request a new OTP code via the **Resend OTP** link.
   * **Normal Flow**:
     + User enters the OTP and clicks **Verify OTP**.
     + The system verifies the OTP, and if valid, it redirects the user to the password reset screen.
   * **Alternative Flow**:
     + If the OTP is incorrect, the user is prompted with an error message ("Invalid OTP").
     + The user can request a new OTP using the **Resend OTP** link.
3. **Reset Password Screen**
   * **Field Initializations**:
     + **New Password Field**: Empty by default, expects the user to enter a new password.
     + **Confirm New Password Field**: Empty by default, requires confirmation of the password entered in the first field.
     + **Reset Password Button**: Enabled when both fields are filled, and the passwords match.
   * **Business Rules**:
     + Passwords must meet the system’s security requirements (minimum length, inclusion of special characters, etc.).
     + The two password fields must match to proceed.
   * **Normal Flow**:
     + The user enters a new password and confirms it by re-entering it in the second field.
     + If the passwords match and meet the system requirements, the user clicks **Reset Password**.
     + The system updates the user’s password and redirects them to the login page.
   * **Alternative Flow**:
     + If the passwords do not match, an error message appears, prompting the user to correct the input ("Passwords do not match").
     + If the password does not meet security criteria, an error message suggests a stronger password.

A screenshot of a login page

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a login screen

Description automatically generated

### 1.5 Register

The **Register screen** in the Online Course Management System (OCMS) is designed for new users to create an account. This screen allows users to either register by providing an email and password or use their **Google** account for quicker registration. The registration process is secured by email verification, using an **OTP** (One-Time Password) sent to the user’s email. The process is divided into two key screens:

1. **Create Account**: Users enter their email, password, and confirm the password.
2. **Enter OTP Code**: Users input the OTP sent to their email to verify the account.

Screen Specifications

1. **Create Your Account Screen**
   * **Field Initializations**:
     + **Email Field**: Empty by default. Expects a valid email format.
     + **Password Field**: Empty by default. Requires the user to input a password that adheres to security standards.
     + **Confirm Password Field**: Empty by default. The user must re-enter the same password for confirmation.
   * **Business Rules**:
     + **Google Sign-In Option**: Allows users to register using their Google account. The system retrieves necessary information like email and bypasses manual input.
     + **Email Validation**: The email field must contain a valid email format (e.g., user@example.com). If invalid, the system displays an error.
     + **Password Validation**: The password must meet the system's security requirements, such as minimum length and inclusion of alphanumeric or special characters.
     + **Password Matching**: The password and confirm password fields must match. If they do not, an error message is shown ("Passwords do not match").
     + After a successful sign-up attempt, the system sends an OTP code to the user’s email to verify the account.
   * **Normal Flow**:
     + User fills in the email, password, and confirms the password.
     + Clicks **Sign Up**.
     + The system validates the fields, registers the account, and sends an OTP code to the user's email.
   * **Alternative Flow**:
     + If any field is invalid (e.g., incorrect email format, weak password, or mismatching passwords), the user is prompted with error messages and needs to correct the input.
2. **Enter OTP Code Screen**
   * **Field Initializations**:
     + **OTP Code Field**: Empty by default. Expects the user to input the OTP received via email.
     + **Verify OTP Button**: Enabled once an OTP is entered.
   * **Business Rules**:
     + The OTP must match the code sent to the user’s email. If the user enters the wrong code, an error message is displayed ("Invalid OTP").
     + If the OTP is not received, the user can request another OTP via the **Resend OTP** link.
     + After verifying the OTP, the account is successfully created, and the user is redirected to the login or dashboard screen.
   * **Normal Flow**:
     + User enters the OTP code and clicks **Verify OTP**.
     + The system checks if the OTP matches the one sent via email. If valid, the account is verified.
   * **Alternative Flow**:
     + If the OTP is invalid or expired, the user is prompted to either try again or request a new code via **Resend OTP**.

A screenshot of a login form

Description automatically generated

A screenshot of a computer

Description automatically generated

### 1.6 View Profile

The **View Profile screen** in the Online Course Management System (OCMS) allows logged-in users to view and manage their personal information, including email, role, gender, and last login details. This screen is accessible after the user logs into the system and clicks on the **Dashboard** button in the header. The screen gives users an overview of their profile details and provides an option to update their information.

**Screen Specifications**

1. **Field Initializations**:
   * **Email Field**: Pre-filled with the email address of the user based on the data stored in the system.
   * **Role Field**: Displays the user’s role (e.g., Student, Admin, Instructor). This field is usually non-editable as it’s defined by system roles.
   * **Gender Field**: Pre-selected based on the user’s previous selection or profile settings.
   * **Last Login Field**: Displays the timestamp of the last login for the user.
2. **Update Profile Button**:
   * The **Update Profile** button allows the user to modify their profile information, such as gender or other details (if applicable), and save the changes. Once clicked, the system validates the changes and updates the user profile accordingly.
3. **Business Rules**:
   * **Login Requirement**: Users must be logged in to access the **Dashboard** and **Profile** page. If a user tries to access this page without being logged in, they will be redirected to the login page.
   * **Data Validation**: When updating profile details, the system checks for valid input (e.g., valid email format). If validation fails, an error message will prompt the user to correct their input.
   * **Update Profile**: Only certain fields like gender and contact information may be editable, while fields like email and role may be locked based on system rules.
4. **Normal Flow**:
   * The user logs into the system using their credentials.
   * After logging in, the user clicks on the **Dashboard** link in the header.
   * The user is directed to the **Dashboard**, where they click on the **My Profile** option in the sidebar.
   * The **Profile screen** is loaded, displaying the user’s profile information.
   * The user can view their details and update fields where applicable.
   * If the user modifies any data, they click **Update Profile** to save the changes.
5. **Alternative Flow**:
   * If the user is not logged in and attempts to access the profile page, they are redirected to the login page.
   * If the user tries to submit invalid data (e.g., incorrect email format), the system will display an error message and prompt the user to correct the information before submitting again.

This **View Profile screen** is a crucial part of the OCMS that allows users to manage and review their personal data securely after authentication.

A screenshot of a computer

Description automatically generated

### 1.7 Edit Profile

The **Edit Profile screen** in the Online Course Management System (OCMS) allows users to view and modify their profile information. This screen is accessed after the user has logged in, navigated to the **Dashboard**, and clicked on the **My Profile** option. The user can update certain fields like their gender or other personal details and save changes by clicking the **Update Profile** button.

**Screen Specifications**

1. **Field Initializations**:
   * **Email Field**: Pre-filled with the user's registered email address. This field is typically non-editable for security reasons.
   * **Role Field**: Displays the user’s role (e.g., Student, Admin, Instructor). This field is non-editable as the role is assigned by the system.
   * **Gender Field**: Pre-selected based on the user’s current profile information but can be updated. Users can select their gender from a dropdown list.
   * **Last Login Field**: Displays the timestamp of the user's last login. This field is non-editable and only for informational purposes.
2. **Update Profile Button**:
   * When clicked, this button saves the changes made to the editable fields. Once submitted, the system checks for any validation errors and updates the user’s profile if all inputs are valid.
3. **Business Rules**:
   * **Validation**: Before saving any changes, the system ensures that all fields follow the correct format (e.g., valid email, selection from dropdown lists). Although the email cannot be changed on this screen, all other fields must be validated.
   * **Editable Fields**: Only certain fields like **gender** are editable. Fields like **email** and **role** are locked and cannot be changed by the user.
   * **Successful Update**: Upon clicking **Update Profile**, if validation is successful, the profile changes are saved, and the user may receive a confirmation message.
4. **Normal Flow**:
   * The user logs in and navigates to the **Dashboard**.
   * The user selects **My Profile** from the sidebar, landing on the **Edit Profile** screen.
   * The user makes changes to editable fields (e.g., gender) and clicks **Update Profile**.
   * The system validates the changes and saves the updated information, reflecting the changes on the profile page.
5. **Alternative Flow**:
   * If the user enters invalid data (e.g., selecting invalid options), the system will prompt an error message requesting the user to correct the input before proceeding.
   * If the user attempts to modify non-editable fields like **email** or **role**, the system will prevent the changes from being made.

The **Edit Profile screen** ensures users can manage and update their personal information securely, while ensuring that critical fields such as email and role remain protected from user changes.

A screenshot of a computer

Description automatically generated

### 1.8 Change password

The **Change Password screen** in the Online Course Management System (OCMS) allows authenticated users to update their account password. This screen is accessed from the user’s **Dashboard**, specifically by selecting the **Change Password** option from the sidebar. The purpose of this screen is to ensure users can securely modify their password by providing their current password and a new password, which is verified through a confirmation field.

**Screen Specifications**

1. **Field Initializations**:
   * **Current Password Field**: This field is empty by default and requires the user to input their existing password for verification.
   * **New Password Field**: Empty by default, expects the user to enter a new password.
   * **Confirm New Password Field**: Empty by default, expects the user to re-enter the new password for confirmation.
2. **Change Password Button**:
   * This button is disabled until all fields are filled. Once the fields are completed with valid data, the button becomes active, allowing the user to submit the password change request.
3. **Business Rules**:
   * **Current Password Validation**: The system checks if the entered current password matches the user’s existing password stored in the system.
   * **New Password Validation**: The new password must meet security criteria such as minimum length, inclusion of numbers or special characters, etc.
   * **Password Confirmation**: The system checks if the **New Password** and **Confirm New Password** fields match. If they do not match, an error message is displayed, prompting the user to re-enter the passwords.
   * **Password Reuse Prevention**: The system may have a rule that prevents the user from using the same password as their previous one.
4. **Normal Flow**:
   * The user logs in, navigates to the **Dashboard**, and selects the **Change Password** option from the sidebar.
   * The user enters their current password in the first field, followed by the new password in the second field, and confirms the new password in the third field.
   * The user clicks **Change Password**.
   * The system validates the current password and ensures the new password follows the security rules and matches the confirmation field.
   * If all validations pass, the system updates the password and displays a success message.
5. **Alternative Flow**:
   * If the **Current Password** is incorrect, the system displays an error message ("Incorrect current password"), and the user must try again.
   * If the **New Password** and **Confirm New Password** do not match, the system prompts the user to correct the input with an error message ("Passwords do not match").
   * If the new password does not meet security requirements, an error message is shown advising the user to follow the password guidelines (e.g., minimum length, special characters).

The **Change Password screen** ensures that users can securely update their passwords while adhering to security standards. The three-step process of entering the current password, new password, and confirming the new password ensures the integrity of the password update operation.

A screenshot of a computer

Description automatically generated

## 2. Instructor

### 2.1 Course

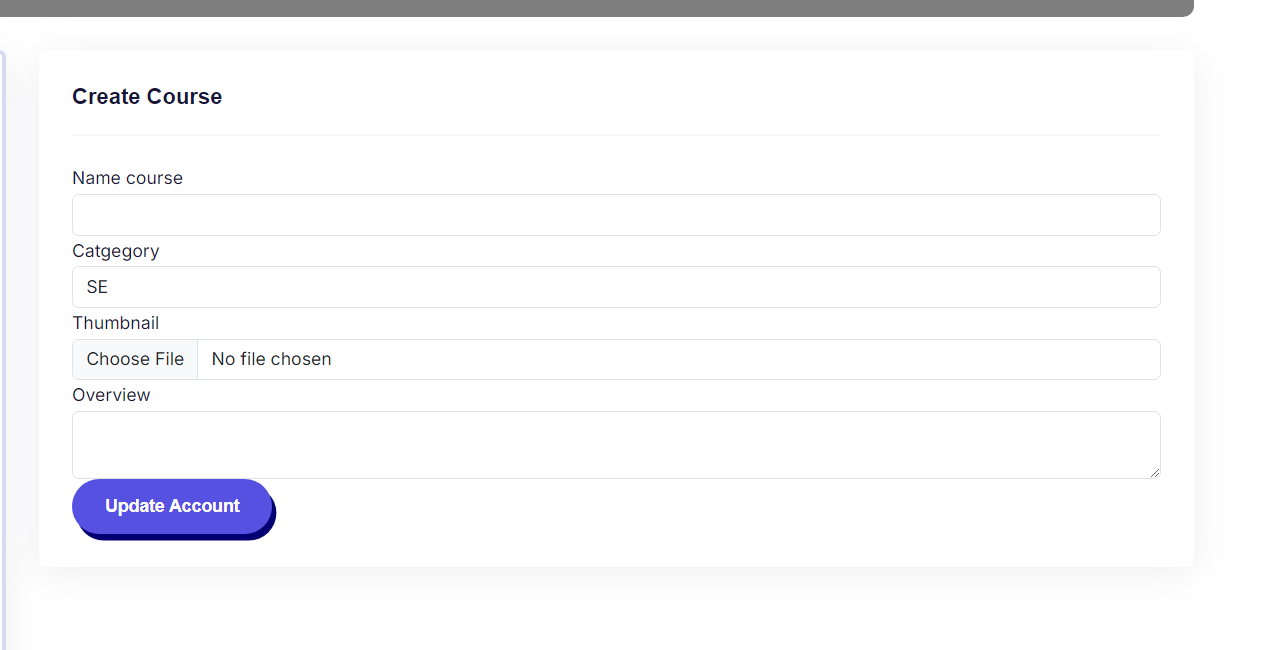
#### 2.1.1 Add course

The **Create Course screen** in the Online Course Management System (OCMS) is designed for instructors to create a new course. This screen allows instructors to fill in course details such as the course name, category, thumbnail image, and an overview of the course content. It is accessed by clicking the **Create A New Course** button available on the instructor's dashboard.

**Screen Specifications**

1. **Field Initializations**:
   * **Course Name Field**: Empty by default. Instructors are required to input the name of the course they want to create.
   * **Category Field**: Pre-filled with default categories (e.g., SE for Software Engineering). Instructors can choose a relevant category for the course.
   * **Thumbnail Field**: Empty by default. Instructors can upload an image file to serve as the course thumbnail. A "Choose File" button allows instructors to browse their computer and select an image file.
   * **Overview Field**: Empty by default. Instructors provide a brief description or overview of the course content to inform prospective students.
2. **Update Account Button** (likely meant as **Create Course**):
   * This button allows the instructor to submit the course details for creation. It is only enabled when all required fields are filled in correctly.
3. **Business Rules**:
   * **Required Fields**: The system checks if all required fields (e.g., course name, category) are filled before allowing the instructor to submit the course.
   * **Valid File Upload**: The thumbnail must be in an acceptable image format (e.g., JPEG, PNG). If the file is invalid, the system will display an error.
   * **Course Name Validation**: The course name must be unique and not match an existing course in the system. If a duplicate is detected, the instructor will be prompted to choose a different name.
4. **Normal Flow**:
   * The instructor clicks **Create A New Course** on the dashboard.
   * The **Create Course screen** is displayed, and the instructor fills in all fields, including the course name, category, thumbnail image, and overview.
   * The instructor clicks the **Update Account** (or **Create Course**) button to submit the course details.
   * The system validates the inputs and, if everything is correct, creates the course and adds it to the course catalogue.
5. **Alternative Flow**:
   * If any required fields are left blank, the system prevents the instructor from submitting the form and highlights the missing fields.
   * If the thumbnail image is in an unsupported format or not provided, the system displays an error and prompts the instructor to upload a valid image.
   * If the course name already exists in the system, the instructor will be prompted to select a different course name.

The **Create Course screen** ensures that instructors can efficiently create and manage their courses while adhering to system validation rules to ensure the course is properly categorized and presented to students.



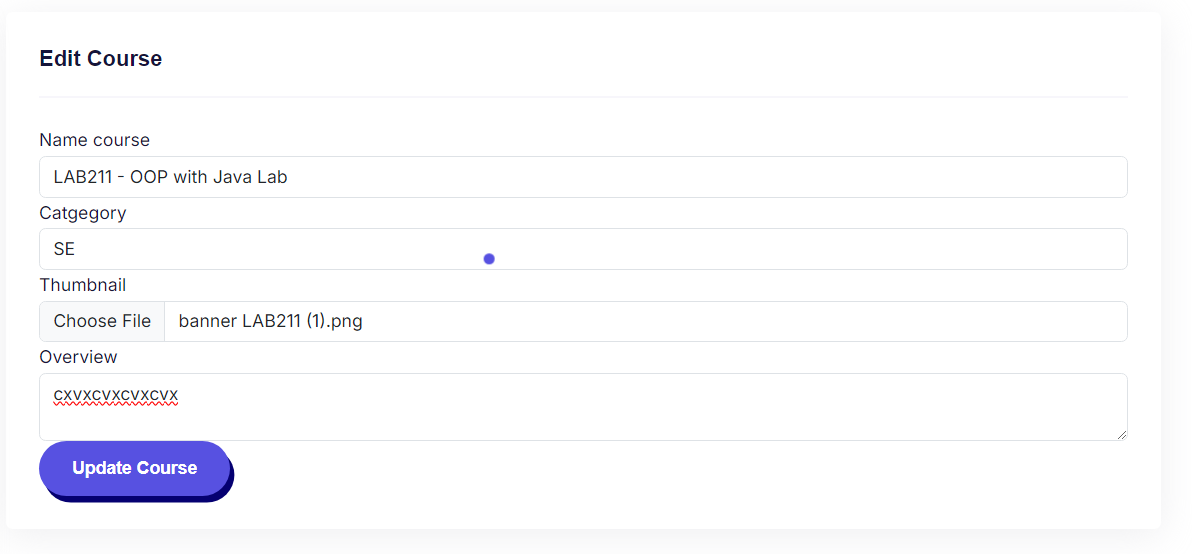
#### 2.1.2 Edit course

The **Edit Course screen** in the Online Course Management System (OCMS) allows instructors to update existing course details. The purpose of this screen is to enable instructors to modify key course attributes such as the course name, category, thumbnail image, and course overview. This screen can be accessed from the instructor’s dashboard or course management panel, where they can select the course they want to edit.

**Screen Specifications**

1. **Field Initializations**:
   * **Name Course Field**: Pre-filled with the current name of the course (e.g., "LAB211 - OOP with Java Lab"). The instructor can modify this field to change the course name.
   * **Category Field**: Pre-filled with the existing category (e.g., "SE" for Software Engineering). The instructor can choose to keep or change the category from the dropdown menu.
   * **Thumbnail Field**: Displays the current thumbnail image file associated with the course (e.g., "banner LAB211 (1).png"). The instructor can upload a new image if necessary by selecting a new file.
   * **Overview Field**: Pre-filled with the current course description or overview. The instructor can edit or update the content to reflect changes to the course.
2. **Update Course Button**:
   * This button allows the instructor to save the changes made to the course. The button becomes active once all required fields are correctly filled.
3. **Business Rules**:
   * **Required Fields**: The system checks that all mandatory fields, such as course name and category, are filled before allowing the instructor to update the course.
   * **Unique Course Name**: The system verifies that the course name is unique across the system. If a duplicate name exists, an error message will prompt the instructor to choose a different name.
   * **Valid Thumbnail Upload**: The system checks that the thumbnail image is in an accepted format (e.g., PNG, JPEG). If the file is invalid, an error message is displayed.
   * **Course Overview**: The course overview field should not exceed a certain character limit (if defined) to ensure concise course descriptions.
4. **Normal Flow**:
   * The instructor navigates to the course they want to edit and selects the **Edit Course** option.
   * The screen is populated with the current course details.
   * The instructor makes the necessary changes to the course name, category, thumbnail, or overview.
   * The instructor clicks the **Update Course** button.
   * The system validates the inputs and updates the course information if all fields are correctly filled.
5. **Alternative Flow**:
   * If any required fields are left blank or if validation fails (e.g., duplicate course name, invalid file format), the system will display an error message and prevent the instructor from submitting the form.
   * If the instructor decides not to proceed with changes, they can navigate away from the screen without saving, and the course details will remain unchanged.

