Khujista Faqiri

Yash Nathani

Identification of system operations and operation contracts

1. <u>SignUp() - for client</u>
    - Operation Contracts

- Pre-Conditions:
    1. The client must not have an account in the system.
    2. The Clerk must approve the request in order for the client to have an account.
- Post-Conditions:
    1. The client gets a confirmation message of account creation.
    2. The client has an active account and can proceed with making service requests

2. <u>ServiceRequest()-from client</u>
    - Operation Contracts

- Pre-Conditions:
    1. The client must have an approved and active account.
    2. The Client must be logged in in order to make the request
- Post-Conditions:
    1. The client can search up for services and make one or multiple choices for service.
    2. A confirmation message is returned to the client by the system.

3. <u>AddAvailabilitySchedule(time,service) -from Expert</u>
    - Operation Contracts

- Pre-Conditions:
    1. The Expert must be logged in using the right credentials.
    2. The Expert must search for services available.
    3. The Expert must indicate the type of services he/she wants to offer and select it.
    4. The expert must choose the time-slot where he/she can offer the services.
- Post-Conditions:
    1. The system stores the time-slot and services chosen.
    2. A confirmation message is returned to the Expert by the system.

4. Read()-For expert and client
    - Operation Contracts
- Pre-Conditions:
1. The Client/Expert must have an active account.
2. The Expert/Client must be logged in using the right credentials.
- Post-Conditions:
    3. The client/expert can view the displayed auctions, object of interest, auction schedule, etc.

5. find(id) - for Expert and client
    - Operation Contracts
- Pre-Conditions:
    1. The cid must be a valid identifier
    2. The system must be operational and able to access client records.

- Post-Conditions:
    1. If the cid exists in the system: The system returns the client's details (cid, name, contact_info).
    2. If the cid does not exist: The system returns "Not Exist" to indicate no matching record was found.

6. LogIn(Name, Email, Password) -Clerk and Expert
- Operation Contracts
    - Pre-Conditions:
        1. The Clerk/Expert must have an active and existing account in the system.
    - Post-Conditions:
        1. If the credentials are valid: Successful login message returned by the system.
        2. If the credentials are invalid: ERROR and TRY AGAIN message returned by the system.

7. LogIn (Username, Password)
- Operation Contracts
    - Pre-Conditions:
        1. The Client account must have been approved upon sign up.
        2. The Client account must be active and existing in the system.

- Post-Conditions:
    1. If the account is not approved upon Sign Up: Return “Approval Pending” message and ask the client to send a new approval request.
    2. If the account is approved upon SignUp:
        - Valid Credentials: Successful login message returned by the system.
        - Invalid Credentials: ERROR and TRY AGAIN message returned by the system.

8. <u>deleteAccount(id) - for Expert and client</u>
    - Operation Contracts
        - Pre-Conditions:
            1. The id must correspond to an existing Expert or Client in the system.
            2. The Administrator (Clerk) must have the necessary permissions to delete the account.
            3. The system must be operational and able to modify records.
        - Post-Conditions:
            1. If the id exists:
                - The Expert or Client account is permanently removed from the system.
                - Any associated services, schedules, or access permissions are revoked.
                - The system returns a confirmation message indicating successful deletion.
            2. If the id does not exist:
                - The system returns an error message, indicating that no such account was found.

9. <u>updateInfo(name,contact_info) - for Expert and client</u>
    - Operation Contracts
        - Pre-Conditions:
            1. The ID of the client or expert must exist in the system.
            2. The clerk must have the necessary permissions to modify the information.
            3. The new name and contact information must be valid and formatted correctly.
            4. The system must be operational and allow record modifications.

- Post-Conditions:
    1. If the ID exists:
        - The name and contact information are updated in the system.
        - The system sends a confirmation message to the clerk.
    2. If the ID does not exist:
        - The system returns an error message, indicating that no such user was found.

10. <u>approvalRequest()- for client</u>
    - Operation Contracts
        - Pre-Conditions:
            1. The client must have submitted a valid signup request.
            2. The client's account must be in a pending state.
            3. The clerk must have the necessary permissions to approve the account.
            4. The system must be operational to store the approval status.
        - Post-Conditions:
            1. If the approval is granted:
                - The client's status is updated to "approved" in the system.
                - A confirmation message is sent to notify the clerk.
                - The system returns a confirmation message indicating successful deletion.
            2. If the request does not exist or is already processed:
                - The system returns an error message, indicating that no pending request was found.

11. <u>AddObject() - Clerk/Administrator</u>

    - Operation Contracts
        - Pre-Conditions:
            1. The clerk must be logged in and should be authorized.
            2. The system must accept new objects and should store it in the database.
            3. All required fields must be completed.

        - Post-Conditions:

1. A new object is added in the object list and available for auction.
2. The system ends operation with a confirmation message.

12. Make_expert_account() - Clerk

- Operation Contracts
    - Pre-Conditions:
        1. The Clerk must be logged in and should be authorized to perform the operation.
        2. The Expert must not have already been registered in the system.
        3. The System allows new user entries and stores it in the database.
- Post-Conditions:
    1. The account of the expert has been created and successfully logged in the database,
    2. The expert can now add their availability in the system and offer their services.
    3. A confirmation message is sent upon successful registration of account.

OCL

```
1)context Expert::addAvailability(timeSlot: String)
inv:
   not Availability.allInstances()->exists(a |
      a.expert_id = self.id and a.time_slots = timeSlot and a.status = 'Booked')

2)context ServiceRequest::requestService()
inv:
   let auctionHouse : AuctionHouse = AuctionHouse.allInstances()->select(a | a.id =
self.auction_house_id)->first() in
     self.time_slot >= auctionHouse.schedule.start_time and self.time_slot <=
auctionHouse.schedule.end_time

3) context Client::requestService()
inv:
   let clientRequests = ServiceRequest.allInstances()->select(s | s.client_id = self.id) in
   clientRequests->forAll(r1, r2 |
      r1 != r2 implies
      (r1.time_slot < r2.time_slot.start_time or r1.time_slot.end_time >
r2.time_slot.start_time))
```