

Yasmine Mouatif 40249967

Hanine Tydrini 40226729

Use Case 1 & 2

Operation Contracts

Operations - Use Case 1:

`createOfferings(location, timeslot, lessonType)`

`ModifyOffering(OfferingID, newDetail)`

`viewOfferings()`

`TakeOffering(OfferingID)`

Contract: `createOfferings`

Operation: `createOfferings(location: Location, timeSlot: TimeSlot, lessonType: LessonType)`

Cross reference(s): Use Case: Process Offerings

Preconditions:

1. The Administrator is authenticated by the system.
2. The Location, TimeSlot, and LessonType are valid objects in the system.

Postconditions:

1. A new instance of Offering has been created. [Instance creation]
2. The newly created Offering has been associated with the Location. [Association formation]
3. The newly created Offering has been associated with the TimeSlot. [Association formation]
4. The newly created Offering has been associated with the LessonType. [Association formation]
5. The new offering has been added to the Offerings List in the system for potential instructors to view. [Attribute Modification]

Contract: `modifyOffering`

Operation: `modifyOffering(OfferingID: String, newDetail: String)`

Cross reference(s): Use Case: Process Offerings

Preconditions:

1. The Administrator is authenticated by the system.
2. The Offering identified by OfferingID exists in the system.

Postconditions:

1. The existing instance of Offering identified by OfferingID has been updated with the newDetail. [Attribute Modification]

2. The modified Offering remains associated with the same Location, TimeSlot, and LessonType unless explicitly changed by the Administrator. [no change]

Contract: viewOfferings

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

Cross reference(s): Use Case: Process Offerings

Preconditions:

1. 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]

Contract: takeOffering

Operation: [takeOffering\(OfferingID: String\)](#)

Cross reference(s): Use Case: Process Offerings

Preconditions:

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

Postconditions:

1. The status of the Offering is updated to indicate it has been taken by the Instructor.
[Attribute modification]
2. The Instructor is now associated with the Offering. [Association formation]
3. The Offering is made publicly visible, displaying the associated instructor's details.
[Association formation]

Operations - Use Case 2:

viewClientOfferings()
registerClient(clientDetails)
selectOffering()
createBooking(offeringDetails)
setOfferingUnavailable(offering)

Contract: viewClientOfferings

Operation: viewClientOfferings()

Cross reference(s): Use Case: View Offerings Available to Clients

Preconditions:

1. The client is authenticated by the system.

Postconditions:

1. The system filters and displays a list of Offerings available for client bookings. [Attribute modification]
2. Each Offering in the list is associated with the relevant Instructor (if any), Location, and LessonID. [Association formation]
3. The availability status of each Offering is displayed. [Attribute modification]

Contract: registerClient

Operation: registerClient(clientDetails: ClientDetails)

Cross reference(s): Use Case: Register Client

Preconditions:

1. The provided clientDetails contain valid and complete information.

Postconditions:

1. A new instance of Client has been created with the provided details. [Instance creation]
2. If the client is under 18 (isOver18 is false):The system prompts the client to register a guardian. [Attribute modification]
3. A new Guardian instance is created once guardian details are provided. [Instance creation]
4. Both Client and Guardian instances are associated with each other. [Association formation]
5. A confirmation is sent to both the Client and Guardian (if applicable) upon successful registration. [Attribute modification]

Contract: selectOffering

Operation: selectOffering(offeringID: String)

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 temporarily 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: createBooking

Operation: [createBooking\(offeringDetails: OfferingDetails\)](#)

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

Preconditions:

1. The client is authenticated by the system.
2. The offeringDetails refer to an existing Offering that is available.

Postconditions:

1. A new instance of Booking has been created for the specified Offering. [Instance creation]
2. The Booking is associated with the Client and the selected Offering. [Association formation]
3. The Offering status is updated to reflect it has been booked by the client. [Attribute modification]
4. A booking confirmation is sent to the client. [Attribute modification]

Contract: setOfferingUnavailable

Operation: [setOfferingUnavailable\(offeringID: String\)](#)

Cross reference(s): Use Case: Update Offering Status

Preconditions:

1. The Offering identified by offeringID exists in the system.

Postconditions:

1. The status of the Offering is updated to indicate it is unavailable for booking. [Attribute modification]