This **Edit Course screen** ensures that instructors can efficiently modify and maintain their courses by providing an intuitive interface for updating course details, while adhering to system validation rules.



#### 2.1.3 View course details

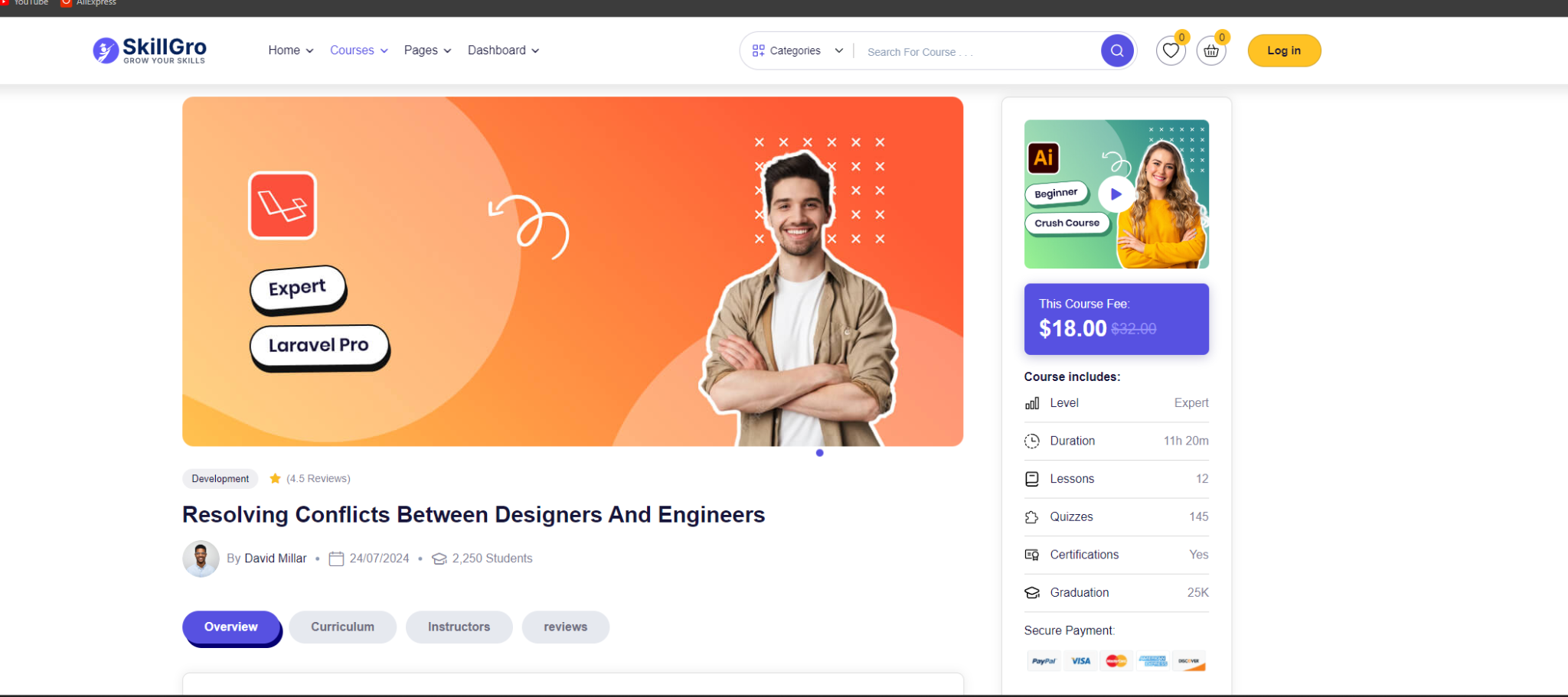
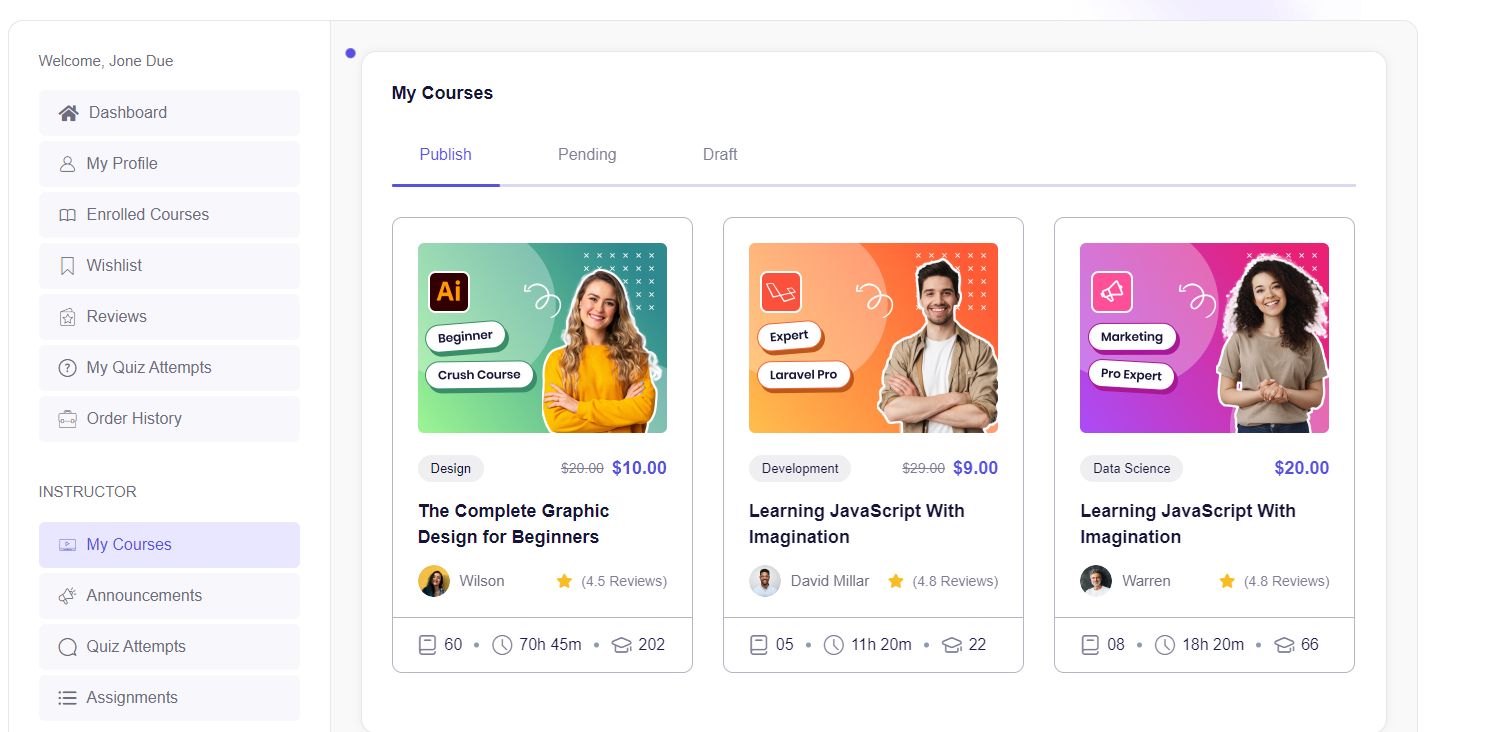
**Screen/Function Description**

The **View Course screen** in the Online Course Management System (OCMS) provides a detailed overview of a specific course. It is accessed when a student or visitor clicks on a course from the course catalogue. The screen allows the user to review information about the course, including its title, instructor, pricing, duration, and more. The purpose of this screen is to offer an in-depth view of the course's content, instructors, and relevant information before enrolment or purchase.

**Screen Specifications**

1. **Field Initializations**:
   * **Course Title**: Displays the name of the course (e.g., "Resolving Conflicts Between Designers And Engineers").
   * **Instructor Name**: Shows the name of the instructor (e.g., "David Millar"), along with the instructor's profile picture.
   * **Course Price**: Displays the current price, often with a discounted price if applicable (e.g., "$18.00" discounted from "$32.00").
   * **Course Rating**: Displays an average rating out of 5 stars based on student reviews (e.g., "4.5 Reviews").
   * **Course Duration**: Displays the total duration of the course content (e.g., "11h 20m").
   * **Lessons**: Shows the number of lessons in the course (e.g., "12 Lessons").
   * **Quizzes**: Indicates whether the course includes quizzes (e.g., "145 Quizzes").
   * **Certifications**: Displays whether the course offers certification upon completion (e.g., "Certifications: Yes").
   * **Graduation**: Displays the number of students who have graduated from the course (e.g., "25K Graduations").
   * **Secure Payment Options**: Displays available payment methods for purchasing the course (e.g., PayPal, Visa, Mastercard).
2. **Course Overview Section**:
   * **Overview Tab**: Provides a general introduction and description of the course content.
   * **Curriculum Tab**: Lists the curriculum, including topics covered, modules, and lessons.
   * **Instructors Tab**: Displays information about the instructors teaching the course.
   * **Reviews Tab**: Shows student reviews and feedback.
3. **Sidebar Specifications**:
   * **Course Thumbnail**: Displays the course's thumbnail image, giving a visual representation of the course (e.g., for "Resolving Conflicts Between Designers And Engineers").
   * **Course Price and Purchase Button**: Displays the discounted course price with a purchase button that allows the user to enroll or purchase the course.
   * **Additional Courses**: Shows related or recommended courses that users might also be interested in.
4. **Business Rules**:
   * **Pricing**: The system applies discounts if available, displaying both the original and discounted prices.
   * **Enrolment Validation**: Users must be logged in to purchase or enroll in the course. If the user is not logged in, clicking on the purchase button will prompt them to sign in.
   * **Secure Payment**: The system only allows purchase if a valid payment method is provided (e.g., PayPal or credit card).
   * **Access Control**: Users who have already enrolled in the course will see a "Go to Course" button instead of a purchase button.
5. **Normal Flow**:
   * The user navigates to the course catalogue and clicks on a specific course.
   * The **View Course** screen is displayed, showing all relevant course details.
   * The user reviews the course description, curriculum, and instructor information.
   * If the user is interested, they click the purchase button and proceed with the payment process.
   * Upon successful payment, the user is enrolled in the course and redirected to the course content.
6. **Alternative Flow**:
   * If the user is not logged in and attempts to purchase the course, they will be prompted to log in before continuing.
   * If the payment fails, an error message will be displayed, and the user will be prompted to try again with a valid payment method.

This **View Course screen** provides a detailed and informative overview of the course, ensuring that users can make informed decisions before enrolling in or purchasing a course. The screen also includes user-friendly navigation and secure payment options for a seamless experience.



### 2.2 Lesson

#### 2.2.1 Create lesson

**Screen/Function Description**

The **Create Lesson screen** in the Online Course Management System (OCMS) allows instructors to upload and organize lesson content for their courses. This screen supports the creation of lessons by enabling the instructor to upload files (e.g., PDFs, Word documents) and video content (e.g., MP4 videos) that will be accessible to students enrolled in the course. The purpose of this screen is to facilitate the addition of learning materials for each course module.

**Screen Specifications**

1. **Field Initializations**:
   * **Lesson Title Field**: Empty by default. This field expects the instructor to input the title of the lesson (e.g., "Introduction to Java").
   * **Lesson Description Field**: Empty by default. The instructor provides a brief description of what the lesson covers (e.g., "This lesson introduces basic concepts of Java programming").
   * **File Upload Field**: Empty by default. Instructors can upload supporting documents such as PDFs, Word documents, or PowerPoint presentations. A "Choose File" button allows instructors to browse and select a file from their computer.
   * **Video Upload Field**: Empty by default. Instructors can upload a video file (e.g., MP4). A "Choose File" button allows instructors to browse and select the video file from their computer.
   * **Lesson Duration Field**: Empty by default. Instructors enter the estimated time required for students to complete the lesson (e.g., "45 minutes").
2. **Create Lesson Button**:
   * This button allows the instructor to submit the lesson details and upload the selected files. It is only enabled when all required fields are filled correctly and at least one file (video or document) is uploaded.
3. **Business Rules**:
   * **Required Fields**: The lesson title and description fields are mandatory. If they are left blank, the system will not allow the submission of the lesson.
   * **Valid File Uploads**: The system checks that the uploaded files are in supported formats (e.g., PDF, DOCX for files and MP4, AVI for videos). If an unsupported format is uploaded, the system will display an error message prompting the instructor to upload a valid file.
   * **File Size Limits**: The system enforces file size limits (e.g., videos cannot exceed 500MB, documents cannot exceed 10MB). If the file size exceeds the limit, the system prompts the instructor to upload a smaller file.
   * **Optional Content**: Instructors can upload either a file, a video, or both. However, at least one type of content must be uploaded to create the lesson.
   * **Lesson Duration**: The lesson duration must be provided in a valid time format (e.g., minutes or hours). The system validates this before allowing submission.
4. **Normal Flow**:
   * The instructor navigates to the course they are managing and selects the **Create Lesson** option.
   * The instructor fills in the lesson title and description fields.
   * The instructor uploads a supporting document file and/or a video file.
   * The instructor inputs the lesson duration (e.g., "30 minutes").
   * The instructor clicks the **Create Lesson** button.
   * The system validates the inputs, checks the file types, and uploads the lesson. The lesson is then added to the course curriculum.
5. **Alternative Flow**:
   * If the required fields (lesson title, description) are left blank, the system prevents the submission and highlights the missing fields.
   * If an unsupported file type is uploaded, an error message is displayed, prompting the instructor to upload a valid file.
   * If the file size exceeds the allowed limit, an error message prompts the instructor to upload a smaller file.
   * If the instructor decides not to proceed with the lesson creation, they can cancel the operation, and the system will discard the inputs without saving the lesson.

This **Create Lesson screen** ensures that instructors can efficiently upload and manage lesson materials while adhering to system validations for file formats and lesson requirements. The screen facilitates content creation in a structured and user-friendly manner, allowing for a smooth lesson development process.

#### 2.2.2 View lesson details

**Screen/Function Description**

The **View Lesson Details screen** in the Online Course Management System (OCMS) allows students or users to view the detailed content of a specific lesson within a course. This screen provides access to both file resources (e.g., PDFs, Word documents) and video lectures associated with the lesson. The purpose of the screen is to enable students to engage with learning materials, download lesson files, and watch video content as part of their coursework.

**Screen Specifications**

1. **Field Initializations**:
   * **Lesson Title Field**: Displays the title of the lesson, initialized with the lesson name provided by the instructor (e.g., "Introduction to Java").
   * **Lesson Description Field**: Displays a brief description of the lesson, initialized with the information entered by the instructor during lesson creation (e.g., "This lesson introduces basic concepts of Java programming").
   * **File Download Section**: Displays a list of downloadable files associated with the lesson. Each file is shown with its filename and format (e.g., "Introduction\_to\_Java.pdf"). Students can click a download button to save the file to their device.
   * **Video Player Section**: Displays a video player where students can watch the video content related to the lesson. The video is initialized and embedded for direct streaming, with controls for play, pause, and seek functions.
   * **Lesson Duration Field**: Displays the estimated duration for the lesson (e.g., "45 minutes").
2. **Additional Navigation**:
   * **Previous/Next Lesson Buttons**: Allows students to navigate to the previous or next lesson in the course. These buttons are initialized based on the lesson sequence within the course.
   * **Mark as Complete Button**: Provides a button for students to mark the lesson as completed once they have finished reviewing the content.
3. **Business Rules**:
   * **File Access**: Only enrolled students or authorized users can view and download lesson files. If a user is not enrolled in the course, the system prompts them to enrol or prevents access to the lesson.
   * **Video Streaming**: The video must be in a supported format (e.g., MP4) and should load seamlessly using the embedded video player. If the video file is too large or unsupported, an error message will be shown, preventing playback.
   * **Lesson Progress Tracking**: When a student clicks the **Mark as Complete** button, the system records the student's progress in the course. If the lesson is completed, it will be reflected in the student's progress tracker.
   * **Access Control**: Users must be logged in and enrolled in the course to access the **View Lesson Details** screen. Unauthenticated users will be redirected to the login page, and unenrolled users will be prompted to sign up for the course.
4. **Normal Flow**:
   * The student navigates to the course and selects a specific lesson to view.
   * The **View Lesson Details** screen is loaded, displaying the lesson title, description, downloadable files, and video content.
   * The student downloads files, watches the video, and interacts with the content as needed.
   * Upon completing the lesson, the student clicks **Mark as Complete** to record their progress.
   * The student can use the **Previous** or **Next Lesson** buttons to move between lessons in the course.
5. **Alternative Flow**:
   * If the student is not logged in, they are redirected to the login page before accessing the lesson.
   * If the student is not enrolled in the course, they are prompted to enroll before viewing the lesson content.
   * If there are issues with video playback (e.g., unsupported format, video not loading), an error message will be displayed, and the student will not be able to watch the video. The instructor may need to resolve this issue by uploading a compatible video format.
   * If the **Mark as Complete** button fails to record progress due to system issues, the student may need to refresh the page or try again later.

This **View Lesson Details screen** provides a comprehensive view of lesson content, combining video and file resources to support learning. It allows students to seamlessly engage with course materials, ensuring their progress is tracked and their access is secure.

#### 2.2.3 Update lesson

**Screen/Function Description**

The **Edit Lesson screen** in the Online Course Management System (OCMS) is designed for instructors to modify the content and details of an existing lesson. The purpose of this screen is to allow instructors to update the lesson title, description, upload or replace associated files (e.g., PDFs, Word documents), and change or update video content (e.g., MP4 files). This screen ensures that instructors can maintain up-to-date and accurate lesson content for their students.

