

Yasmine Mouatif 40249967

Hanine Tydrini 40226729

Use Case 1 & 2

Operation Contracts

Operations - Use Case 1:

`createLesson(type, scheduleId, locationId, privacy, spaceId, capacity, locationId)`

`ModifyLesson(lessonId)`

`viewAllLessons()`

`createOfferingForLesson(lessonId,instructorId)`

`viewOfferings()`

Contract: `createLesson`

Operation: `createLesson(type: String, scheduleId: int, locationId: int, privacy: String, spaceId: int, capacity: int)`

Cross reference(s): Use Case: Process Offerings

Preconditions:

1. The Administrator is authenticated by the system.
2. The following inputs are valid:
 - a. `type`: a valid lesson type string (e.g., "Yoga", "Math Class").
 - b. `scheduleId`: references an existing and valid schedule in the system.
 - c. `locationId`: references an existing and valid location in the system.
 - d. `spaceId`: references an existing and valid space within the location.
 - e. `capacity`: is a positive integer and within the permissible range for the `spaceId`.

Postconditions:

1. A new instance of Lesson has been created. [Instance creation]
2. The newly created Lesson has been associated with the specified `locationId`. [Association formation]
3. The Lesson has been associated with the specified `scheduleId`. [Association formation]
4. The Lesson has been associated with the specified `type`. [Attribute assignment]
5. The Lesson privacy settings (`privacy`) have been configured. [Attribute assignment]
6. The Lesson has been associated with the specified `spaceId`. [Association formation]
7. The capacity of the new lesson has been set as specified. [Attribute assignment]
8. The new lesson has been added to the Offerings List in the system for potential instructors to view. [Attribute modification]

Contract: modifyLesson

Operation: modifyLesson(lessonId: int)

Cross reference(s): Use Case: Process Offerings

Preconditions:

1. The Administrator is authenticated by the system.
2. The Lesson identified by LessonId exists in the system.

Postconditions:

1. The existing instance of Lesson identified by LessonId has been updated with the newDetail. [Attribute Modification]
2. The modified Offering remains associated with the same LocationId, ScheduleId, and LessonType unless explicitly changed by the Administrator. [no change]

Contract: viewAllLessons

Operation: [viewAllLessons\(\)](#)

Cross reference(s): Use Case: Process Offerings

Preconditions:

1. The Instructor is authenticated by the system.

Postconditions:

1. A list of all available lessons in the system is displayed to the user. This includes details such as:
 - a. Lesson type
 - b. Schedule (date and time)
 - c. Location
 - d. Space capacity and current availability [No modification]
2. No changes are made to the lesson data or system records. [No Modification]

Contract: createOfferingForLesson

Operation: [createOfferingForLesson\(lessonId: int, instructorId: int\)](#)

Cross reference(s): Use Case: Process Offerings

Preconditions:

1. The Instructor is authenticated by the system.
2. The Lesson identified by lessonId exists in the system.
3. The Lesson has not been taken by another Instructor.

Postconditions:

1. A new instance of Offering has been created, linking the specified lessonId and instructorId. [Instance Creation]
2. The newly created Offering is associated with the lesson and the instructor. [Association Formation]
3. The Offering has been added to the system's Offerings List, visible to public. [Attribute Modification]

Contract: viewOfferings

Operation: viewOfferings()

Cross reference(s): Use Case: Process Offerings

Preconditions:

2. The user (Administrator, Instructor, or client) is authenticated by the system.

Postconditions:

1. The system filters and displays a list of Offerings based on the user's access level.
[Attribute modification]
2. Each Offering in the list is associated with the relevant Instructor (if any), Location, TimeSlot, and LessonType. [Association formation]

Operations - Use Case 2:

viewOfferings()

createClientAccount(ClientDetails, password)

registerClient(Client GuardianDetails)

handleOfferingSelection(offeringId)

new Booking(offeringId, clientId)

Contract: viewOfferings

Operation: viewOfferings()

Cross reference(s): Use Case: Process Offerings

Preconditions:

3. The user (Administrator, Instructor, or client) is authenticated by the system.

Postconditions:

3. The system filters and displays a list of Offerings based on the user's access level.
[Attribute modification]
4. Each Offering in the list is associated with the relevant Instructor (if any), Location, TimeSlot, and LessonType. [Association formation]

Contract: createClientAccount

Operation: createClientAccount(clientDetails: Client, password: String)

Cross reference(s): Use Case: Register Client

Preconditions:

1. The clientDetails references a valid Client instance in the system.

Postconditions:

1. The Client instance provided in clientDetails is updated with the specified password. [Attribute Modification]
2. If the client is under 18 (isOver18 is false):
3. The system prompts the client to register a guardian. [System Prompt]
4. A new Client instance is created once guardian details are provided. [Instance Creation]
5. The Client and Guardian instances are associated with each other. [Association Formation]
6. A confirmation is sent to both the Client and Guardian (if applicable) upon successful registration. [Attribute modification]

Contract: handleOfferingSelection

Operation: handleOfferingSelection(offeringId: int)

Cross reference(s): Use Case: Select Offering for Booking

Preconditions:

1. The client is authenticated by the system.
2. The Offering identified by offeringID exists and is available.

Postconditions:

1. The selected Offering is associated with the client's account for booking. [Association formation]
2. The Offering remains available until the booking is confirmed by the client. [no change]

Contract: new Booking

Operation: new Booking(offeringId: int, clientId:int)

Cross reference(s): Use Case: Book an Offering

Preconditions:

1. The client is authenticated by the system.
2. The offeringId references an existing and available offering in the system.
3. The clientId references a valid client in the system.
4. The offering has available capacity for booking.

Postconditions:

1. A new instance of Booking has been created, linking the specified offeringId and clientId. [Instance Creation]
2. The Booking instance is associated with the client and the offering. [Association Formation]
3. The capacity of the offering is updated (decremented by 1). [Attribute Modification]
4. A confirmation notification is sent to the client about the successful booking. [Notification]