OPERATIONS CONTRACTS
PROCESS OFFERINGS

contract: createLesson
operations: createLesson(location, schedule, type, capacity)
cross references: use case: process offerings
precondition

- Location must exist in the location DB
- Lesson type must exist in the system
- Administrator must be the one to make the creation
- Time should be in the range of the location's schedule

postcondition:

- An instance of lesson should be created with specified attributes
- Association formed in between location and lesson

contract: viewAvailableLessons
operations: viewAvailableLessons()
cross references: use case: process offerings
precondition:

- Instructor is logged in correctly

postcondition:

- A list of all available offerings retrieved and displayed to the instructor

---

contract: acceptOffer()
operations: acceptOffer(Offer O)
cross references: use case: process offerings
precondition:

- Instructor is logged in correctly
- Offering in not thought by another instructor
- Instructor has the same specialization as the Offer

postcondition:

- Lesson is linked to the instructor
- A Offering is created based on the lessons details
- OfferID generated

Contract: createBooking

Operation: createBooking(lessonId, clientId)

Cross References: Use Case: Process Bookings

Precondition:

- The client must be registered and logged in.
- The lesson with lessonId exists and is available.
- The client does not have a conflicting booking for the same time slot.

Postcondition:

- A new Booking instance is created with the specified lessonId and clientId.
- The lesson's availability is updated to reflect the new booking.
- The association between the client and the booking is established.

Contract: cancelBooking

Operation: cancelBooking(bookingId, clientId)

Cross References: Use Case: Process Bookings
Precondition:

- The client is logged in.
- The booking with bookingId exists.
- The booking belongs to the client with clientId.

Postcondition:

- The booking status is updated to "canceled."
- If the offering mode is "group," the available spots for the lesson are increased by one.
- If the offering mode is "private," the lesson status is updated to "available."

Contract: viewBookings

Operation: viewBookings(clientId)

Cross References: Use Case: Process Bookings

Precondition:

- The client is logged in.

Postcondition:

- A list of all bookings associated with clientId is retrieved and displayed to the client.

Contract: increaseAvailableSpots

Operation: increaseAvailableSpots(lessonId)

Cross References: Use Case: Process Bookings
Precondition:

- The lesson with lessonId exists and has a "group" offering mode.

Postcondition:

- The available spots for the lesson are increased by one.

Contract: updateLessonStatus

Operation: updateLessonStatus(lessonId, status)

Cross References: Use Case: Process Offerings, Process Bookings
Precondition:

- The lesson with lessonId exists.
- The status provided is valid (e.g., "available," "canceled," "in-progress").

Postcondition:

- The lesson’s status is updated to the specified status.