**Screen Specifications**

1. **Field Initializations**:
   * **Lesson Title Field**: Pre-filled with the existing title of the lesson. Instructors can update the lesson name as needed (e.g., "Introduction to Java Basics").
   * **Lesson Description Field**: Pre-filled with the current lesson description. Instructors can modify the description to reflect any changes in the lesson content (e.g., "This lesson introduces the fundamental concepts of Java programming including variables, data types, and control flow.").
   * **File Upload Section**: Displays the currently uploaded file(s), if any. Instructors can choose to upload new files to replace or supplement the existing resources. A "Choose File" button allows instructors to browse and upload new document files (e.g., PDFs, DOCX).
   * **Video Upload Section**: Displays the currently associated video file, if any. Instructors can upload a new video to replace the current one. A "Choose File" button is available for browsing and uploading video files (e.g., MP4 format).
   * **Lesson Duration Field**: Pre-filled with the current estimated lesson duration. The instructor can modify the duration based on changes made to the lesson content (e.g., changing from "30 minutes" to "45 minutes").
2. **Update Lesson Button**:
   * This button allows the instructor to save the changes made to the lesson. It is only enabled when all required fields are filled and the content is valid. If any mandatory fields are left blank, the button remains disabled until corrected.
3. **Business Rules**:
   * **Required Fields**: Lesson title and description fields must be filled out. The system will not allow the lesson to be updated if these fields are left blank.
   * **Valid File Uploads**: The system ensures that all uploaded files, whether documents or videos, are in supported formats (e.g., PDF, DOCX for files; MP4 for videos). If an unsupported file format is uploaded, an error message prompts the instructor to upload a valid file.
   * **File Size Limits**: There may be file size limits for video and document uploads (e.g., videos cannot exceed 500MB and documents cannot exceed 10MB). If a file exceeds the allowed size, the system will show an error message and prevent the upload.
   * **Optional Content**: Instructors are not required to upload both a file and a video, but at least one form of content must be included to update the lesson. The instructor can choose to only upload a file, only upload a video, or upload both.
   * **Lesson Duration**: The lesson duration must be provided in a valid format (e.g., minutes or hours). The system validates this input before saving the lesson.
4. **Normal Flow**:
   * The instructor navigates to the lesson they want to edit and selects the **Edit Lesson** option.
   * The **Edit Lesson** screen is populated with the current details of the lesson (title, description, file(s), video, duration).
   * The instructor makes the necessary changes to the lesson title, description, and uploads new files or videos.
   * The instructor clicks the **Update Lesson** button to save the changes.
   * The system validates the inputs, processes the uploaded content, and updates the lesson details accordingly.
5. **Alternative Flow**:
   * If the instructor leaves mandatory fields (e.g., lesson title or description) blank, the system highlights the missing fields and prevents submission until corrected.
   * If the instructor uploads unsupported file formats or files that exceed the allowed size, an error message will be displayed, prompting the instructor to upload a valid file.
   * If the system fails to save changes due to connectivity issues or other technical problems, the instructor may need to retry the process.

The **Edit Lesson screen** allows instructors to seamlessly update and improve lesson content, ensuring that students receive the most accurate and relevant materials. The screen ensures smooth file and video management while enforcing system validation rules to maintain high content standards.

#### 2.2.4 Delete lesson

**Screen/Function Description**

The **Delete Lesson screen** in the Online Course Management System (OCMS) allows instructors to remove an entire lesson, including any associated files (e.g., PDFs, Word documents) and video content (e.g., MP4 files) from a course. The purpose of this screen is to provide a way for instructors to permanently delete lessons that are outdated, no longer needed, or incorrect. Deleting a lesson removes it from the course structure, making it inaccessible to students.

**Screen Specifications**

1. **Field Initializations**:
   * **Lesson Title Display**: Displays the title of the lesson that is about to be deleted. This field is non-editable and initialized with the current lesson title (e.g., "Lesson 3: Introduction to Java Basics").
   * **Lesson Description Display**: Shows the description of the lesson that will be deleted. This field is also non-editable and initialized with the existing description (e.g., "This lesson covers the basic principles of Java programming").
   * **Associated Files and Video**: Displays the list of files and videos associated with the lesson. This includes document filenames (e.g., "Introduction\_to\_Java.pdf") and video files (e.g., "Java\_Introduction.mp4").
   * **Confirmation Message**: A message appears informing the instructor that this action will permanently delete the lesson, including all related files and videos, and it cannot be undone (e.g., "Are you sure you want to delete this lesson and all associated content? This action cannot be undone.").
2. **Delete Lesson Button**:
   * This button is only enabled once the instructor confirms the deletion action. When clicked, the system permanently deletes the lesson and its related content. The button remains disabled until the instructor acknowledges the confirmation message by selecting a checkbox or similar element (e.g., "I understand that this action is permanent").
3. **Business Rules**:
   * **Deletion Confirmation**: To prevent accidental deletion, the instructor must confirm their intention by selecting a checkbox or clicking a confirmation link. If this confirmation is not provided, the **Delete Lesson** button remains disabled.
   * **Lesson Removal**: When the **Delete Lesson** button is clicked, the system permanently removes the lesson, its files, and video content from the course database and all associated student views. Once deleted, the lesson cannot be recovered.
   * **Access Control**: Only authorized instructors or administrators have the permission to delete lessons. If the instructor does not have the correct permissions, they will not be able to access this screen.
   * **Content Dependencies**: If other lessons or modules are dependent on this lesson, the system will notify the instructor and may prevent deletion until dependencies are resolved.
4. **Normal Flow**:
   * The instructor navigates to the lesson they wish to delete and selects the **Delete Lesson** option.
   * The **Delete Lesson** screen displays the lesson title, description, and associated content (files and videos).
   * The instructor reviews the content and confirms that they want to delete the lesson by selecting the confirmation checkbox.
   * The instructor clicks the **Delete Lesson** button.
   * The system permanently deletes the lesson and its associated content from the course.
5. **Alternative Flow**:
   * If the instructor does not confirm the deletion by selecting the confirmation checkbox, the **Delete Lesson** button remains disabled, and the lesson is not deleted.
   * If the system detects that the lesson has dependencies (e.g., quizzes, assignments, or modules relying on the lesson), the instructor will receive an error message or a prompt to resolve the dependencies before proceeding.
   * If there is a system failure or technical issue, the deletion process will not be completed, and the instructor may need to retry the operation later.

The **Delete Lesson screen** is essential for maintaining the course content, ensuring that instructors can remove outdated or unnecessary lessons while protecting against accidental deletions. It includes a confirmation process to ensure that lesson removal is deliberate and irreversible.

### 2.3 Assignment

#### 2.3.1 Create assignment

**Screen/Function Description**

The **Create Assignment screen** in the Online Course Management System (OCMS) allows instructors to design and upload assignment tasks for students. Instructors can attach files (e.g., PDFs, Word documents) and videos (e.g., MP4) as part of the assignment instructions. The purpose of this screen is to facilitate the creation of detailed assignments, ensuring that students can access all necessary resources for completing the task. Instructors can specify deadlines, assignment details, and upload relevant media files to support the assignment.

**Screen Specifications**

1. **Field Initializations**:
   * **Assignment Title Field**: Empty by default. This field requires the instructor to input the name of the assignment (e.g., "Assignment 1: Java Basics").
   * **Assignment Description Field**: Empty by default. The instructor provides detailed instructions and guidelines for the assignment (e.g., "Complete the coding exercises and submit your solution as a .zip file.").
   * **File Upload Field**: Empty by default. Instructors can upload supporting documents such as PDFs, Word documents, or PowerPoint presentations to provide further instructions or materials. A "Choose File" button allows instructors to browse and select a file from their computer.
   * **Video Upload Field**: Empty by default. Instructors can upload a video file (e.g., MP4) that students can watch as part of the assignment. A "Choose File" button allows instructors to browse and select the video file from their computer.
   * **Assignment Deadline Field**: Empty by default. Instructors specify a deadline for the assignment, using a date-picker or time-input field to set the submission date (e.g., "October 30, 2024, 11:59 PM").
2. **Create Assignment Button**:
   * This button becomes enabled once the instructor has filled in all required fields (title, description, deadline) and uploaded at least one file or video. Clicking the button saves the assignment and makes it available to students.
3. **Business Rules**:
   * **Required Fields**: The assignment title, description, and deadline fields are mandatory. The system will not allow the instructor to create the assignment if any of these fields are left blank.
   * **Valid File Uploads**: The system supports document uploads in formats such as PDF, DOCX, and PPTX, and videos in MP4 or AVI formats. If an unsupported format is uploaded, an error message will prompt the instructor to upload a valid file.
   * **File Size Limits**: Files must adhere to the system's size restrictions (e.g., document size cannot exceed 10MB, and video size cannot exceed 500MB). If a file exceeds these limits, the system displays an error, and the instructor must choose a smaller file.
   * **Deadline Enforcement**: The deadline must be a valid future date. The system prevents the instructor from setting a past date or leaving the deadline field blank.
   * **Optional Content**: Instructors may upload either a file, a video, or both as part of the assignment. However, at least one content type must be uploaded to create the assignment.
4. **Normal Flow**:
   * The instructor accesses the course and selects the **Create Assignment** option.
   * The instructor enters the assignment title, description, and deadline.
   * The instructor uploads necessary files and/or videos to support the assignment.
   * The instructor clicks the **Create Assignment** button.
   * The system validates the inputs, ensuring all required fields are filled, and then creates the assignment. The assignment becomes available for students in the course to view and complete.
5. **Alternative Flow**:
   * If the instructor leaves any required fields (e.g., title, description, deadline) blank, the system highlights the missing fields and prevents the submission.
   * If the instructor uploads unsupported file types or files that exceed the size limits, an error message will appear, prompting the instructor to upload valid files.
   * If the system fails to save the assignment due to technical issues (e.g., network connectivity), the instructor may need to retry the process later.

The **Create Assignment screen** provides a comprehensive and user-friendly interface for instructors to design assignments with supporting materials, ensuring that students have all necessary resources to complete the tasks. The screen enforces strict validation rules to maintain consistency and quality in assignment creation.

#### 2.3.2 Edit assignment

Screen/Function Description

The **Edit Assignment screen** in the Online Course Management System (OCMS) allows instructors to modify an existing assignment, including changing the title, description, deadline, and any attached files or videos. The purpose of this screen is to enable instructors to update assignment instructions or resources to ensure students have the correct and up-to-date materials for completing their tasks. Instructors can replace or add new files and videos, update deadlines, and revise the assignment description as needed.

**Screen Specifications**

1. **Field Initializations**:
   * **Assignment Title Field**: Pre-filled with the current assignment title. Instructors can edit this field to update the name of the assignment (e.g., "Assignment 1: Java Basics").
   * **Assignment Description Field**: Pre-filled with the current assignment description. Instructors can modify the text to revise or provide additional instructions for the assignment (e.g., "This updated assignment includes additional coding exercises.").
   * **File Upload Section**: Displays the current file(s) associated with the assignment, if any. Instructors can upload new files to replace or supplement the existing documents. A "Choose File" button allows instructors to browse and select a file (e.g., PDF, DOCX).
   * **Video Upload Section**: Displays the current video, if applicable. Instructors can upload a new video to replace the existing one. A "Choose File" button allows instructors to browse and upload the video (e.g., MP4).
   * **Assignment Deadline Field**: Pre-filled with the existing deadline. Instructors can change the deadline using a date-picker or time-input field to set a new due date (e.g., "October 30, 2024, 11:59 PM").
2. **Update Assignment Button**:
   * This button is enabled once the instructor has completed the required updates. Clicking the button saves the changes to the assignment and updates it in the system for students.
3. **Business Rules**:
   * **Required Fields**: The assignment title, description, and deadline must be filled. The system will not allow the instructor to save changes if any of these fields are left blank.
   * **Valid File Uploads**: Files must be in supported formats (e.g., PDF, DOCX for documents and MP4 for videos). If an unsupported file type is uploaded, the system will display an error message, prompting the instructor to upload a valid file.
   * **File Size Limits**: Files must comply with the size limits imposed by the system (e.g., document files cannot exceed 10MB, video files cannot exceed 500MB). If the file exceeds these limits, an error message will be displayed.
   * **Deadline Validation**: The deadline must be a future date. The system will prevent the instructor from setting a past deadline or leaving the deadline field empty.
   * **Optional Content**: Instructors can modify either files, videos, or both. However, they must ensure that the assignment has at least one content type (file or video) to remain valid.
4. **Normal Flow**:
   * The instructor navigates to the course and selects the **Edit Assignment** option for the specific assignment they want to modify.
   * The screen is populated with the current assignment details, including the title, description, attached files or videos, and the existing deadline.
   * The instructor makes the necessary changes to the assignment fields (e.g., updates the title, changes the description, or uploads new resources).
   * The instructor clicks the **Update Assignment** button.
   * The system validates the inputs, saves the changes, and updates the assignment for students to view.
5. **Alternative Flow**:
   * If the instructor leaves any mandatory fields (e.g., title, description, deadline) blank, the system will prevent the submission and highlight the fields that need attention.
   * If the instructor uploads unsupported file formats or files that exceed the size limit, an error message will appear, prompting the instructor to upload valid files.
   * If there is a system failure or network issue, the instructor will need to retry saving the updates after resolving the issue.

The **Edit Assignment screen** ensures that instructors can easily update and maintain their assignments while providing flexibility to modify attached files and videos. The screen enforces validation rules to ensure that students receive accurate and up-to-date assignment information, enabling a smooth course management process.

#### 2.3.3 Delete assignment

**Screen/Function Description**

The **Delete Assignment screen** in the Online Course Management System (OCMS) allows instructors to permanently remove an existing assignment from a course, including all associated files (e.g., PDFs, Word documents) and video content (e.g., MP4). The purpose of this screen is to provide a secure method for instructors to delete assignments that are outdated, incorrect, or no longer relevant. Deleting an assignment removes it from the student view and prevents further submissions for that task.

**Screen Specifications**

1. **Field Initializations**:
   * **Assignment Title Display**: Displays the title of the assignment that is about to be deleted. This field is non-editable and initialized with the current assignment name (e.g., "Assignment 1: Java Basics").
   * **Assignment Description Display**: Shows the description of the assignment that will be deleted. This field is non-editable and initialized with the current assignment description (e.g., "This assignment covers the basics of Java programming and OOP concepts").
   * **Associated Files and Video Display**: Displays the list of files and videos associated with the assignment. This includes document filenames (e.g., "Java\_Basics.pdf") and video files (e.g., "Java\_Intro.mp4").
   * **Confirmation Message**: A warning message appears notifying the instructor that deleting the assignment is permanent and cannot be undone (e.g., "Are you sure you want to permanently delete this assignment and all related content? This action cannot be undone.").
2. **Delete Assignment Button**:
   * This button remains disabled until the instructor confirms their intention to delete the assignment by checking a confirmation box or selecting a similar option. Once the confirmation is acknowledged, the button becomes active and allows the instructor to proceed with the deletion.
3. **Business Rules**:
   * **Deletion Confirmation**: To prevent accidental deletion, the instructor must confirm their intention to delete the assignment by selecting a checkbox or clicking a confirmation link (e.g., "I understand that this action is permanent"). Without this confirmation, the system will not allow the deletion process to proceed.
   * **Assignment Removal**: Upon clicking the **Delete Assignment** button, the system permanently removes the assignment, including any files or videos, from the course database and all associated student views. The deletion is irreversible.
   * **Access Control**: Only authorized instructors or administrators with appropriate permissions can delete an assignment. If an instructor does not have the necessary permissions, the system will not allow access to the delete function.
   * **Assignment Dependencies**: If the assignment has dependencies, such as quizzes, assignments, or grades related to it, the system will notify the instructor and may prevent the deletion until those dependencies are resolved.
4. **Normal Flow**:
   * The instructor navigates to the course and selects the **Delete Assignment** option for the specific assignment.
   * The **Delete Assignment** screen displays the assignment title, description, and associated content (files and videos).
   * The instructor reviews the details and selects the confirmation checkbox to acknowledge that the deletion is permanent.
   * The instructor clicks the **Delete Assignment** button.
   * The system removes the assignment and its associated content from the course.
5. **Alternative Flow**:
   * If the instructor does not confirm the deletion by selecting the confirmation checkbox, the **Delete Assignment** button remains disabled, and the assignment is not deleted.
   * If the system detects dependencies (e.g., related quizzes, grading, or student submissions) that prevent the assignment from being deleted, the instructor will receive a notification or error message. The instructor may need to address these dependencies before proceeding with deletion.
   * If there is a system failure or connectivity issue during the deletion process, the instructor may need to retry the operation after resolving the issue.

The **Delete Assignment screen** ensures that instructors can securely remove outdated or unnecessary assignments from a course. By requiring confirmation, the screen reduces the risk of accidental deletions while maintaining a clear and secure process for managing course content.

## 3. Student

### 3.1 Course

#### 3.1.1 Enroll course

**Screen/Function Description**

The **Enroll Course for Student screen** in the Online Course Management System (OCMS) allows students to enroll in available courses. This screen provides students with essential course details, pricing information, and payment options if the course is paid. The purpose of this screen is to facilitate the enrollment process, enabling students to join courses they are interested in by either directly enrolling or completing a purchase, if necessary.

**Screen Specifications**

1. **Field Initializations**:
   * **Course Title Display**: Displays the name of the course the student wishes to enroll in (e.g., "Introduction to Java Programming"). This field is non-editable and initialized based on the course selected.
   * **Course Description Display**: Shows a brief description of the course, including its objectives and key features (e.g., "Learn the basics of Java programming, including variables, control flow, and object-oriented principles").
   * **Course Fee Display**: Displays the cost of the course. If the course is free, it will show "Free." If the course is paid, the price will be shown (e.g., "$49.99").
   * **Payment Options**: For paid courses, a section is initialized that shows the available payment methods (e.g., credit/debit card, PayPal). The student can choose their preferred method of payment from this section.
   * **Enroll Now Button**: This button is enabled by default if the course is free. For paid courses, it is enabled once the student selects a payment method.
2. **Business Rules**:
   * **Prerequisite Check**: The system checks if the course has any prerequisites (e.g., prior courses or skills needed). If prerequisites are not met, the student will be informed, and enrollment will not proceed until prerequisites are fulfilled.
   * **Enrollment Limits**: If the course has a maximum number of students allowed, the system checks availability. If the course is full, the student will be notified, and enrollment will be blocked until a spot opens up.
   * **Payment Validation**: For paid courses, the system validates the payment method. If payment is unsuccessful (e.g., card declined or insufficient funds), the system will notify the student, and enrollment will not be completed.
   * **Free Course Enrollment**: For free courses, the student can enroll without entering payment details, and the **Enroll Now** button is available immediately.
   * **Confirmation Message**: Once enrollment is successful, the system displays a confirmation message informing the student they have successfully enrolled in the course.
3. **Normal Flow**:
   * The student navigates to the course catalog and selects the course they want to enroll in.
   * The **Enroll Course for Student** screen is displayed, showing the course title, description, fee, and payment options.
   * If the course is free, the student clicks the **Enroll Now** button and is immediately enrolled in the course.
   * If the course is paid, the student selects their preferred payment method, enters the necessary payment details, and clicks the **Enroll Now** button.
   * The system processes the enrollment and displays a confirmation message upon successful enrollment.
   * The student is redirected to their dashboard, where they can access the newly enrolled course.
4. **Alternative Flow**:
   * If the student has not met the course prerequisites, the system will display an error message indicating that enrollment cannot proceed until the prerequisites are fulfilled.
   * If the course is full, the system will prevent the student from enrolling and will provide a notification or a waitlist option, if available.
   * If the payment is unsuccessful (for paid courses), the system will notify the student and ask them to retry the payment or choose a different payment method.
   * If the student decides to cancel the enrollment process, they can navigate back to the course catalog or their dashboard without enrolling in the course.

