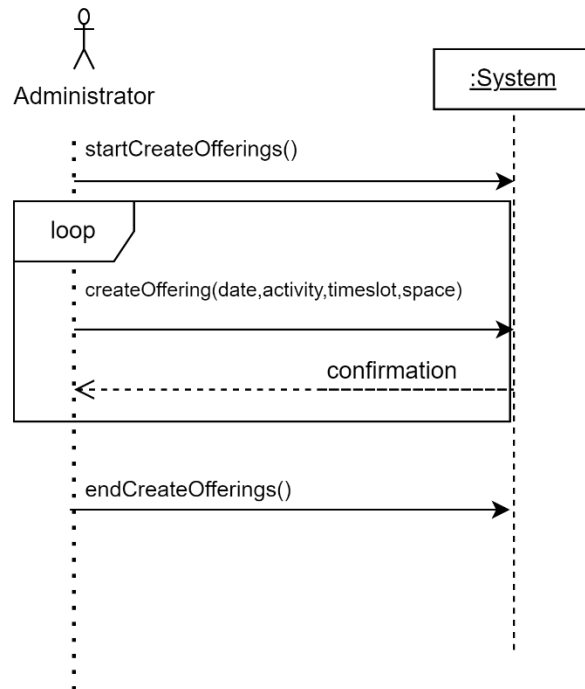


## Part I: Sequence diagram from instructor to system



### System operations:

1. `startCreateOfferings()`
2. `createOffering(date,activity,timeslot,space)`
3. `endCreateOfferings()`

### operation contacts:

1.

contract CO1: `startCreateOfferings`

Operation: `startCreateOfferings()`

Reference: sequence diagram part I

Preconditions:

-The admin should be logged in the system

Postconditions:

- The system is ready to accept new offering creation requests.

2.

contract CO2: createOffering

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

Reference: sequence diagram part I

Preconditions:

- The startCreateOffering() method has been called.
- All parameters must be valid and formatted correctly.

Postconditions:

- A new offering is created in the system.
- A confirmation is sent to the admin indicating successful creation.

3.

contract CO3: endCreateOfferings

Operation: endCreateOfferings()

Reference: sequence diagram part I

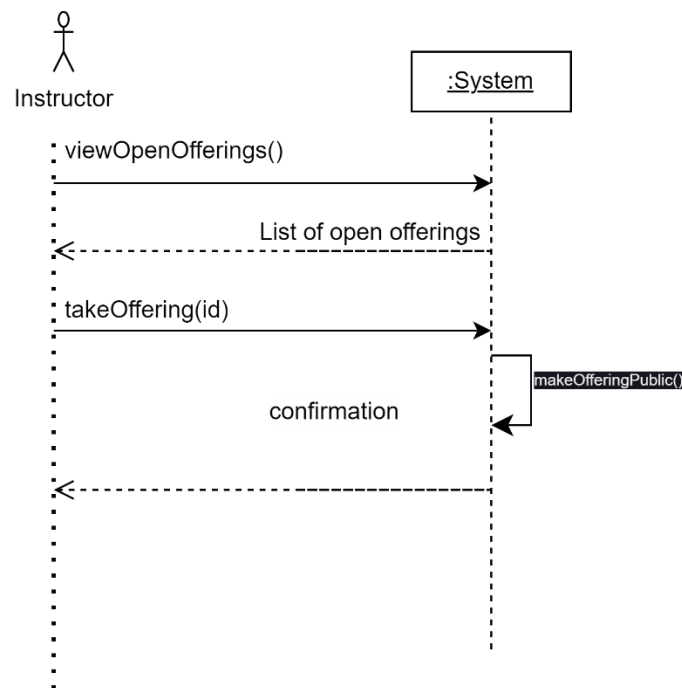
Preconditions:

- The admin must have called startCreateOffering() and created at least one offering.

Postconditions:

- The system will finalize the created offerings during the session
- The system also prepares for the next operation.

