

Online Lesson Management System

Use Case Model

1. Introduction

This document provides a detailed description of all the main functionalities of the Online Lesson Management System through use cases. Each use case is presented with a clear structure highlighting the involved actors, event flow, and possible exceptions.

2. Detailed Use Cases

UC1: A user registers as a student or a tutor

- **Main Actors:** Visitor
- **Description:** The visitor can register on the system by creating an account to access the lesson booking services
- **Pre-conditions:** The visitor must have internet access.
- **Post-conditions:** The user is registered and can log in.
- **Main Flow:**
 1. The visitor accesses the registration page.
 2. Enters their name, email, and password.
 3. The system verifies the data and creates the account.
 4. The user receives a confirmation email.
- **Extensions:**
 - 2a: If the email is already registered, an error message is displayed.
 - 3a: If the password does not meet security criteria, a modification is requested.

UC2: A student books a lesson

- **Main Actors:** Registered User
- **Description:** The user can book a lesson by selecting from the professional's available slots.
- **Pre-conditions:** The user must be logged in.
- **Post-conditions:** The lesson is booked and visible in the user's profile.

- **Main Flow:**
 1. The user accesses the lesson calendar.
 2. Selects an available date and time.
 3. Confirms the booking.
 4. The system sends a confirmation via notification/email.
- **Extensions:**
 - 2a: If no availability exists, an informational message is displayed.
 - UC3: Lesson Package Management

UC3: A tutor can manage lesson packages

- **Main Actors:** Freelancer
- **Description:** The professional can create and modify lesson packages with customized discounts.
- **Pre-conditions:** The professional must be logged in.
- **Post-conditions:** The package is visible to users. Main Flow:
 1. The professional accesses the "Package Management" section.
 2. Enters the package name, number of lessons, and discounted price.
 3. Saves the changes.
- **Extensions:**
 - 2a: If the price is inconsistent with the number of lessons, the system displays an error.

UC4: A student sends a lesson customisation request

- **Main Actors:** Registered User, Freelancer
- **Description:** The user can send a customized request to the professional to organize tailored lessons.
- **Pre-conditions:** The user must be logged in.
- **Post-conditions:** The professional receives the request and can respond.
- **Main Flow:**
 1. The user opens the chat with the professional.
 2. Sends a request with desired details.
 3. The professional reviews and responds.
- **Extensions:**
 - 2a: If the professional does not respond within 48 hours, the system sends a reminder.

UC6: A tutor or student cancels a lesson

- **Main Actors:** Registered User
- **Description:** The user can cancel a booked lesson following the refund policies.
- **Pre-conditions:** The lesson must be booked.
- **Post-conditions:** The lesson is canceled, and the refund is processed.
- **Main Flow:**
 1. The user accesses the booked lessons.
 2. Selects the lesson to cancel.
 3. Confirms the cancellation.
 4. The system processes the refund according to the policy.
- **Extensions:**
 - 3a: If the cancellation occurs too late, no refund is provided.

UC8: A student can review tutors after a lesson

- **Main Actors:** Registered User
- **Description:** A user can submit a review of the freelancer after completing a lesson.
- **Pre-conditions:** The user must have completed a lesson with the freelancer.
- **Main Flow:**
 1. The user accesses their completed lessons.
 2. The user selects the completed lesson with the freelancer they wish to review.
 3. The user enters a star rating (1 to 5) and optionally writes a detailed review.
 4. The user submits the review.
 5. The system saves the review, updates the freelancer's average rating, and notifies the freelancer about the new review.
- **Extensions:**
 - 3a: If the user attempts to submit a blank review, the system prompts them to provide at least a rating.
 - 4a: If the system detects inappropriate language or spam, it blocks the submission and informs the user.

3. Final Considerations

This alternative version of the use case model is structured to facilitate the reading and management of operational flows. The next steps include creating UML diagrams and integrating additional details into the use cases. Document under review. Further updates will be implemented progressively.