The **Enroll Course for Student screen** ensures that students can easily join the courses they are interested in while handling any prerequisites, payment requirements, or enrollment limits. It provides a seamless and secure process for enrolling in both free and paid courses.

#### 3.1.2 Access course material

**Screen/Function Description**

The **Access Course Material screen** in the Online Course Management System (OCMS) provides students with access to all the learning materials associated with a course. This screen allows students to view, download, or interact with various types of content, including lecture notes, reading materials, videos, and assignments. The purpose of this screen is to serve as a central hub for students to engage with the course content as they progress through the course.

**Screen Specifications**

1. **Field Initializations**:
   * **Course Title Display**: Displays the title of the course the student is currently enrolled in (e.g., "Introduction to Java Programming"). This field is non-editable and is initialized with the name of the course selected from the student’s dashboard.
   * **Course Material List**: Initialized with all the available course materials, categorized by type (e.g., "Lecture Notes," "Videos," "Assignments"). Each item is clickable, allowing students to view or download the material (e.g., "Week 1: Java Basics - PDF").
   * **Video Player Section**: Displays any video lectures or tutorials associated with the lesson. The videos are embedded and can be streamed directly from the screen, allowing students to play, pause, rewind, or fast-forward content.
   * **Assignment Section**: If the course has active assignments, this section lists them with links to view or download the instructions. It may also include a due date for each assignment.
   * **Download All Materials Button**: Allows students to download all available materials for the course in a compressed format (e.g., ZIP). This button is only visible when multiple files are available for download.
2. **Business Rules**:
   * **Enrollment Check**: The system checks if the student is enrolled in the course. If they are not, access is denied, and they are redirected to the course catalog or prompted to enroll.
   * **Material Availability**: Only materials that have been made available by the instructor are displayed. The system hides future materials if they are scheduled for release at a later date.
   * **File Format Validation**: The system ensures that all uploaded files are in supported formats (e.g., PDFs, DOCX for documents, MP4 for videos). Students will not see or be able to download unsupported files.
   * **Access Control**: Materials are only accessible to students who have completed the necessary prerequisites for each module or lesson. If a module is locked, the system displays a message indicating what steps the student must take to unlock the material.
   * **Progress Tracking**: When students view or download course materials, the system tracks this activity to update their progress in the course. This data is displayed in their dashboard to help them monitor what content they have accessed.
3. **Normal Flow**:
   * The student navigates to the course and selects the **Access Course Material** option from their dashboard.
   * The screen displays the course title, along with a list of all available materials, such as lecture notes, reading materials, videos, and assignments.
   * The student clicks on a file or video to view or download the content.
   * The system opens the selected file or begins playing the video within the embedded player.
   * If the student wishes to download all materials, they click the **Download All Materials** button, and the system compiles the files into a ZIP folder for download.
   * The system tracks the student’s interaction with the course materials and updates their progress accordingly.
4. **Alternative Flow**:
   * If the student is not enrolled in the course, the system denies access to the screen and prompts the student to enroll in the course or redirects them back to the course catalog.
   * If certain materials are not yet available (e.g., future lessons), the system will hide or disable those items until they are released by the instructor.
   * If there is an issue with a specific file or video (e.g., file corruption, unsupported format), the system will display an error message and prevent the student from accessing that content. The student may need to contact the instructor for further assistance.

The **Access Course Material screen** provides students with a streamlined way to interact with the learning resources for their courses. It ensures that all materials are organized and easily accessible while maintaining security through enrollment checks and progress tracking. This screen helps students engage with course content efficiently and keeps them on track with their coursework.

#### 3.1.3 Submit assigment

**Screen/Function Description**

The **Submit Assignment screen** in the Online Course Management System (OCMS) allows students to upload and submit their completed assignments for evaluation by the instructor. The purpose of this screen is to provide a streamlined interface where students can submit files (e.g., PDFs, DOCX) or project files before the assignment deadline. It ensures that students can complete their assignments and submit them in a timely manner, while instructors receive and review them for grading.

**Screen Specifications**

1. **Field Initializations**:
   * **Assignment Title Display**: Displays the title of the assignment that the student is submitting (e.g., "Assignment 1: Java Basics"). This field is non-editable and initialized with the assignment name selected from the course page.
   * **Assignment Description Display**: Shows a brief description or instructions for the assignment, as provided by the instructor (e.g., "Complete the coding exercises and submit your solution as a .zip file"). This field is non-editable.
   * **File Upload Section**: Allows students to upload their completed assignments. A "Choose File" button lets students browse their local device and select the file they want to submit (e.g., PDFs, ZIP files containing code).
   * **Submission Status Display**: Displays whether an assignment has already been submitted (e.g., "Submitted on October 12, 2024"). If a student has already submitted the assignment, the status shows the date and time of the previous submission.
   * **Comments Section**: Provides an optional text box where students can add comments or notes to the instructor along with their submission (e.g., "Please review the attached code for any improvements").
   * **Submit Assignment Button**: This button becomes active once the student has selected a file for upload. Clicking this button submits the file for grading.
2. **Business Rules**:
   * **File Upload Validation**: The system only allows valid file formats to be uploaded (e.g., PDF, DOCX, ZIP). If the student attempts to upload an unsupported file type, the system will display an error and prevent submission.
   * **File Size Limits**: Uploaded files must adhere to system-imposed size limits (e.g., files cannot exceed 50MB). If a file exceeds this size, the system will reject the upload and display an error.
   * **Deadline Enforcement**: The system checks if the assignment is being submitted before or after the deadline. If the submission occurs after the deadline, the system may either block the submission or mark it as "Late," depending on the instructor’s settings.
   * **Multiple Submissions**: The system can allow or block multiple submissions depending on the assignment settings. If multiple submissions are allowed, the student can resubmit a revised file before the deadline.
   * **Resubmission Policy**: If resubmission is allowed, the system tracks each submission and shows the history of previous submissions to both the student and instructor.
3. **Normal Flow**:
   * The student navigates to the assignment in the course and selects the **Submit Assignment** option.
   * The screen displays the assignment title, description, and the file upload section.
   * The student clicks the **Choose File** button, selects the file they wish to upload, and optionally adds comments in the text box.
   * The student clicks the **Submit Assignment** button to upload their file.
   * The system validates the file type and size, then submits the assignment.
   * The student sees a confirmation message indicating that the assignment has been successfully submitted, and the **Submission Status** is updated with the submission date and time.
4. **Alternative Flow**:
   * If the student attempts to submit an unsupported file type or a file exceeding the size limit, the system will display an error and prevent submission until the correct file is uploaded.
   * If the assignment is submitted after the deadline, the system may either reject the submission or mark it as "Late," depending on the settings configured by the instructor.
   * If the student has already submitted an assignment and resubmission is allowed, they will be able to replace the previous submission before the deadline. If resubmission is not allowed, the system will block any further uploads.

The **Submit Assignment screen** ensures that students can easily and securely upload their assignments for instructor review. With built-in validation for file types, sizes, and deadlines, it provides a smooth submission process while allowing instructors to manage assignment submissions efficiently.

#### 3.1.4 Track progress

**Screen/Function Description**

The **Track Progress screen** in the Online Course Management System (OCMS) provides students with an overview of their progress in a specific course. The purpose of this screen is to allow students to monitor their advancement through lessons, assignments, quizzes, and overall course completion. It also offers insights into grades, completed tasks, and upcoming assignments or activities. This helps students stay organized and focused on their learning journey.

**Screen Specifications**

1. **Field Initializations**:
   * **Course Title Display**: Displays the title of the course for which the progress is being tracked (e.g., "Introduction to Java Programming"). This field is non-editable and is initialized based on the course selected from the student’s dashboard.
   * **Progress Bar**: A visual progress bar showing the percentage of course completion (e.g., "65% Completed"). The bar dynamically adjusts based on the number of completed lessons, quizzes, and assignments.
   * **Completed Lessons List**: Displays a list of lessons the student has completed. Each entry includes the lesson title (e.g., "Lesson 1: Variables and Data Types") and the completion date.
   * **Upcoming Assignments and Quizzes**: Displays a list of upcoming assignments and quizzes, along with their due dates (e.g., "Assignment 3: Object-Oriented Programming - Due October 15, 2024").
   * **Grade Overview Section**: Displays the student’s current grades for assignments and quizzes that have been graded (e.g., "Assignment 1: 85/100").
   * **Time Spent Tracker**: Displays the total time the student has spent on the course (e.g., "12 hours spent"). This field is updated as the student continues interacting with the course materials.
2. **Business Rules**:
   * **Progress Calculation**: The system calculates the progress percentage based on the total number of lessons, quizzes, and assignments in the course. It tracks completion of each element and updates the progress bar accordingly.
   * **Assignment and Quiz Deadlines**: The system automatically tracks upcoming assignments and quizzes, showing them in this section until the due date passes. If a due date is missed, the assignment or quiz is marked as "Overdue."
   * **Grade Updates**: The grade section is updated automatically after the instructor has graded a submission. It shows grades for each completed assignment and quiz. If an assignment is not yet graded, it will display as "Pending."
   * **Completion Criteria**: The course is considered complete when the student has finished all required lessons, quizzes, and assignments. Once complete, the system may display a message indicating that the course has been fully completed.
3. **Normal Flow**:
   * The student logs into the system and selects the course they want to track progress for.
   * The **Track Progress** screen displays the course title, the progress bar, and a breakdown of completed lessons, upcoming assignments, and current grades.
   * The student reviews their progress, checking completed tasks, upcoming assignments, and overall course performance.
   * The system updates the progress bar and completion percentage as the student finishes more lessons and submits assignments.
   * The student can return to the dashboard or continue with the course after reviewing their progress.
4. **Alternative Flow**:
   * If the student has not completed any lessons, quizzes, or assignments yet, the progress bar will display "0% Completed" and the **Completed Lessons List** will be empty.
   * If the student has missed deadlines, the system marks the relevant assignments or quizzes as "Overdue" and highlights them in the **Upcoming Assignments and Quizzes** section.
   * If there are grading delays from the instructor, the **Grade Overview Section** will display "Pending" next to the ungraded assignments or quizzes.

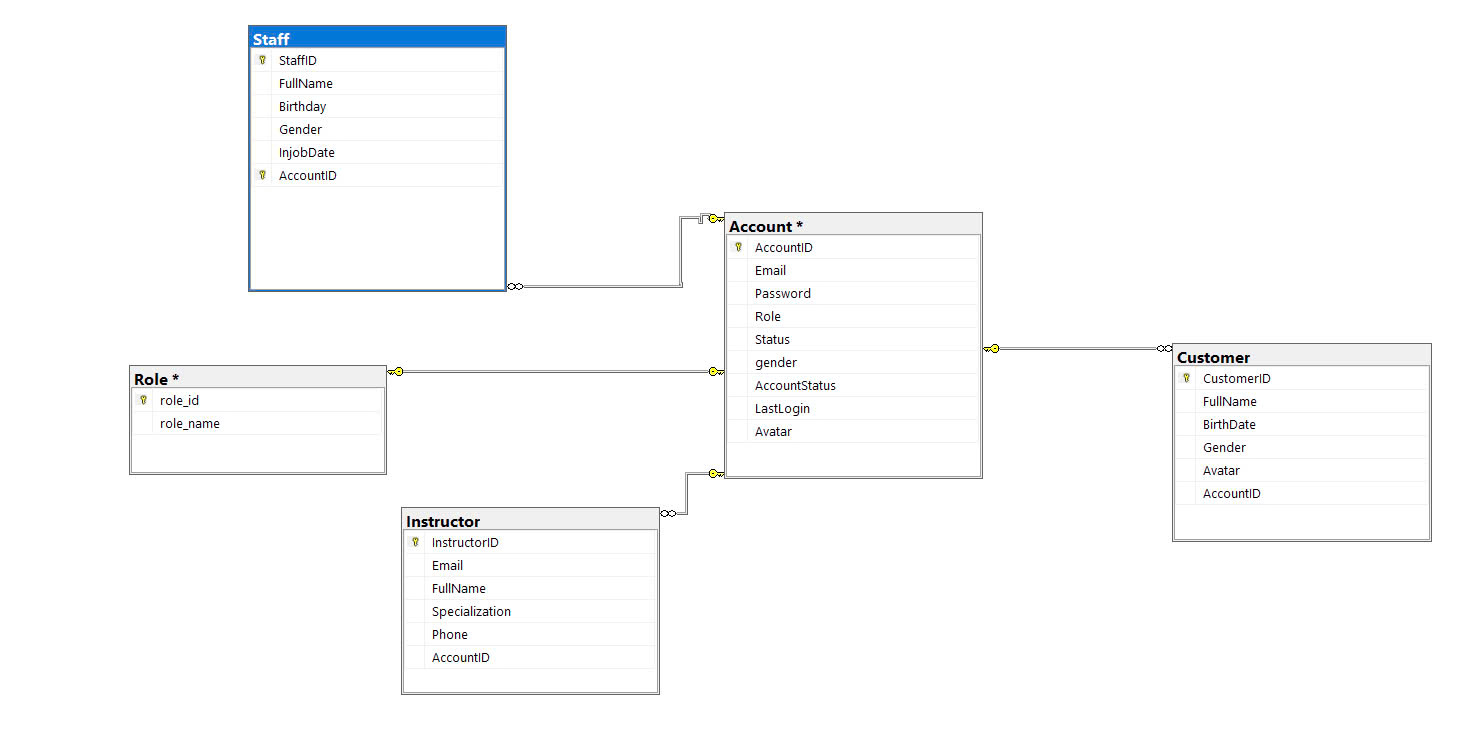
The **Track Progress screen** helps students stay informed about their learning progress, manage their tasks effectively, and see a clear overview of their performance in the course. It provides a structured way to ensure students stay on track and meet course completion requirements.

# III. System Design

## 1. Database Design

*[Provide the tables relationship like example below]*

### 1.1 Database Schema



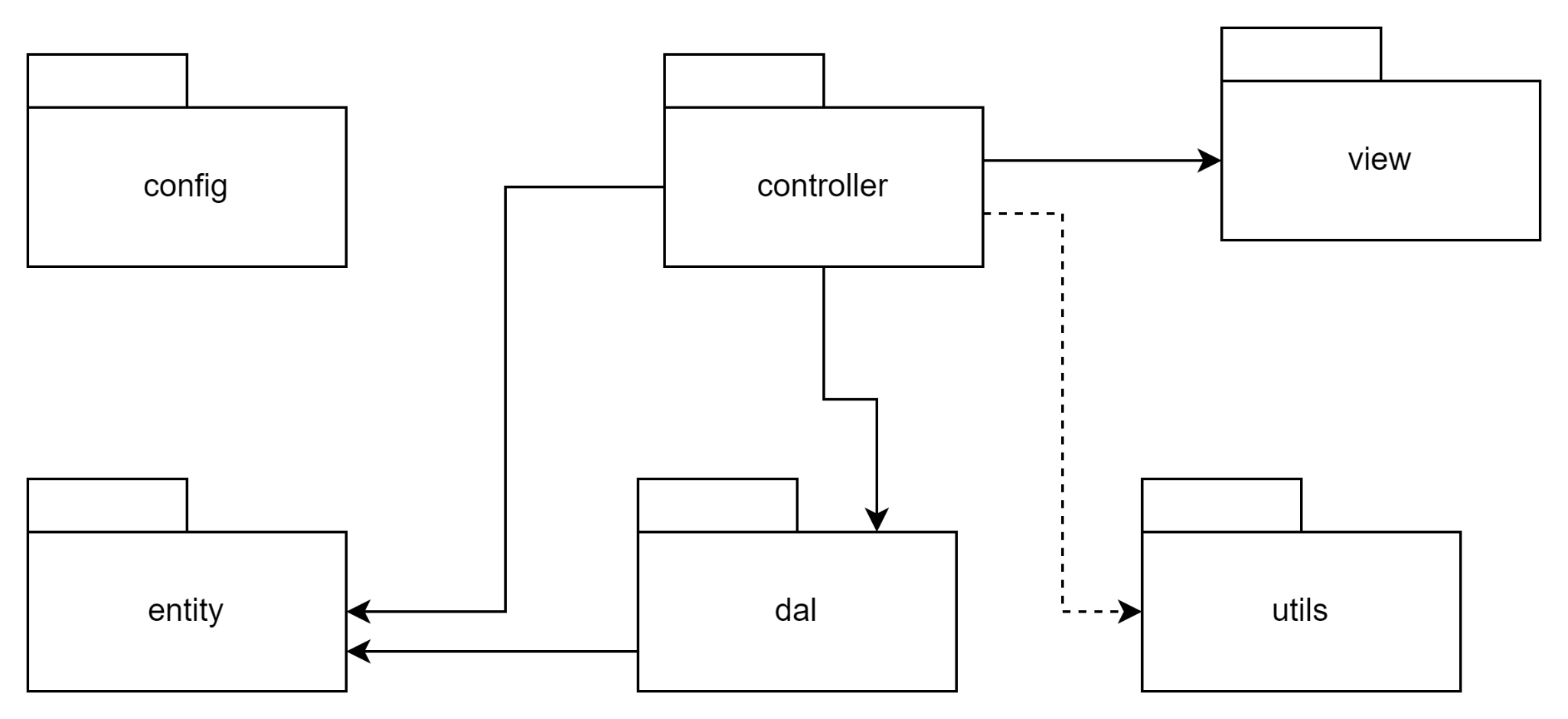
### 1.2 Table Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | Account | This table stores user account information such as email, password, role, and status.  - Primary keys: AccountID  - Foreign keys: None  - Unique keys: Email |
| 02 | Customer | This table holds customer details, including their personal information and account association.  - Primary keys: CustomerID  - Foreign keys: AccountID (references Account.AccountID)  - Unique keys: None |
| 03 | Instructor | This table holds instructor details, including their full name, specialization, and associated account.  - Primary keys: InstructorID  - Foreign keys: AccountID (references Account.AccountID)  - Unique keys: None |
| 04 | Role | This table defines the different roles that can be assigned to users, such as Admin, Instructor, or Customer.  - Primary keys: role\_id  - Foreign keys: None  - Unique keys: role\_name |
| 05 | Staff | This table stores information about staff members, including their account association and personal information such as birthday and in-job date.  - Primary keys: StaffID, AccountID  - Foreign keys: AccountID (references Account.AccountID)  - Unique keys: None |
|  |  |  |

## 2. Code Packages

*[Provide the package diagram for the system (or sub systems) and package description similar to sample diagram and using description table format below]*