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



### System operations:

`viewOpenOfferings()`

`takeOffering(id)`

`makeOfferingPublic()`

### operation contacts:

1.

contract CO1: `viewOpenOfferings`

Operation: `viewOpenOfferings()`

Reference: sequence diagram part II

Preconditions:

- The instructor must be logged into the system.

Postconditions:

- A list of open offerings is returned to the instructor.

2.

contract CO2: takeOffering

Operation: takeOffering(id)

Reference: sequence diagram part II

Preconditions:

- The viewOpenOffering() method has been called, and the instructor has received a list of offerings.
- The offering ID must be valid and exist in the list of open offerings.

Postconditions:

- The system will attempt to make the offering public and notify the instructor of the outcome.

3.

contract CO3: makeOfferingPublic

Operation: makeOfferingPublic()

Reference: sequence diagram part II

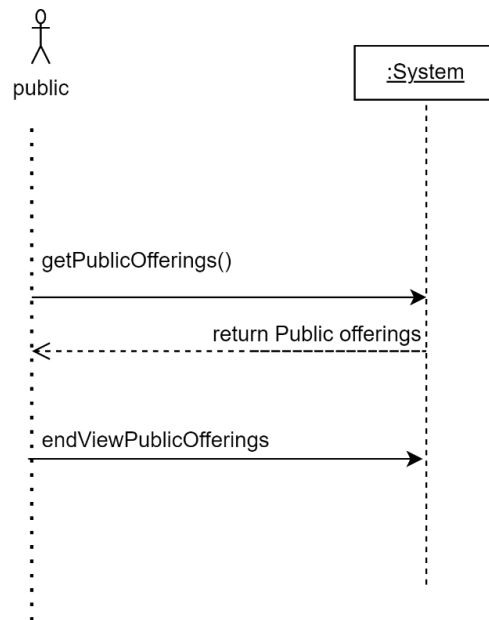
Preconditions:

- The offering must exist and be eligible to be made public. The ID was processed in the takeOffering method before the method called itself to the system.

Postconditions:

- The offering is marked as public in the system.

## Part III: Sequence diagram from instructor to system



### System operations:

`getPublicOfferings()`

### operation contacts:

1.

contract CO1: `getPublicOfferings`

Operation: `getPublicOfferings ()`

Reference: sequence diagram part III

### Preconditions:

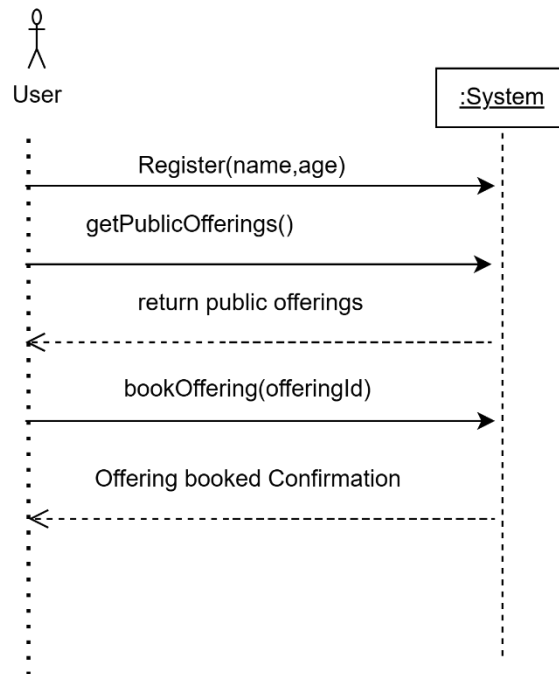
- The system is operational and available to respond.
- The public user has access to request public offerings.
- There must be available public offerings in the system.

### Postconditions:

- A list of public offerings is returned to the public user.
- The system logs the request for public offerings.
- The public user enters the state of viewing available offerings.

