

System Operations and Operation Contracts

1. displayAvailableOfferings(ArrayList<Lesson> lessons)

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

2. processOfferings()

- Class: MySystem
- Type: System Process - Managing Lessons
- Cross-reference: Use Case 1 - "Process Lessons"
- Preconditions: The system must have a list of lessons in the lessons ArrayList.
- Postconditions: The system processes the lessons, checks for conflicts or overlaps, and ensures availability.

3. managesScheduling()

- Class: MySystem
- Type: System Process - Managing Scheduling
- Cross-reference: Use Case 2 - "Manage Scheduling"
- Preconditions: The system contains active lessons and schedules.
- Postconditions: The system ensures that schedules are allocated correctly, avoiding conflicts.

4. displayWelcomeMenu()

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

5. displayMenuOption(Person user)

- Class: MySystem
- Type: 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. run()

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

7. close(int exitCode)

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

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

- Class: LoginSystem
- Type: 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. loginUser(ArrayList<User> users, Scanner scan)

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

10. createNewUser(ArrayList<User> users, Scanner scan)

- Class: LoginSystem
- Type: 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. selectLesson(ArrayList<Lesson> lessons, Scanner scan)

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

12. addLesson(ArrayList<Lesson> lessons, Scanner scan)

- Class: Admin
- Type: Use Case - Admin Adds Lesson
- Preconditions: The system contains available lessons.
- Postconditions: The admin adds a new lesson to the system.

13. bookLesson(Lesson lesson)

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

14. cancelBooking(Booking booking)

- Class: Client
- Type: Use Case - Client Cancels a Booking
- Preconditions: The client must have a valid booking in the system.
- Postconditions: The booking is canceled, and the slot is made available for others.

15. updateLesson(DatabaseHandler dbHandler, Scanner scan)

- Class: Admin, Instructor
- Type: Use Case - Update a Lesson
- Preconditions: The lesson exists in the system.
- Postconditions: The lesson is updated with new details.