### 2.1 Package Diagram



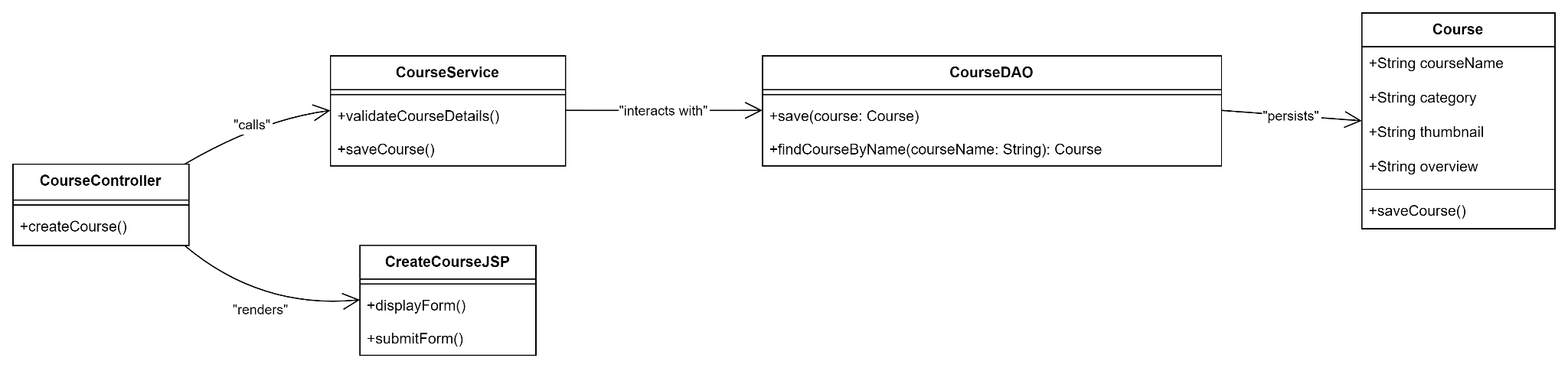
### 2.2 Package Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | config | This package is responsible for configuration-related components of the application. It contains settings or classes needed to initialize or configure services, database connections, or application parameters. |
| 02 | controller | This package is responsible for handling the business logic of the application. It processes requests from the view layer and interacts with the data access layer (DAL) or service classes to retrieve or modify data. |
| 03 | view | This package is responsible for managing the user interface and presentation layer. It handles displaying data to the user and capturing user input. |
| 04 | entity | This package contains the entity or model classes representing the core business objects. These classes typically map to database tables (like Account, Customer, etc.). |
| 05 | dal | The DAL (Data Access Layer) package is responsible for directly interacting with the database. It contains methods for performing CRUD (Create, Read, Update, Delete) operations on the data models. |
| 06 | utils | This package contains utility classes and helper functions. These are often reusable components that support tasks such as formatting, validation, or logging throughout the application. |

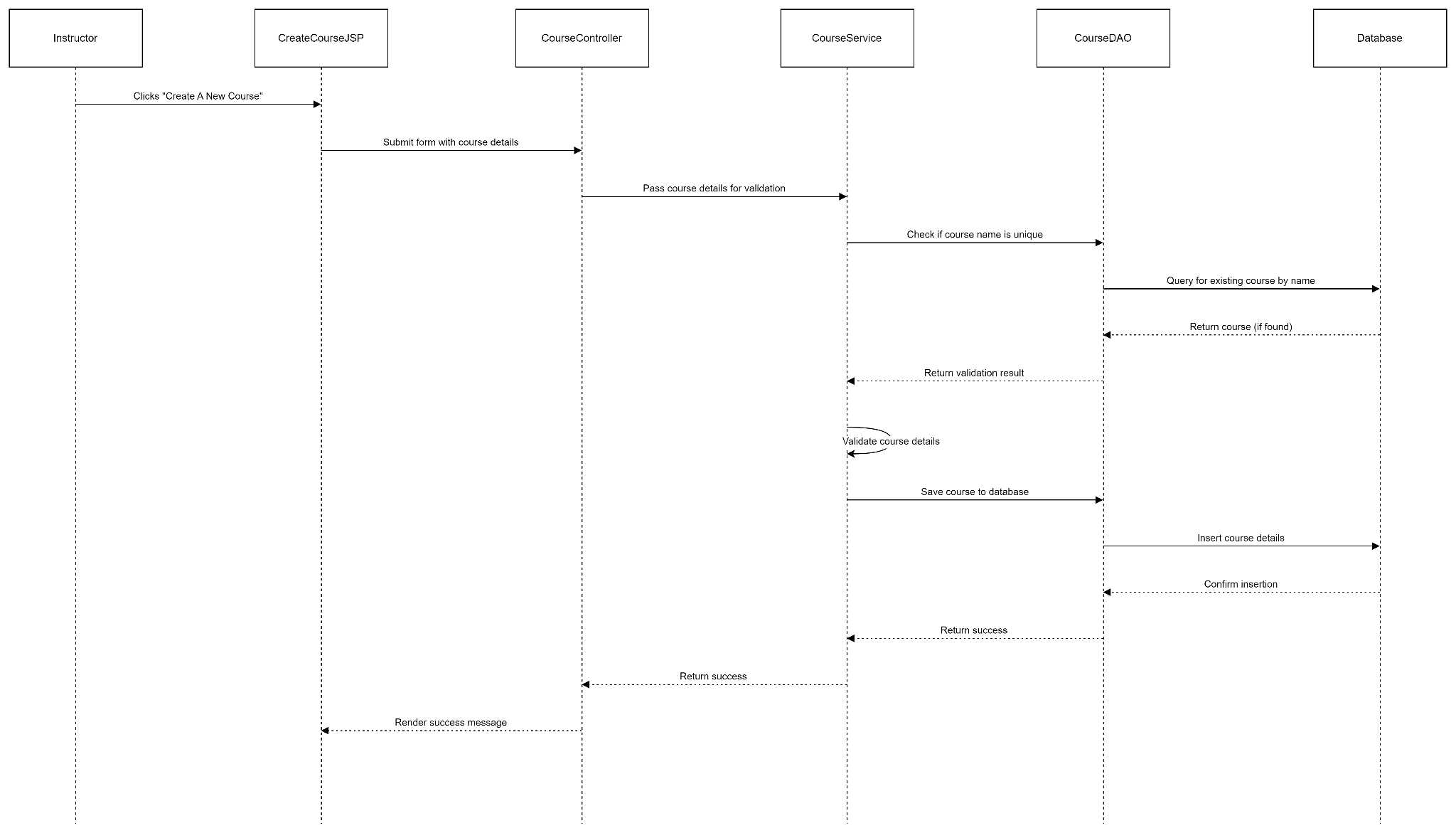
# IV. Details Design

## 1. Add course

#### 1.1. Class diagram



#### 1.2. Sequence diagram

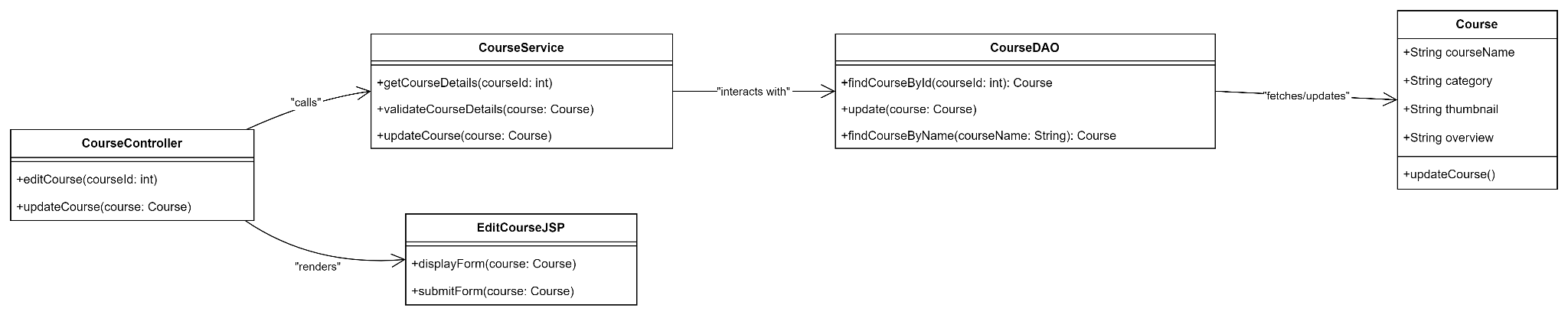


#### 1.3. Database queries

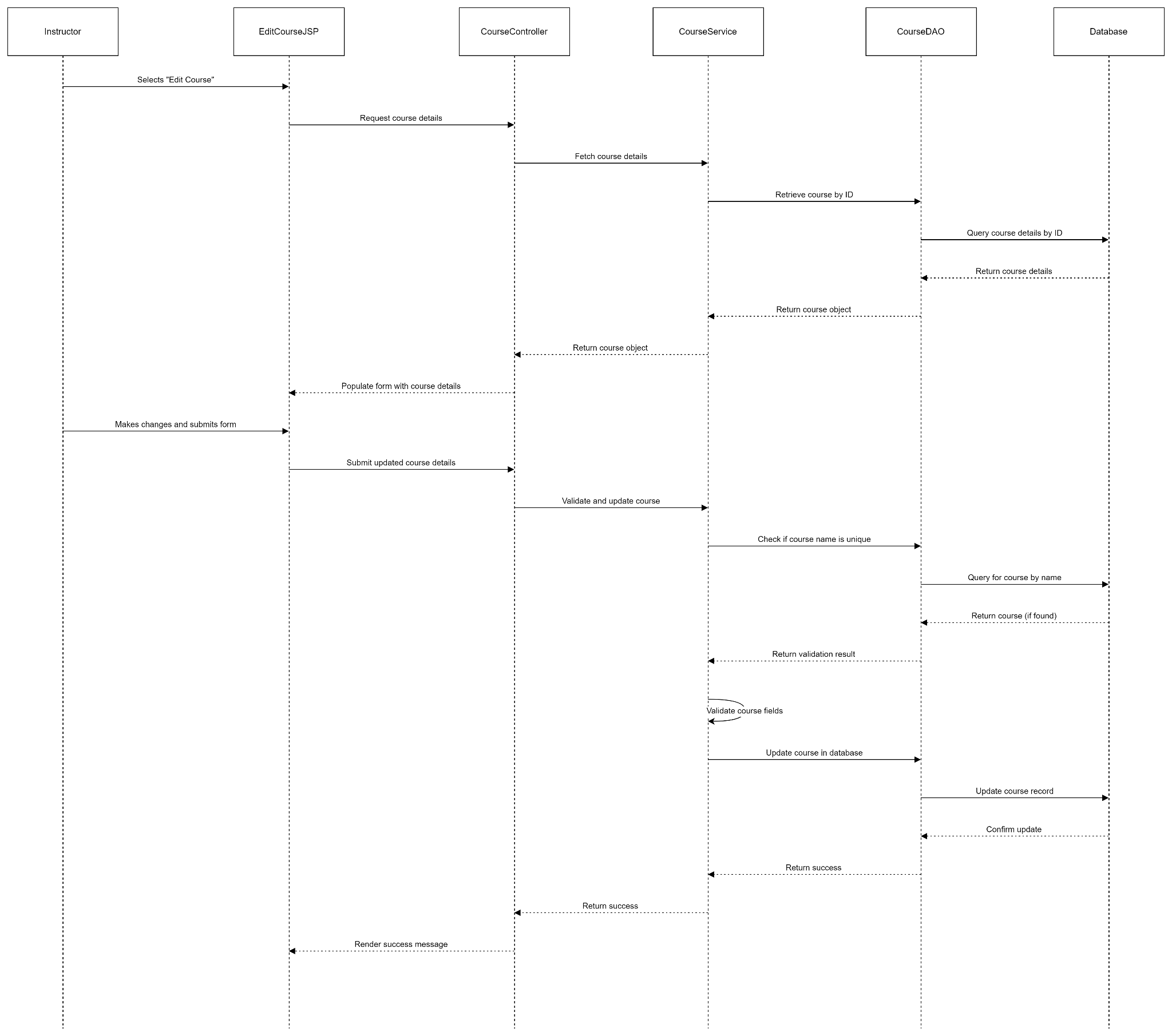
…..

## 2. Edit course

#### 1.1. Class diagram



#### 1.2. Sequence diagram

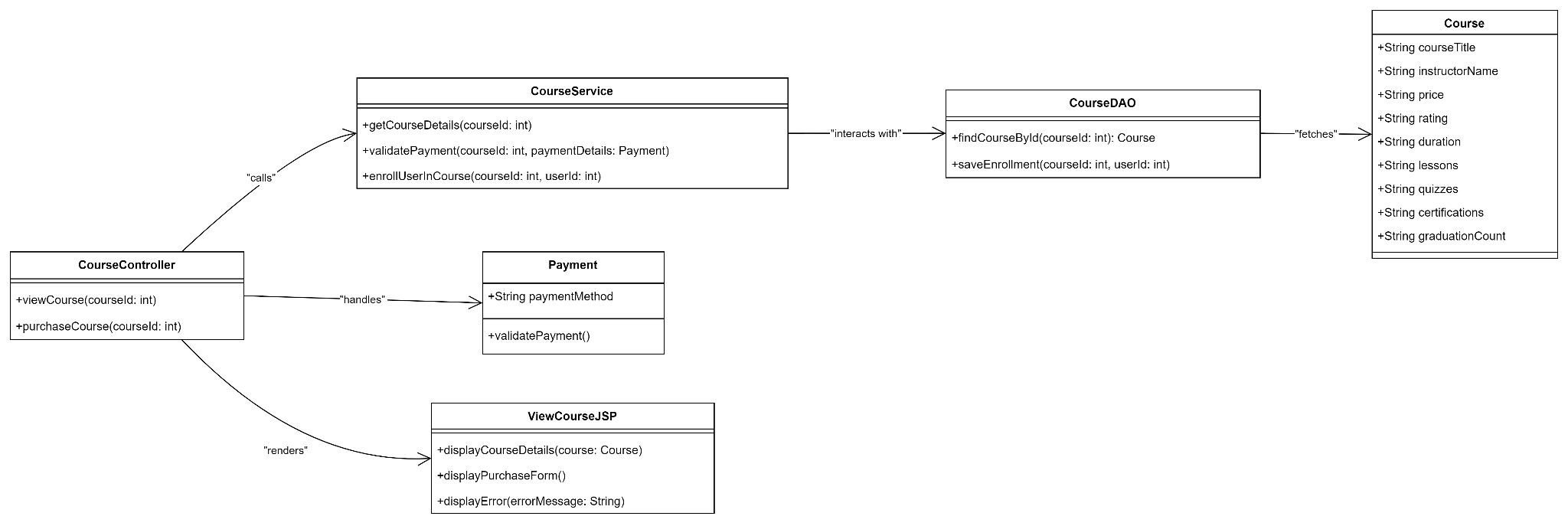


#### 1.3. Database queries

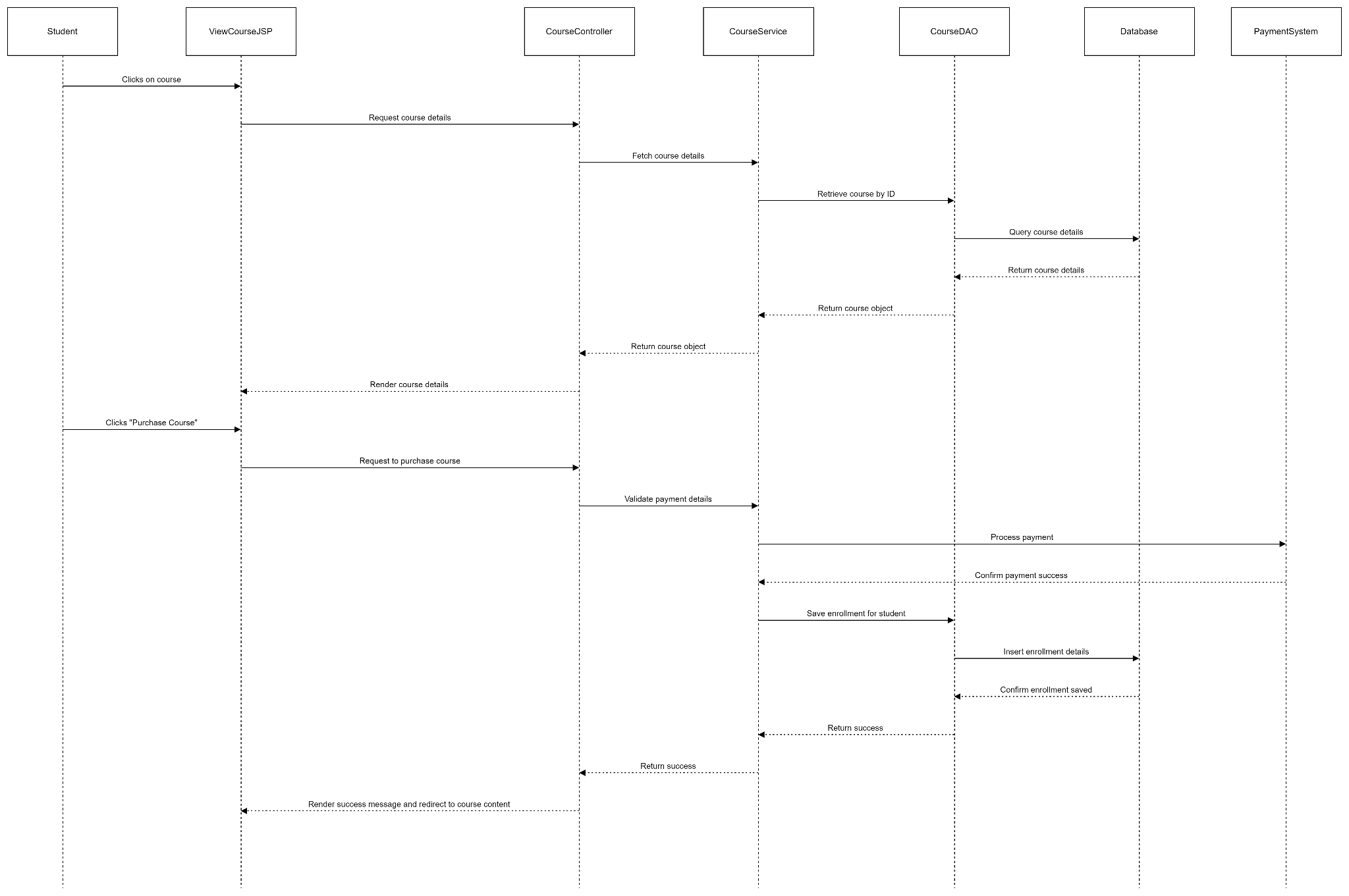
…..