## Part IV: Sequence diagram from user to system

### a) If the user is above 18 years old



#### **System operations:**

Register(name,age)

getPublicOfferings()

bookOffering(offeringid)

#### **operation contacts:**

1.

contract CO1: Register

Operation: Register(name,age)

Reference: sequence diagram part IV (a)

Preconditions:

- The name must be non-empty.
- The age must be above a specified minimum requirement for registration.

Postconditions:

- A new Client instance is created with name and age.
- The Client instance is added to the system's list of registered clients.

2.

contract CO2: getPublicOfferings

Operation: getPublicOfferings()

Reference: sequence diagram part IV (a)

Preconditions:

- The client must be registered in the system.

Postconditions:

- The system retrieves all instances of Offering where isPublic is true.
- Returns a list of public offerings to the client.

3.

contract CO3: bookOffering

Operation: bookOffering(offeringid)

Reference: sequence diagram part IV (a)

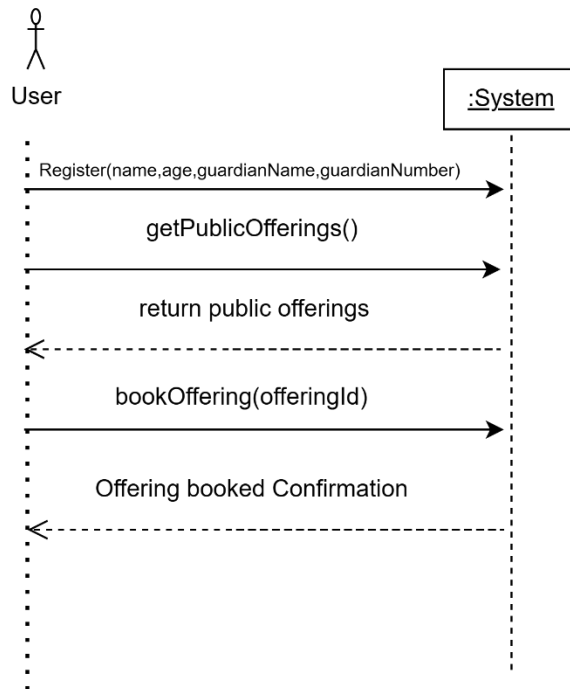
Preconditions:

- The client must be registered in the system.
- The offering with offeringId exists and is public.
- The offering has available capacity.

Postconditions:

- A new Booking instance is created, linking the Client and the specified Offering.
- The booking confirmation is generated and returned to the client.
- The capacity for the selected offering is decreased by one.

**b) If the user is underage**



**System operations:**

Register(name,age, guardianName, guardianNumber)

getPublicOfferings()

bookOffering(offeringid)

**operation contacts:**

1.

contract CO1: Register

Operation: Register(name,age, guardianName, guardianNumber)

Reference: sequence diagram part IV (b)

Preconditions:

- name is a non-empty string.
- age is greater than or equal to the minimum age for clients. If age is under the age threshold for legal booking, guardianName and guardianNumber must be provided.



Postconditions:

- A new Client instance is created with the provided name, age, guardianName, and guardianPhoneNumber (if applicable).
- The Client instance is added to the system's list of registered clients.

2.

contract CO2: getPublicOfferings

Operation: getPublicOfferings()

Reference: sequence diagram part IV (b)

Preconditions:

- The client must be registered in the system.

Postconditions:

- The system retrieves all Offering instances where isPublic is set to true.
- Returns a list of public offerings to the client.

3.

contract CO3: bookOffering

Operation: bookOffering(offeringid)

Reference: sequence diagram part IV (b)

Preconditions:

- The client must be registered in the system.
- The offering with offeringId exists, is public, and has available capacity.
- If the client is underage, the client must have a legal guardian registered.

Postconditions:

- A new Booking instance is created, linking the Client and the specified Offering.
- The booking confirmation is generated and returned to the client.
- The capacity for the selected offering is decreased by one.