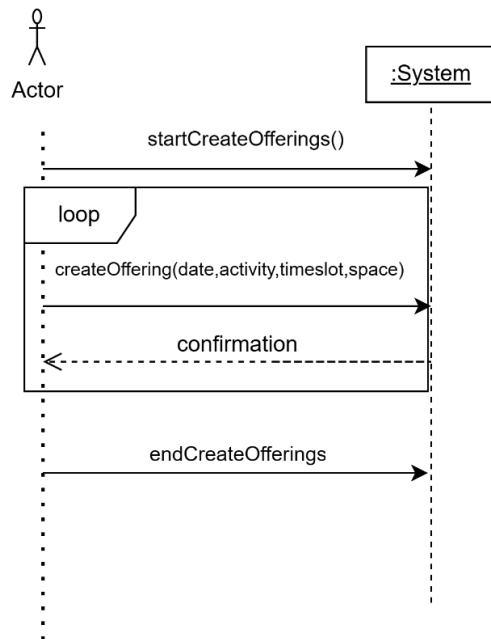
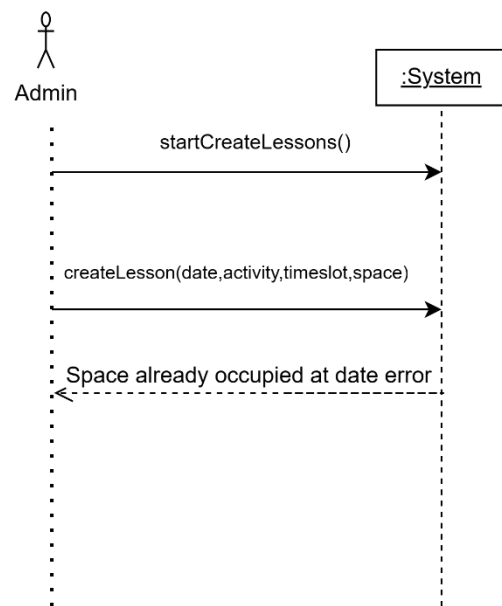


## Part I: Sequence diagram from Actor to system

### Success scenario



### Fail Scenario



### System operations:

`startCreateOfferings()`  
`createOffering(date,activity,timeslot,space)`  
`endCreateOfferings()`

### operation contacts:

1.

contract CO1: `startCreateOfferings`

Operation: `startCreateOfferings()`

Reference: sequence diagram part I

Preconditions:

- The administrator is logged into the system with the required permissions to create an offering.

Postconditions:

- A new session for creating an offering is started.
- The system is ready to accept offering details from the administrator.

2.

contract CO2: createOffering

Operation: createOffering(date,activity,timeslot,space)

Reference: sequence diagram part I

Preconditions:

- startCreateOffering has been initiated and the session is active.
- The lesson, type, time, and location parameters are valid and correspond to the system's offering requirements.

Postconditions:

- A new offering with the specified lesson, type, time, and location is created in the system.
- The system confirms the creation of this offering to the administrator.
- The offering data is stored in the system's offerings database or list.

3.

contract CO3: endCreateOfferings

Operation: endCreateOfferings()

Reference: sequence diagram part I

Preconditions:

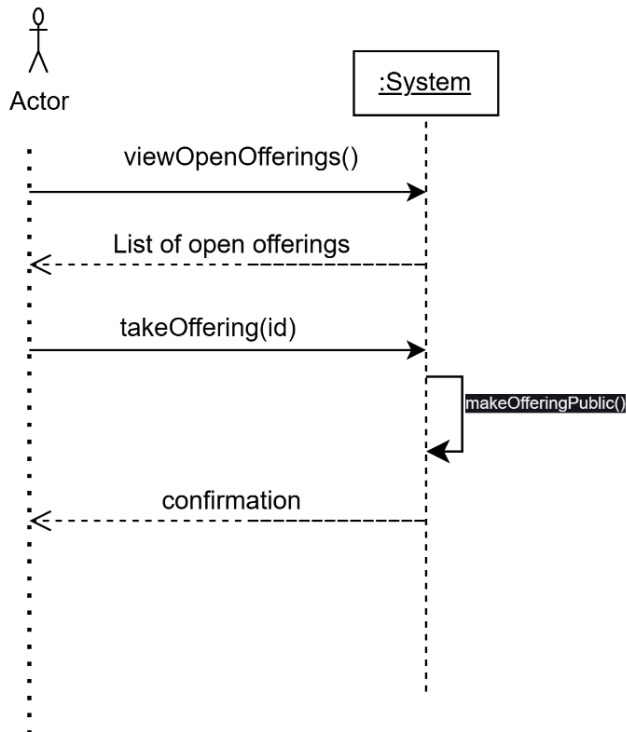
- At least one offering has been created using createOffering.
- The administrator has completed the session of creating offerings.

Postconditions:

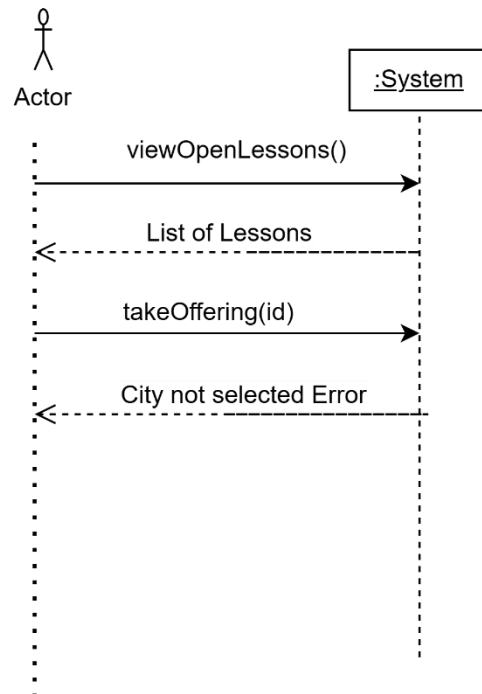
- The session for creating offerings is closed.
- The system no longer accepts new offerings in this session.
- The system finalizes and saves all created offerings from the session.