## 3. View course details

#### 1.1. Class diagram



#### 1.2. Sequence diagram

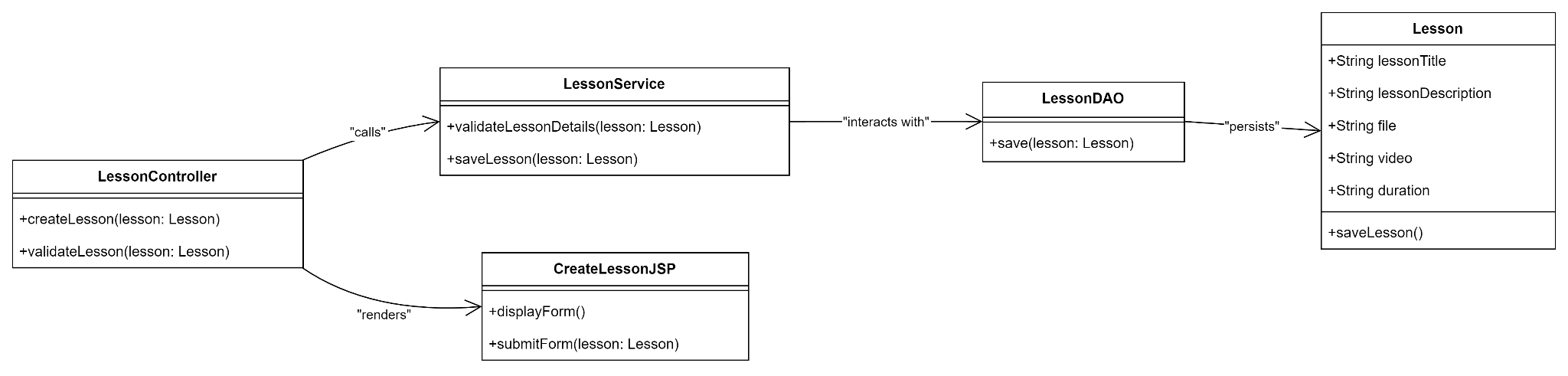


#### 1.3. Database queries

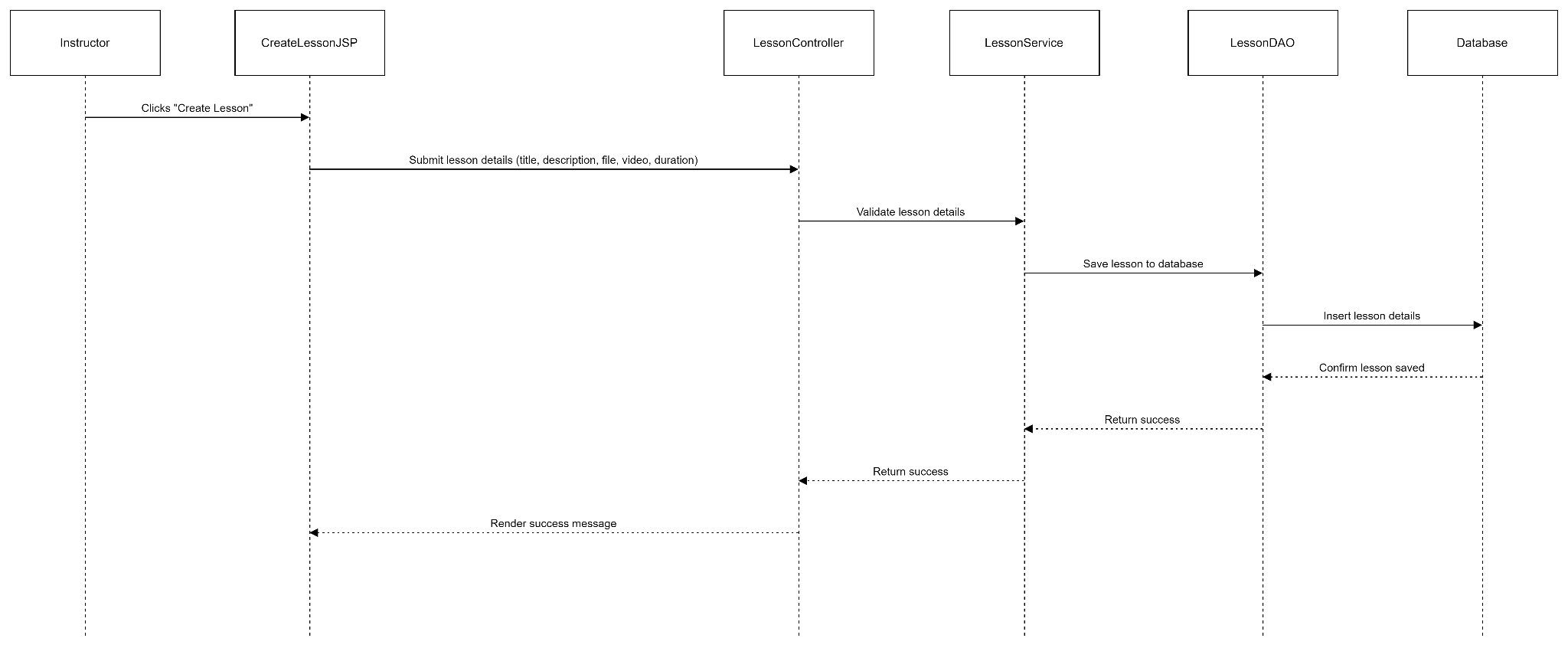
…..

## 4. Create lesson

#### 1.1. Class diagram



#### 1.2. Sequence diagram

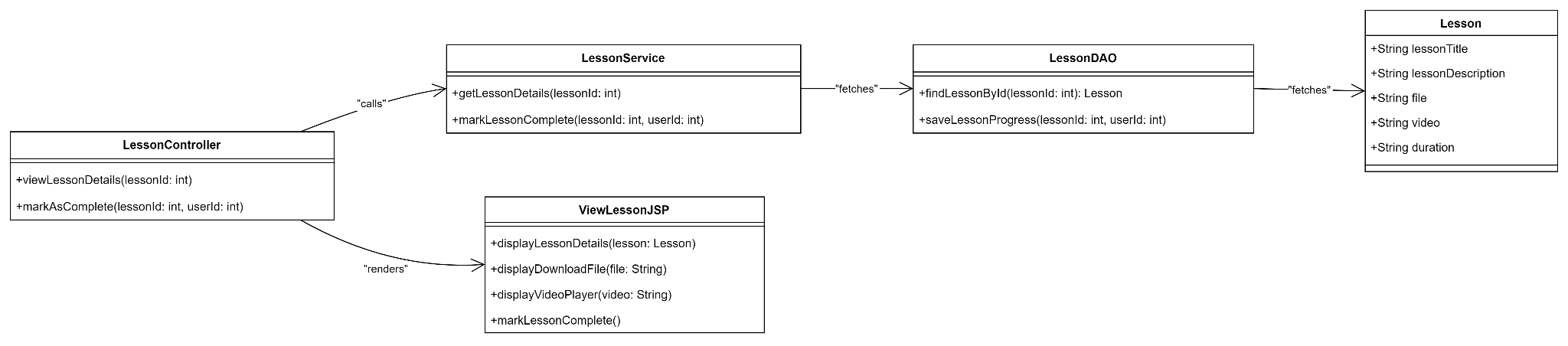


#### 1.3. Database queries

…..

## 5. View lesson details

#### 1.1. Class diagram



#### 1.2. Sequence diagram

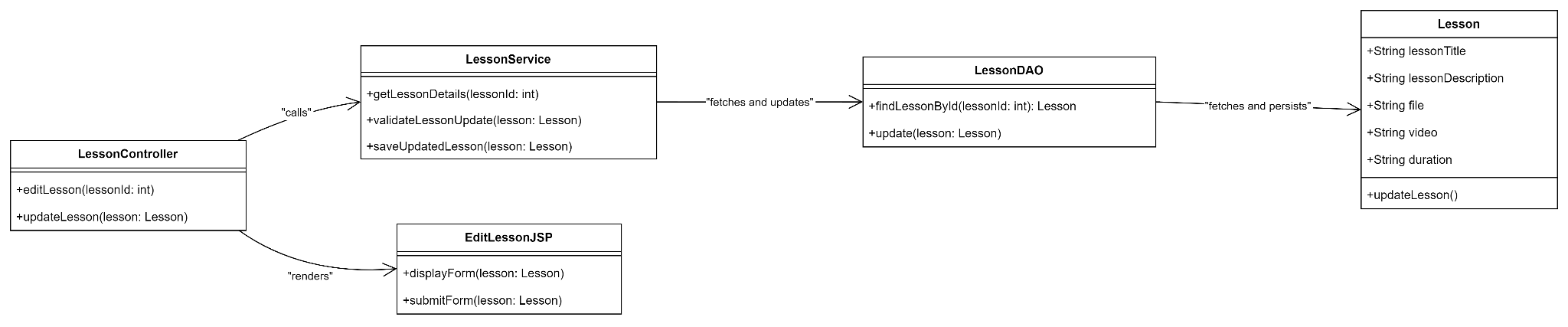


#### 1.3. Database queries

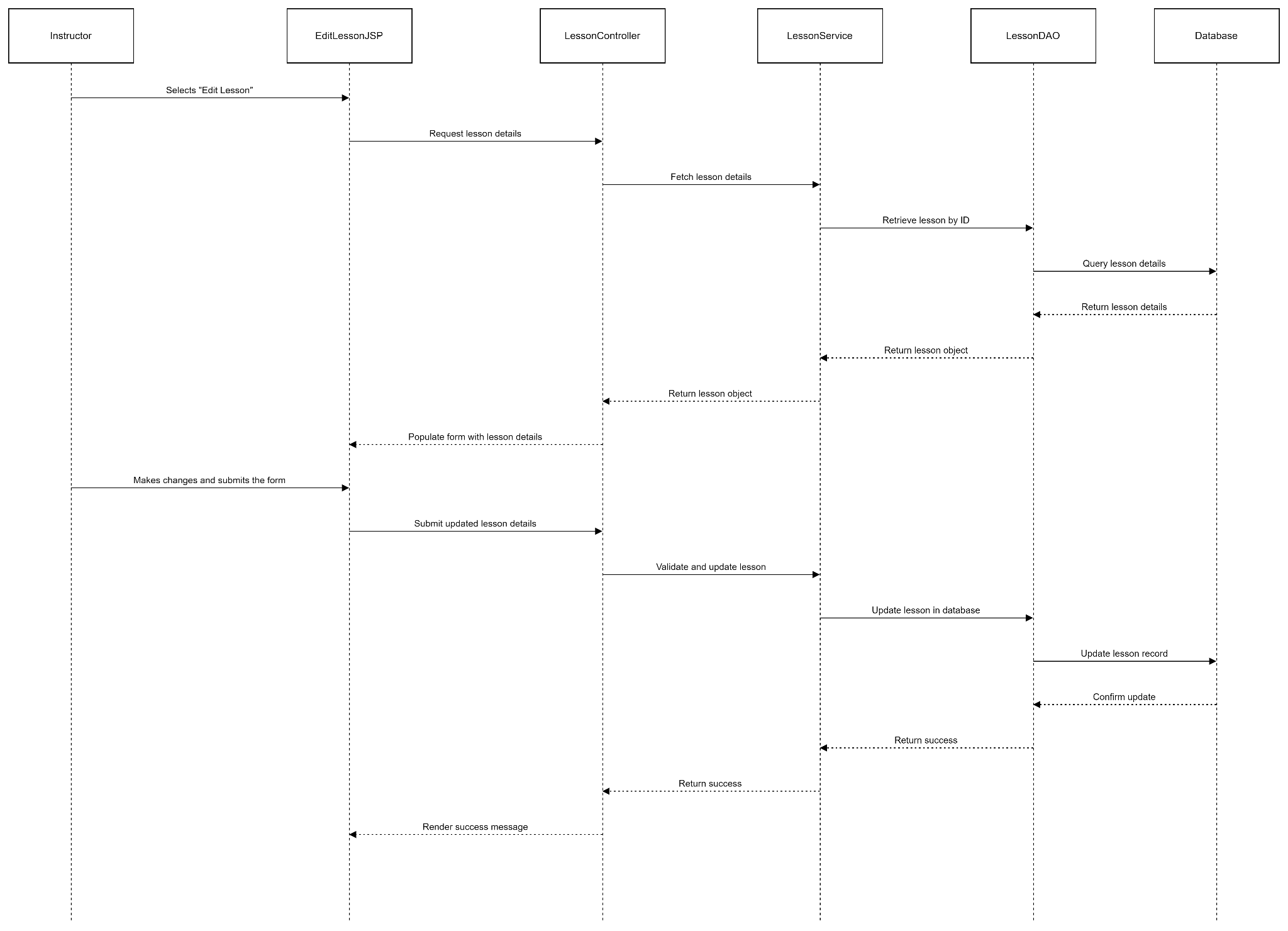
…..

## 6. Update lesson

#### 1.1. Class diagram



#### 1.2. Sequence diagram

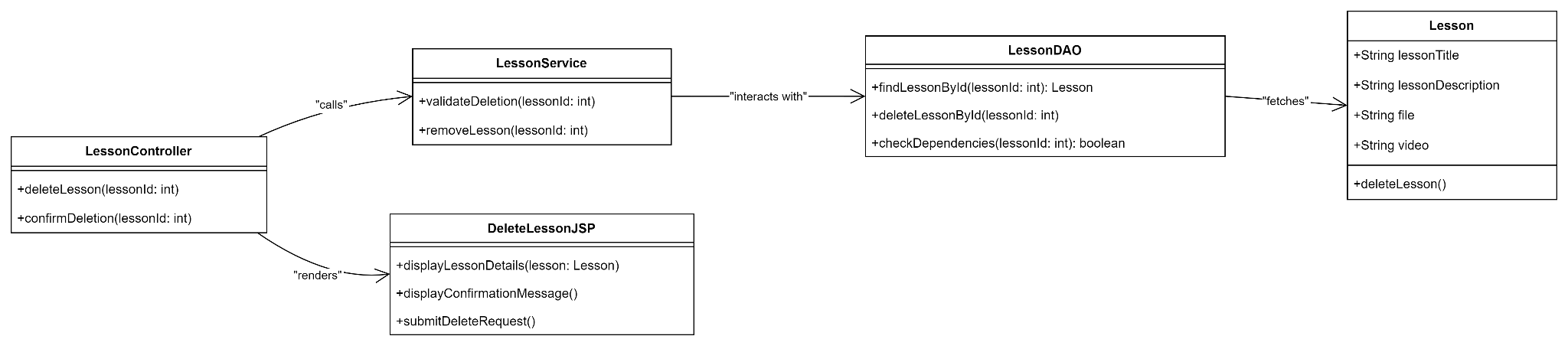


#### 1.3. Database queries

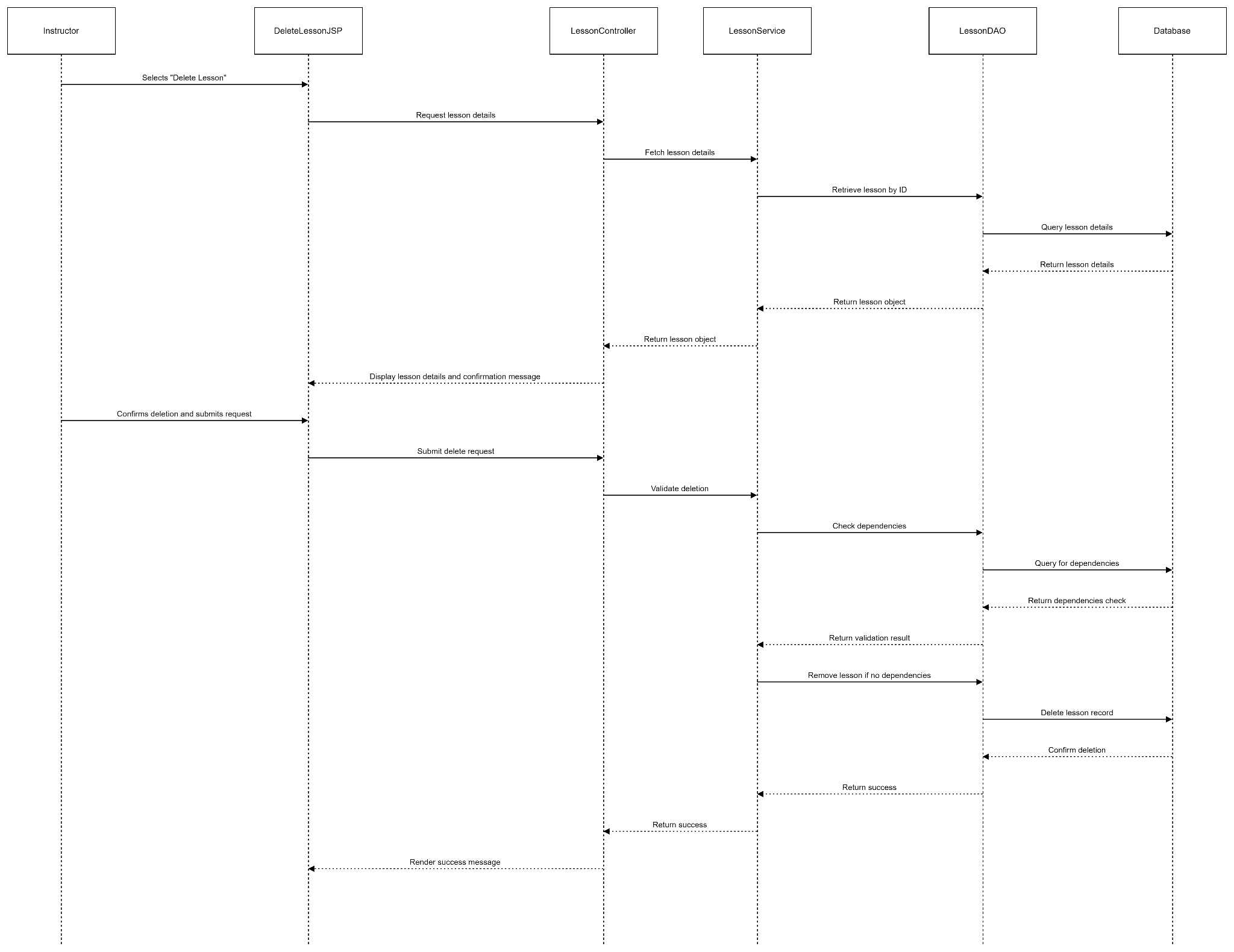
…..

## 7. Delete lesson

#### 1.1. Class diagram



#### 1.2. Sequence diagram



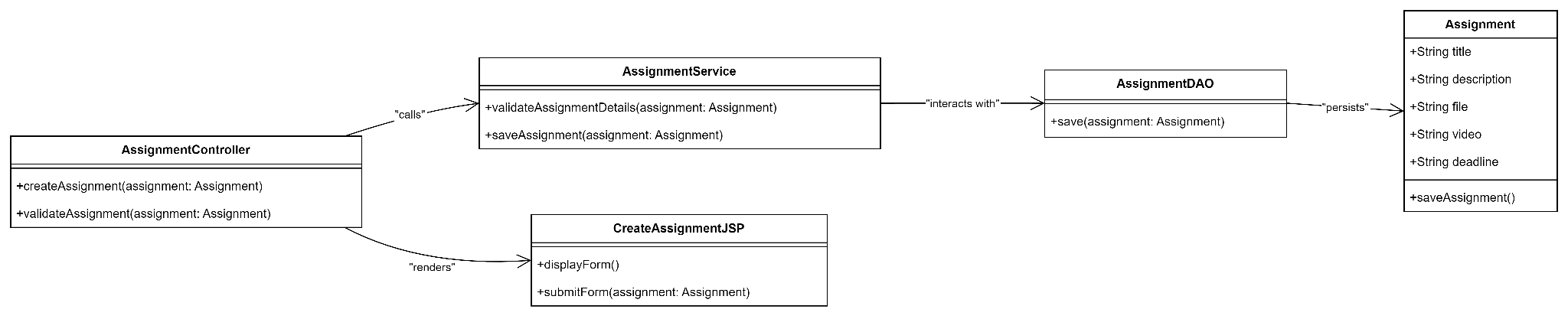
#### 1.3. Database queries

…..

