

# System Operations and Operation Contracts

## **1. Operation: displayAvailableOfferings(ArrayList<Offering> offerings)**

- Class: Person
- Type: Static
- Cross-reference: Use Case 3 - "Public Views Offerings"
- Preconditions:
  - The system contains a list of offerings that are not fully booked.
- Postconditions:
  - The system retrieves and displays the list of available offerings for the public to view.

## **2. Operation: processOfferings()**

- Class: MySystem
- Cross-reference: System Process - Managing Offerings
- Preconditions:
  - The system must have a list of lessons in the offerings ArrayList.
- Postconditions:
  - The system processes the offerings, checks for conflicts or overlaps, and ensures the availability.

## **3. Operation: managesScheduling()**

- Class: MySystem
- Cross-reference: System Process - Managing Scheduling
- Preconditions:
  - The system contains active lessons and schedules.
- Postconditions:
  - The system ensures that schedules are allocated correctly, avoiding conflicts.

#### **4. Operation: displayWelcomeMenu()**

- Class: MySystem
- Cross-reference: System Startup - Initializing the User Interface
- Preconditions:
  - System is running and active.
- Postconditions:
  - Displays the welcome menu to the user upon system initialization.

#### **5. Operation: displayMenuOption(Person user)**

- Class: MySystem
- Cross-reference: System Interaction - Menu Options Based on User Role
- Preconditions:
  - User is authenticated and identified by the system.
- Postconditions:
  - Displays menu options based on the user type (Admin, Instructor, Client).

#### **6. Operation: run()**

- Class: MySystem
- Cross-reference: System Initialization and Startup
- Preconditions:
  - System is initialized and prepared to run.
- Postconditions:
  - Starts and runs the system's main functionality loop.

#### **7. Operation: close()**

- Class: MySystem
- Cross-reference: System Shutdown - Safe Closure
- Preconditions:
  - System is active and running.
- Postconditions:
  - Closes and shuts down the system safely.

**8. Operation: promptUserLoginOrCreateAccount(ArrayList<User> users, Scanner scan)**

- Class: LoginSystem
- Cross-reference: Use Case - Login or Account Creation
- Preconditions:
  - The system contains a list of users.
- Postconditions:
  - The system prompts the user to either log in or create an account.
  - If an account is created, the new user is added to the system.

**9. Operation: loginUser(ArrayList<User> users, Scanner scan)**

- Class: LoginSystem
- Cross-reference: Use Case - User Login
- Preconditions:
  - The user must have previously created an account.
- Postconditions:
  - The user is authenticated and logged into the system.

**10. Operation: createUser(ArrayList<User> users, Scanner scan)**

- Class: LoginSystem
- Cross-reference: Use Case - Account Creation
- Preconditions:
  - The user does not have an account in the system.
- Postconditions:
  - The user creates a new account, which is added to the system's user list.

**11. Operation: selectLesson(ArrayList<Offering> offerings, Scanner scan)**

- Class: Instructor
- Cross-reference: Use Case - Instructor Selects a Lesson
- Preconditions:
  - The system contains a list of available offerings.
- Postconditions:
  - The instructor selects a lesson to assign themselves to teach.

**12. Operation: addLesson(ArrayList<Offering> offerings, Scanner scan)**

- Class: Admin
- Cross-reference: Use Case - Admin Adds Lesson
- Preconditions:
  - The system contains available offerings.
- Postconditions:
  - The admin adds a new lesson offering to the system.

**13. Operation: bookLesson(Lesson lesson)**

- Class: Client
- Cross-reference: Use Case - Client Books a Lesson
- Preconditions:
  - The client must have viewed available offerings.
- Postconditions:
  - The client successfully books a lesson from the available offerings.