## Part II: Sequence diagram from Actor to system.

### Success scenario



### Fail Scenario



#### System operations:

`viewOpenOfferings()`  
`takeOffering(id)`  
`makeOfferingPublic()`

#### operation contacts:

1.

contract CO1: `viewOpenOfferings`

Operation: `viewOpenOfferings()`

Reference: sequence diagram part II

Preconditions:

- The instructor is logged into the system with permissions to view offerings.

Postconditions:

- The system retrieves and returns a list of open offerings available for the instructor.
- The list of open offerings is displayed to the instructor.

2.

contract CO2: takeOffering

Operation: takeOffering(id)

Reference: sequence diagram part II

Preconditions:

- viewOpenOffering has been called, and the instructor has identified an offering they wish to take.
- The id parameter corresponds to a valid and open offering in the system.

Postconditions:

- The system updates the offering identified by id to assign it to the instructor.
- The system triggers the internal operation makeOfferingPublic() to make this offering available or visible to others as necessary.
- The system sends confirmation to the instructor that the offering has been successfully taken.

3.

contract CO3: makeOfferingPublic

Operation: makeOfferingPublic()

Reference: sequence diagram part II

Preconditions:

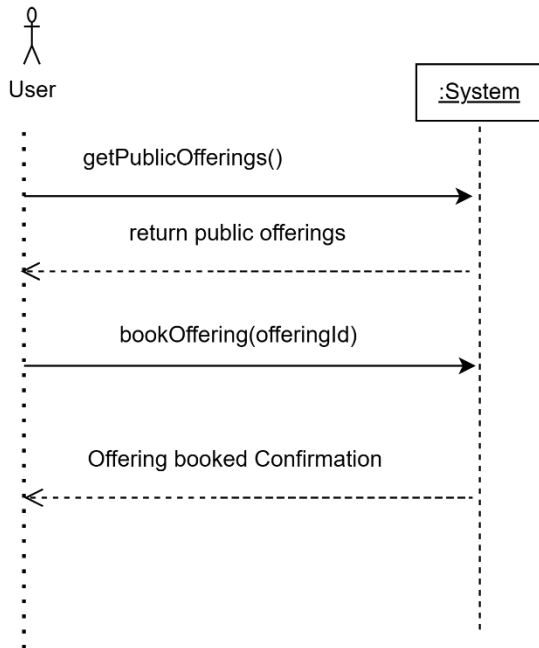
- takeOffering(id) has been successfully called.
- The offering with the specified id exists in the system.

Postconditions:

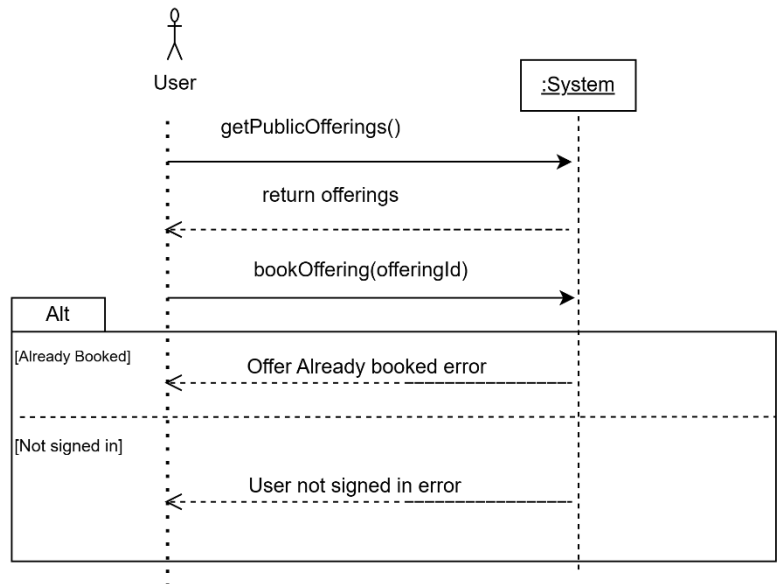
- The system updates the offering's status to be public or available to the designated audience.
- The updated offering status is recorded in the system, confirming the action.

## Part III: Sequence diagram from Actor to system

### Success scenario



### Fail scenario



### System operations:

`getPublicOfferings()`

`bookOffering(offeringsId)`

### operation contacts:

1.

contract CO2: `getPublicOfferings`

Operation: `getPublicOfferings ()`

Reference: sequence diagram part III

Preconditions:

- The client must be registered in the system (from a previous successful call to `registerBooking`).

Postconditions:

- The system retrieves and returns a list of all offerings where isPublic is set to true.

2.

contract CO3: bookOffering

Operation: bookOffering(offeringId)

Reference: sequence diagram part III

Preconditions:

- The client must be registered in the system.
- The specified offeringId must correspond to a public offering.
- The offering must have available capacity for new bookings.

Postconditions:

- The specified offering is updated to reflect the client's booking.
- A new booking record has been created linking the client and the offering.
- A confirmation of successful booking is returned to the client.