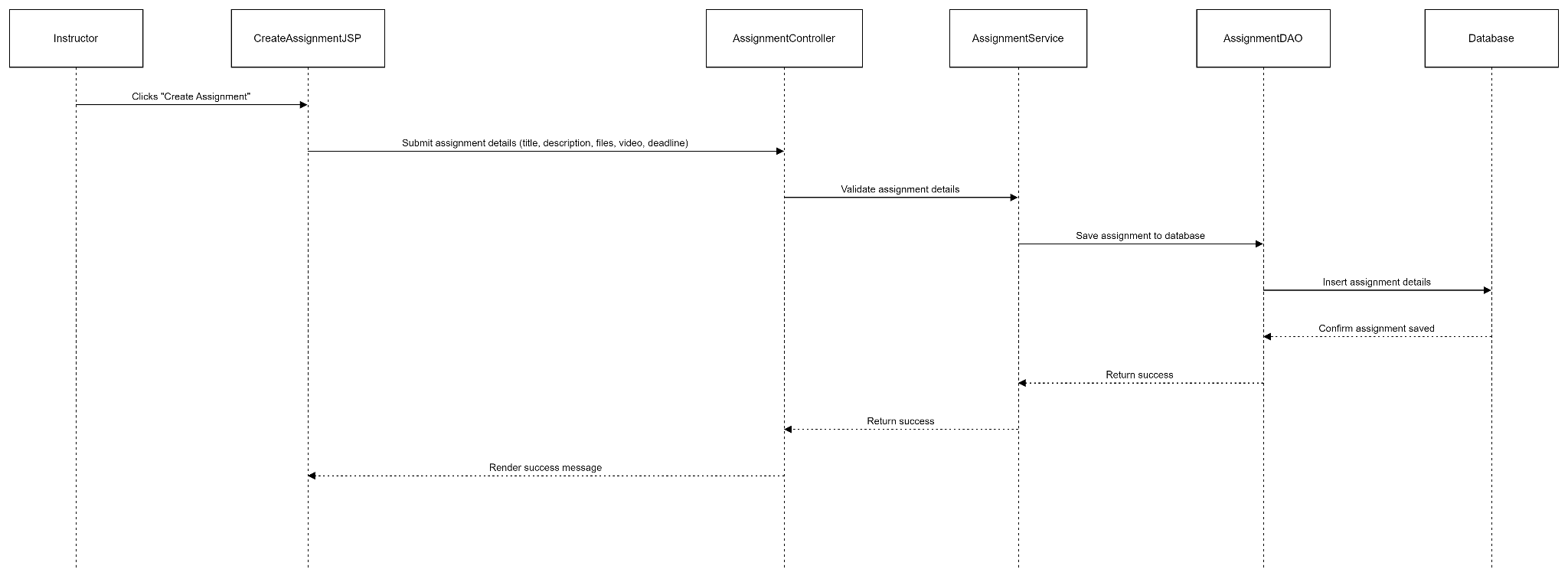
### 

## 8. Create assignment

#### 1.1. Class diagram



#### 1.2. Sequence diagram



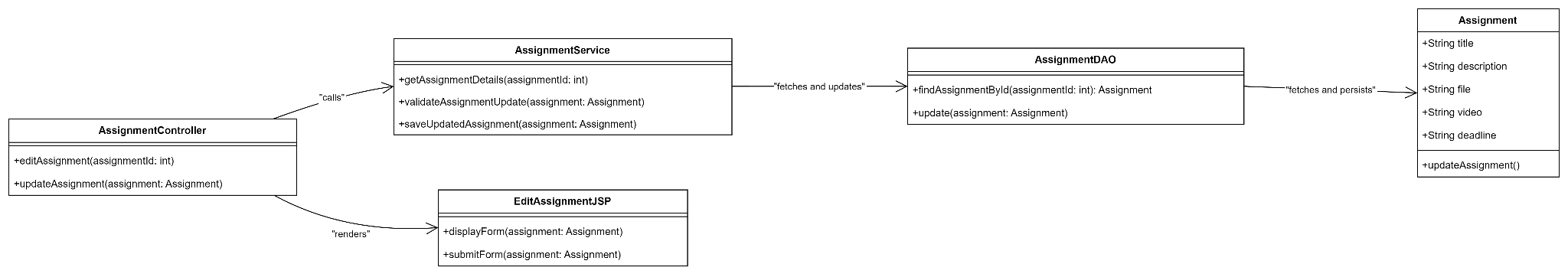
#### 1.3. Database queries

…..

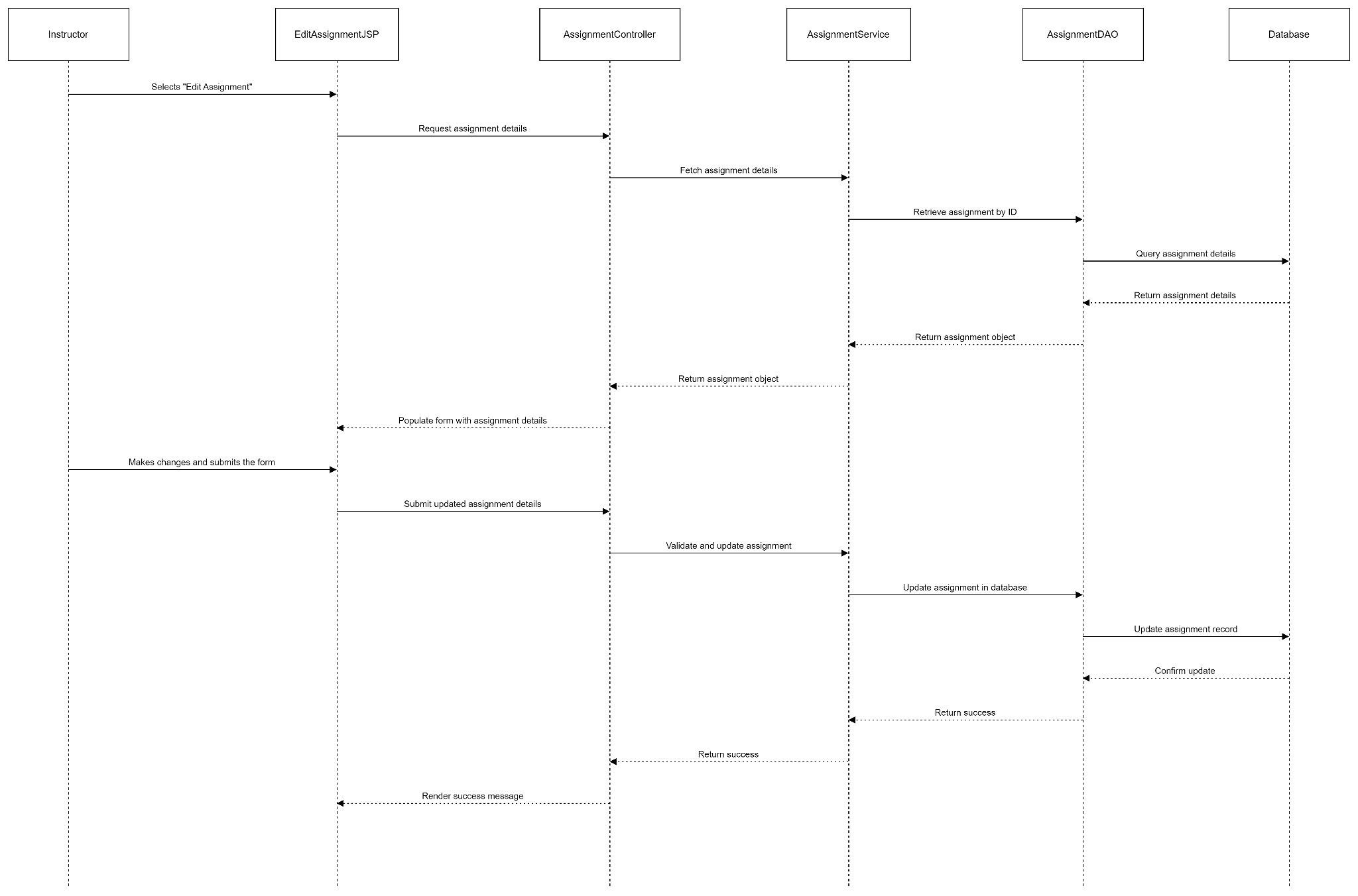
### 

## 9. Edit assignment

#### 1.1. Class diagram



#### 1.2. Sequence diagram



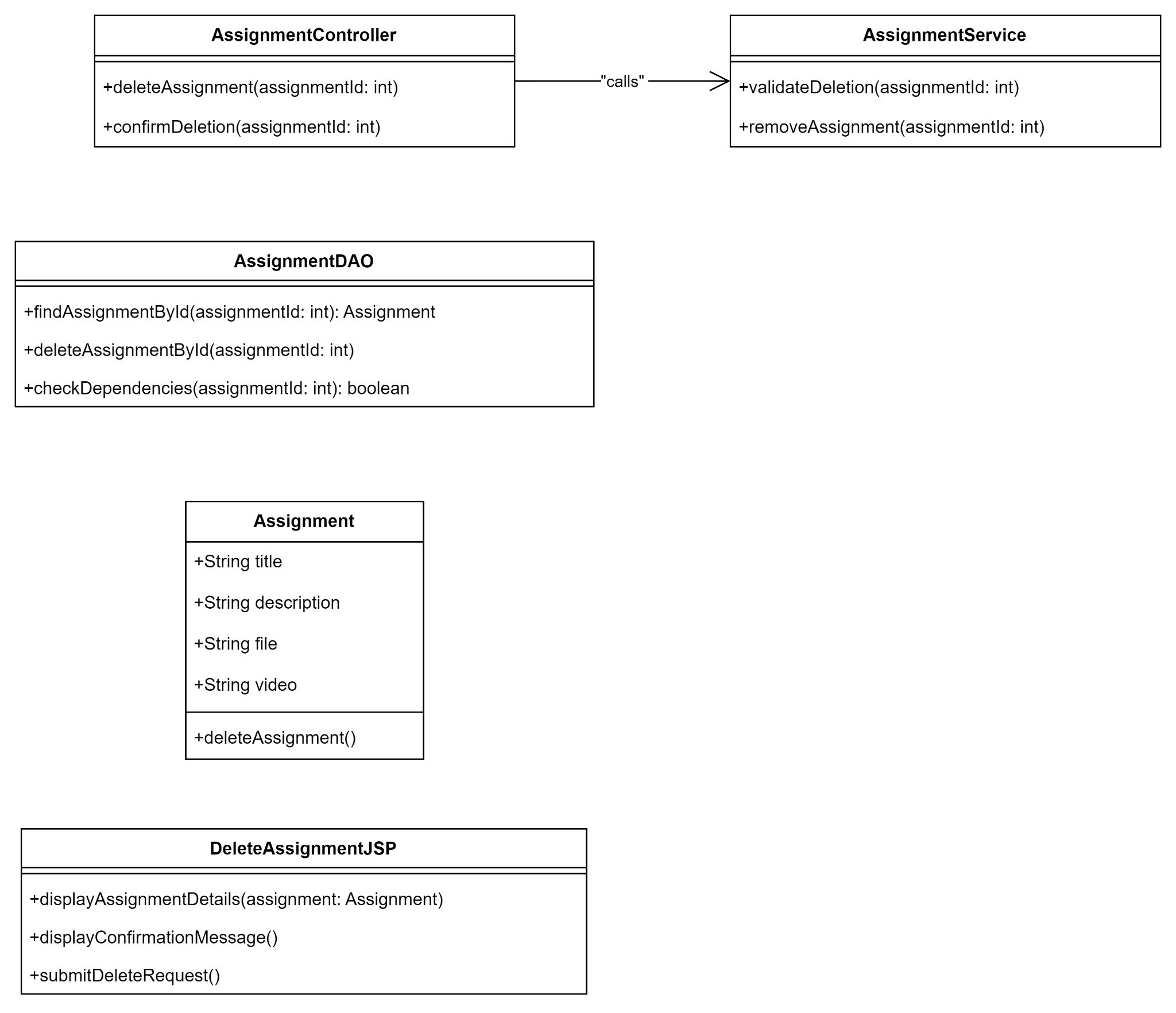
#### 1.3. Database queries

…..

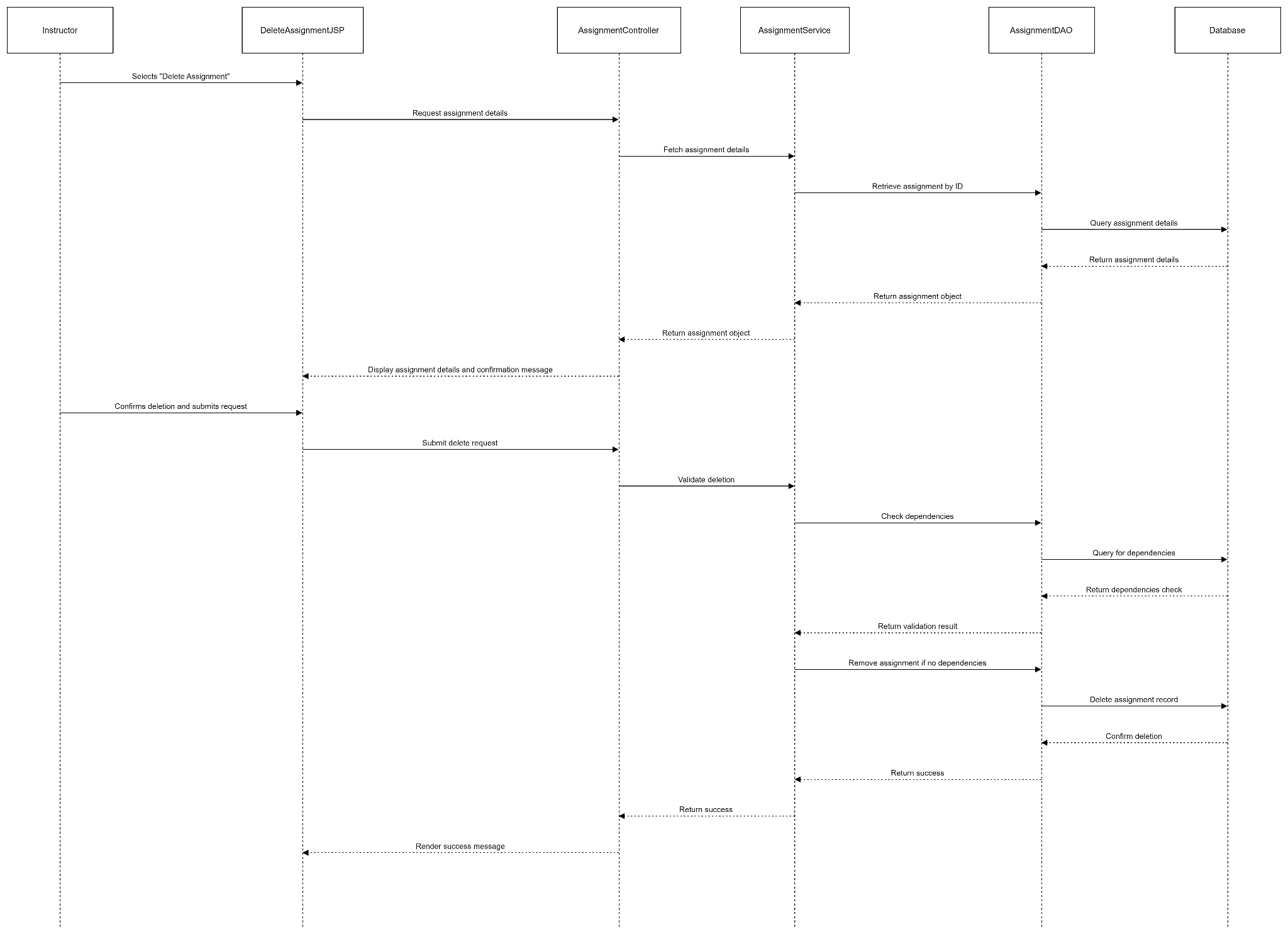
### 

## 10. Delete assignment

#### 1.1. Class diagram



#### 1.2. Sequence diagram



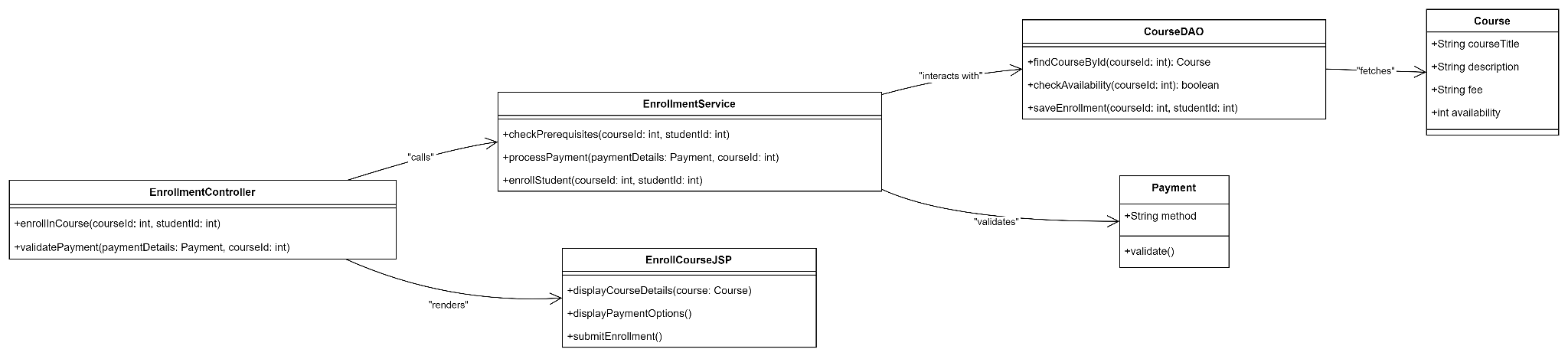
#### 1.3. Database queries

…..

