# Part I: Sequence diagram from instructor to system

A black background with white text

Description automatically generated

**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.

A black screen with white text

Description automatically generated

**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

A black screen with white text

Description automatically generated

**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.