**Use Case 1: Sign Up**

**Use case ID:** UC1  
**Use Case Name:** Sign Up  
**Primary Actor:** Client

**Stakeholders and Interests:**

* **Client:** Wants to register quickly and securely to access services.
* **Admin:** Needs accurate info to approve or reject clients.

**Preconditions:**

* Client is not already registered.

**Success Guarantee (Postconditions):**

* Client's info is stored in the system in “Pending” status.

**Main Success Scenario (Basic Flow):**

1. Client navigates to the sign-up option.
2. Client enters email, password, name, phone number, affiliation, and intent.
3. System validates the input format.
4. System stores client info with status “Pending.”
5. Admin is able to see the signup request.

**Extensions (Alternative Flows):**

* If required fields are missing → Prompt to complete.

**Special Requirements:**

* Secure password entry

**Technology and Data Variations:**

* Email may be entered in uppercase or lowercase → system stores normalized.

**Open Issues:**

* Should the client receive a confirmation email?

**Use Case 2: Log In**

**Use case ID:** UC2  
**Use Case Name:** Log In  
**Primary Actor:** Client

**Stakeholders and Interests:**

* **Client:** Wants quick, secure access to the system.

**Preconditions:**

* Client account is approved by the admin.

**Success Guarantee (Postconditions):**

* Client is logged into the system.

**Main Success Scenario:**

1. Client enters email and password.
2. System verifies credentials.
3. System checks account status = “Approved.”
4. System grants access and loads dashboard.

**Extensions:**

* If password is incorrect → Show error.
* If account is not approved → Show “Pending approval” message.

**Open Issues:**

* Should we hide the password field?

**Use Case 3: Search Auctions**

**Use case ID:** UC3  
**Use Case Name:** Search Auctions  
**Primary Actor:** Client

**Stakeholders and Interests:**

* **Client:** Wants to view auctions of interest.

**Preconditions:**

* Client is logged in.
* Client knows the auction especially that they want to search for.

**Success Guarantee:**

* A list of auctions matching the specialty is shown.

**Main Success Scenario:**

1. Client chooses the “search auction” option from the main menu.
2. Client chooses the auction specialty he/she wants.
3. System queries database and fetches results.
4. Results are displayed to the client.

**Extensions:**

* If no auctions match → Show “no results” message.

**Special Requirements:**

* Search should support input written both in lowercase and uppercase.

**Open Issues:**

* Should we support sorting by date?

**Use Case 4: Search Objects of Interest**

**Use case ID:** UC4  
**Use Case Name:** Seach Objects of Interest  
**Primary Actor:** Client

**Stakeholders and Interests:**

* **Client:** Wants to see search before requesting services.

**Preconditions:**

* Client is logged in.

**Success Guarantee:**

* Object details matching object type requested are displayed clearly.

**Main Success Scenario:**

1. Client selects search objects of interest.
2. Client inputs the object type.
3. System retrieves objects matching the object type.
4. System displays the matching objects.

**Extensions:**

* If no objects available → Show message.

**Open Issues:**

* Should clients be allowed to see who owns the objects?

**Use Case 5: Request Service**

**Use case ID:** UC5  
**Use Case Name:** Request Service  
**Primary Actor:** Client

**Stakeholders and Interests:**

* **Client:** Wants expert guidance about objects of interests.
* **Expert:** Needs availability requests.

**Preconditions:**

* Client is logged in.

**Success Guarantee:**

* A service request is created and stored in the expert account.

**Main Success Scenario:**

1. Client selects “request service” from the options menu.
2. Client writes the name of the object he/she wants advice for.
3. Client selects desired time and service type.
4. System stores the requests in the pool of requests for experts.

**Extensions:**

* If request overlaps with an existing one → Reject with warning.

**Special Requirements:**

* Conflict-checking for overlapping service requests

**Use Case 6: Buy Object of Interest**  
**Use case ID:** UC6  
**Use Case Name:** Buy Object of Interest  
**Primary Actor:** Client

**Stakeholders and Interests:**  
• Client: Wants to securely purchase an available object with confirmation and clarity.

**Preconditions:**  
• Client is logged in.  
• Object of Interest is marked as available for purchase.

**Success Guarantee (Postconditions):**  
• The object’s status is removed from the database  
• The client receives confirmation of purchase.

**Main Success Scenario (Basic Flow):**

1. Client navigates to “Buy Objects of Interest section”.
2. Client selects the object desired among the displayed objects.
3. System processes the transaction.
4. System removes the object from the database
5. System displays a confirmation and purchase summary to the client.

**Extensions (Alternative Flows):**  
• If object is no longer available → Show error and remove from listing.

**Special Requirements:**  
• Secure and atomic transaction recording

**Open Issues:**  
• Should clients be able to cancel or request a refund after purchase?