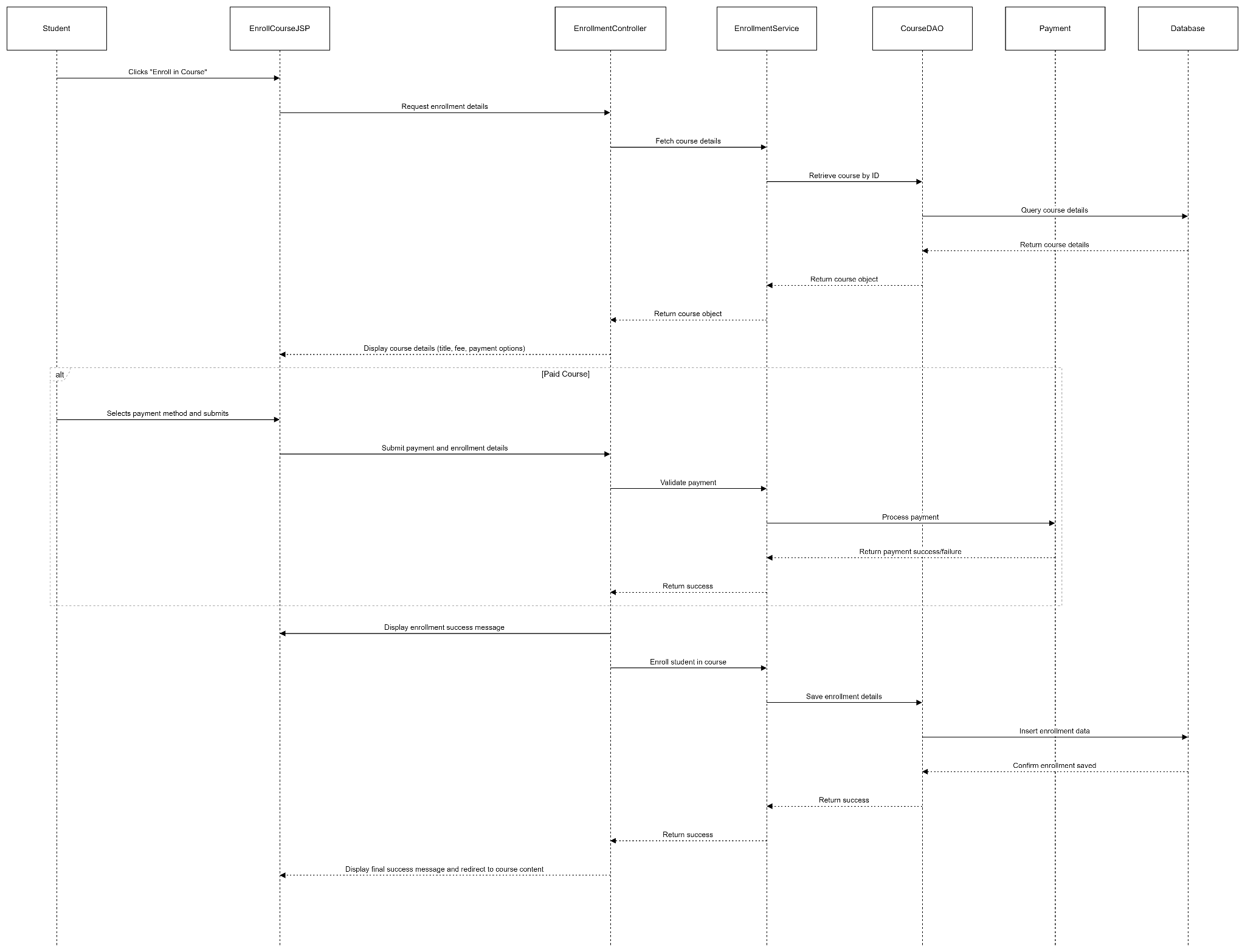
### 

## 11. Enroll course

#### 1.1. Class diagram



#### 1.2. Sequence diagram



#### 1.3. Database queries

…..

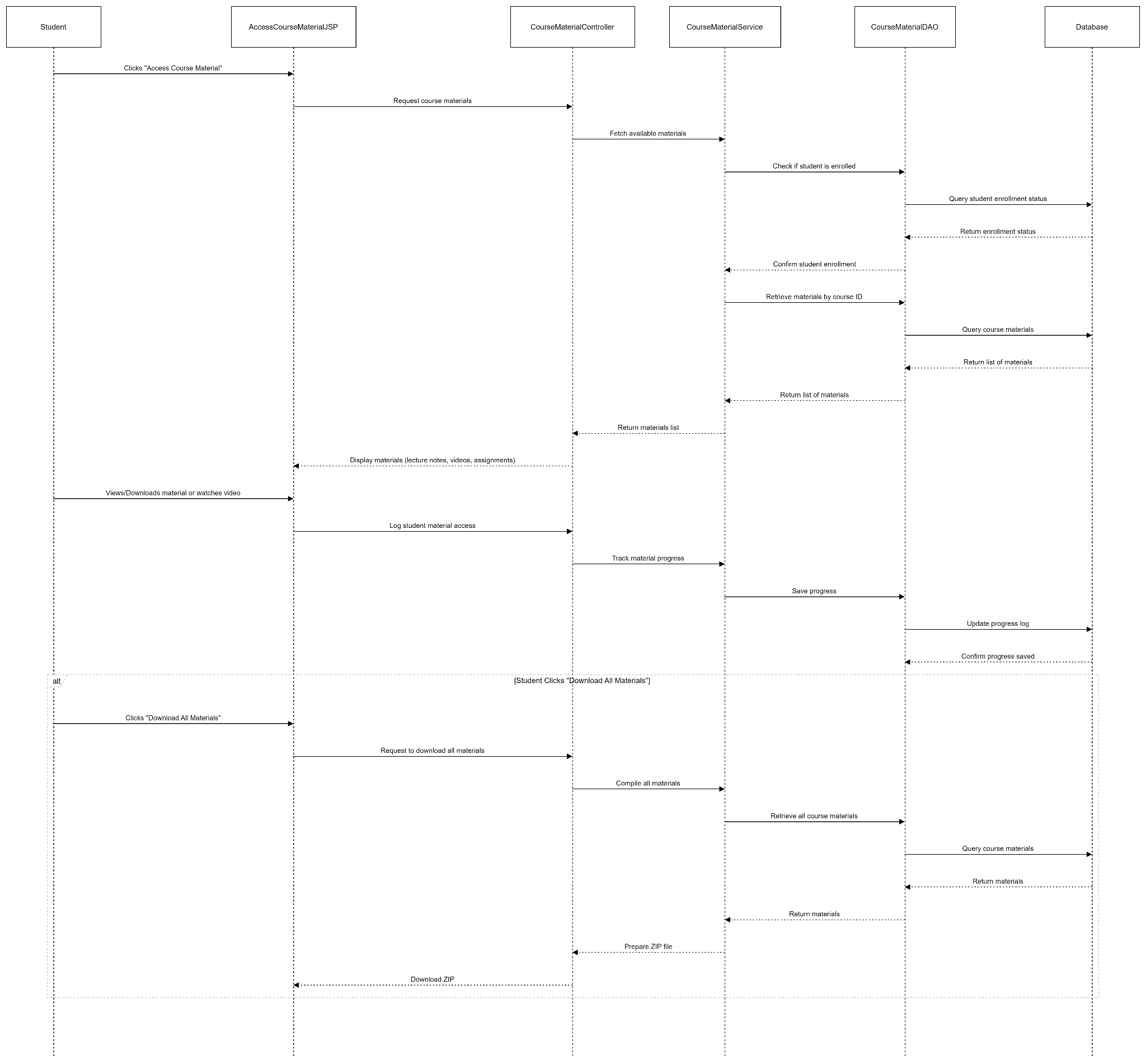
### 

## 12. Accessing course material

#### 1.1. Class diagram



#### 1.2. Sequence diagram



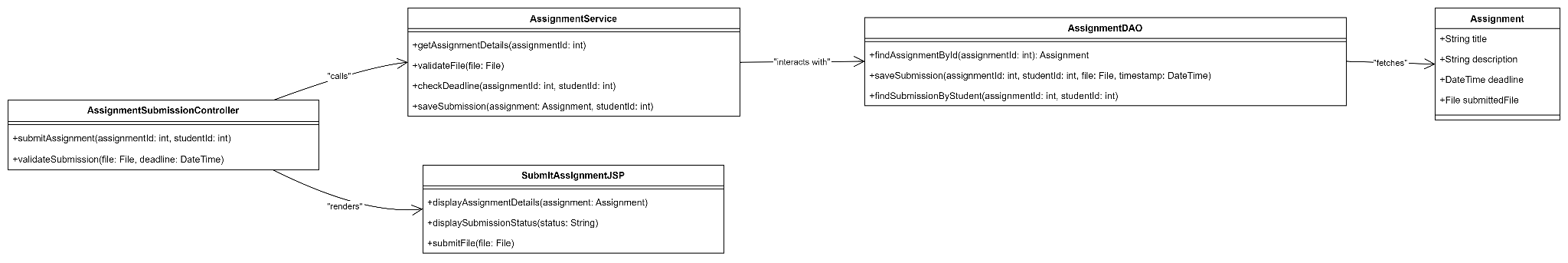
#### 1.3. Database queries

…..

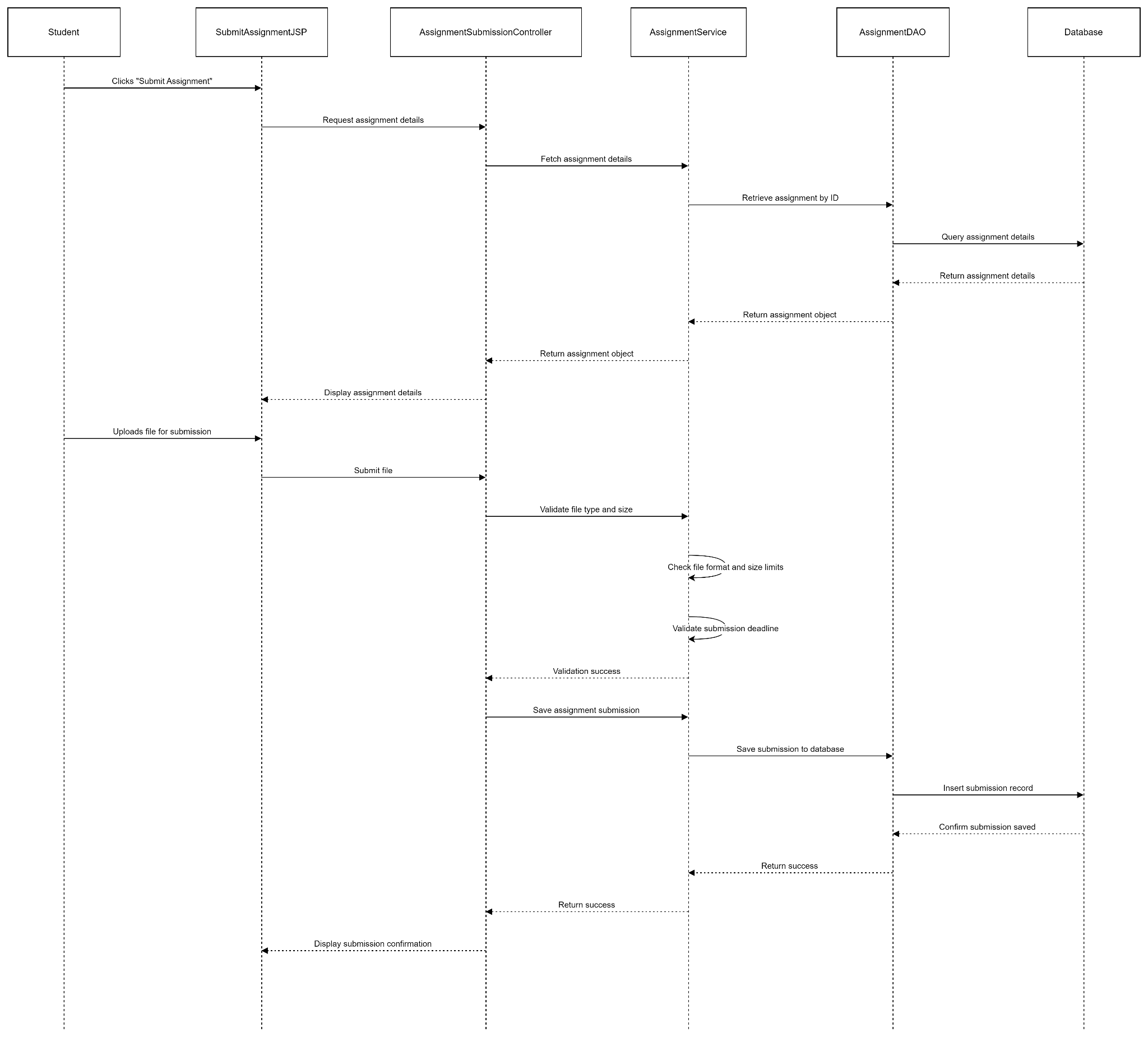
### 

## 13. Submit assignment

#### 1.1. Class diagram



#### 1.2. Sequence diagram

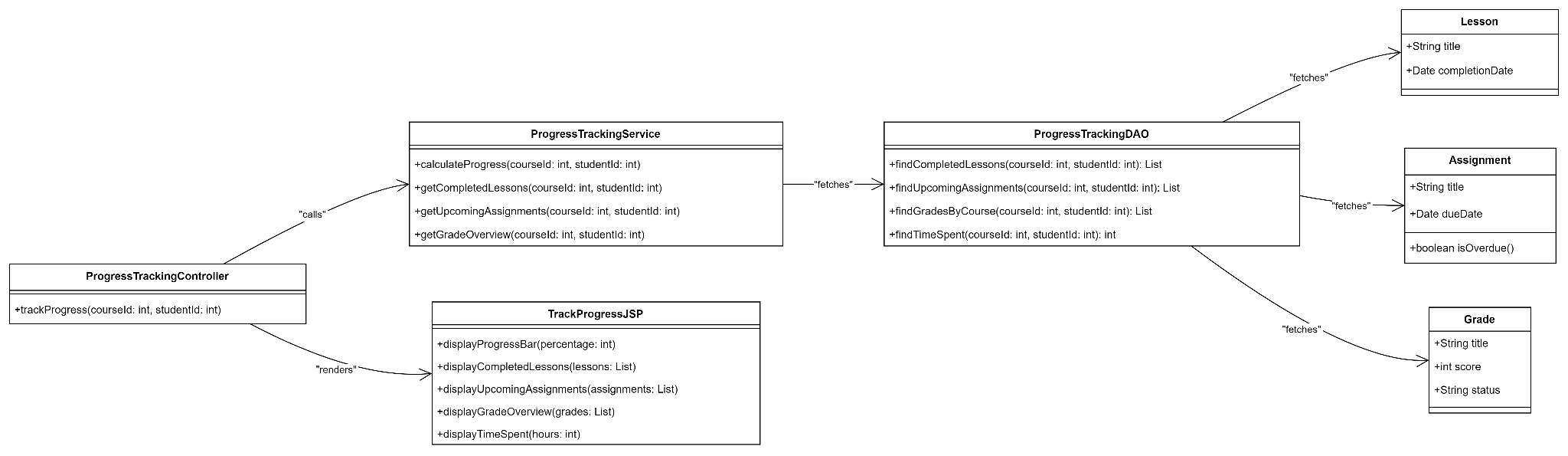


#### 1.3. Database queries

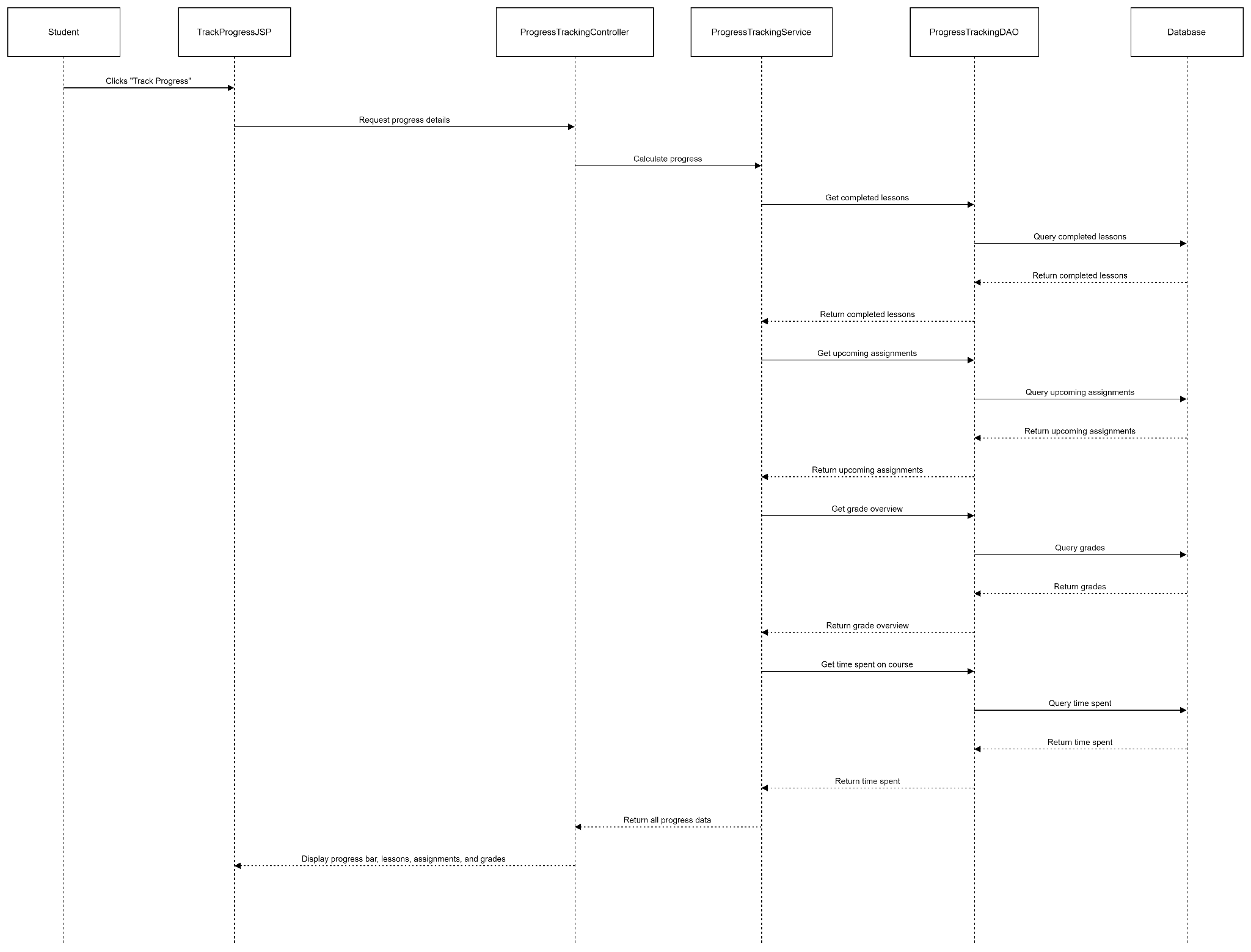
…..

## 13. Track progress

#### 1.1. Class diagram



#### 1.2. Sequence diagram



#### 1.3. Database queries

…..

### 

### 