

## Use Case UC-<> :- <>

**Related Requirements:** REQ-<>, REQ-<>.

**Initiating Actor:** <>

**Participating Actors:** <>

**Actor's Goal:** <>.

**Preconditions:**

- <>
- <>
- <>

**Postconditions:**

- <>

**Flow of Events for Main Success Scenario:**

- ← <>
- → <>

## Use Case UC 1:- login

**Related Requirements:** REQ-1, REQ-2, REQ-3, REQ-4, REQ-5, REQ-10

**Initiating Actor:** Registered User

**Participating Actors:** Database , Main UI

**Actor's Goal:**

- Login in successfully and retrieve personal information about the registered user.

**Preconditions:**

- The actor is registered.
- The actor is not logged in.

**Postconditions:**

- The system redirects to the homepage of the user.

**Flow of Events for Main Success Scenario:**

- → Actor enters data according to his/her previously stored data .
- → User Interface calls login() method.
- → System gets the user's data from the database.
- ←System redirects to his/her homepage on success.
- ← System notifies the actor upon wrong entered data.

## Use Case UC 2:- signUp

**Related Requirements:** REQ-1, REQ-2, REQ-3, REQ-4, REQ-5, REQ-10

**Initiating Actor:** Unregistered User

**Participating Actors:** Database, UI

**Actor's Goal:**

- Sign up successfully and store his/her personal information in the website's database.

**Preconditions:**

- The actor is unregistered.

**Postconditions:**

- The system redirects to the homepage of a newly created user.

**Flow of Events for Main Success Scenario:**

- → User click the Register button
- → User enters data about him/her -self
- → User Interface call methods signUp() method .
- → System stores the actor's data in the database.
- ← System redirects to his/her homepage on success.
- ← System notifies the actor upon violating entered data.

### Use Case UC 3:- logout

**Related Requirements:** REQ-8, REQ-9, REQ-11

**Initiating Actor:** Registered User

**Participating Actors:** Website

**Actor's Goal:**

- No longer be treated as a registered user.

**Preconditions:**

- The actor is registered.
- The actor is logged in.

**Postconditions:**

- The system redirects to the main page, treats actor as a guest.

**Flow of Events for Main Success Scenario:**

- User clicks on logout.
- Main UI calls the logOut() method to remove the user session from the system.
- Guest View is returned to the now logged out user.

#### Use Case UC 4:- assignExperts

**Related Requirements:** REQ-4, REQ-5, REQ-6, REQ-12, REQ-13, REQ-14

**Initiating Actor:** Registered User

**Participating Actors:** Admin, Database, User Interface

**Actor's Goal:**

- Become an expert, with certain privileges.

**Preconditions:**

- A user applied for an expert position.

**Postconditions:**

- The system treats actor as an expert and offers other functionalities.

**Flow of Events for Main Success Scenario:**

- Admin approves the expert request, assign the registered user to become an expert.
- System treats user as an expert and offers expert's functionalities.

### Use Case UC 5:- assignAdmin

**Related Requirements:** REQ-4, REQ-5, REQ-6, REQ-16, REQ-17, REQ-18

**Initiating Actor:** Registered User

**Participating Actors:** Owner

**Actor's Goal:**

- Create and possess an admin account with further access.

**Preconditions:**

- The user is registered.
- The actor already applied for adminship.

**Postconditions:**

- The system treats the user as an admin and adds him/her to the dashboard with more access.

**Flow of Events for Main Success Scenario:**

- User applies for being an admin.
- Owner accepts user's adminship request.
- User account now have further access to further functionalities.

### Use Case UC 6:- search

**Related Requirements:** REQ-5, REQ-18, REQ-19

**Initiating Actor:** Registered User

**Participating Actors:** Database,main UI

**Actor's Goal:**

- To search for experts by topic or name.

**Preconditions:**

- The user is registered.
- The system displays the homepage with a search area.

**Postconditions:**

- The system redirect to a page that contains all experts from the search.

**Flow of Events for Main Success Scenario:**

- →User selects a name or topic to search by on main UI and press search.
- →User Interface makes call search(string): arraylist<expert>.
- ←The search function will retrieve the data from the database
- →The Database will then return Record objects
- ← main UI redirect to a page that contains all experts from the search.

--

Use Case UC 18:- createTopic
<p><b>Related Requirements:</b> REQ-6, REQ-51</p> <p><b>Initiating Actor:</b> Expert User</p> <p><b>Participating Actors:</b> Database</p> <p><b>Actor's Goal:</b> To create a topic he/she wants to talk about.</p> <p><b>Preconditions:</b></p> <ul style="list-style-type: none"><li>• The User is an expert.</li></ul> <p><b>Postconditions:</b></p> <ul style="list-style-type: none"><li>• System pop up a message that the topic was added.</li><li>• The system add the topic to the expert information.</li><li>• Users can search for expert by this topic.</li></ul> <p><b>Flow of Events for Main Success Scenario:</b></p> <ul style="list-style-type: none"><li>• → User(Expert) selects a topic to add on main UI using createTopic method .</li><li>• → System adds the topic to the expert information in the database.</li><li>• ← System pop up a message that the topic was added.</li></ul>



### **Use Case UC-19 :- expertApply**

**Related Requirements:** REQ-52

**Initiating Actor:** User

**Actor's Goal:** To apply for being an Expert user in the website.

**Preconditions:**

- The user must be a registered user.

**Postconditions:**

- A request is sent to the admin to whether approve or disapprove the applicant.

**Flow of Events for Main Success Scenario:**

- →The user clicks on a button to apply in the settings.
- →System asks to specify the field he/she wants to be an expert in.
- →System sends request by email using expertApply() method to the Admin.

## Use Case UC-20 :- expertTopics

**Related Requirements:** REQ-6, REQ-53

**Initiating Actor:** Expert

**Participating Actors:** Database/UI

**Actor's Goal:** The Expert can select a topic to be added to his/her list to be available for discussion in office hours.

**Preconditions:**

- That the user is an Expert.
- Database has existing topics to provide.
- Expert is adding to his list of topics through settings.

**Postconditions:**

- The chosen topic is added to Expert's list of topics in his/her profile.

**Flow of Events for Main Success Scenario:**

- The Expert selects a topic to add to his/her list.
- A list of topics is viewed to the expert from database to choose from.
- The selected topic is added to the expert's list of topics.

### Use Case UC-21 :- requestSession

**Related Requirements:** REQ-44, REQ-45, REQ-61, REQ-63.

**Initiating Actor:** User

**Participating Actors:** Expert, Database, User interface

**Actor's Goal:** Request a session with a certain expert .

**Preconditions:**

- The user has already asked a question
- The expert has accepted the user's question
- The user has to be in the expert's profile

**Postconditions:**

- Pop-up message will appear to notify the user that the request has been successfully sent

**Flow of Events for Main Success Scenario:**

→ The user will click on the button reserve session

→

- The expert will reply with 3 free slots

## Use Case UC-22 :- sessionType

**Related Requirements:**REQ-71

**Initiating Actor:** User

**Actor's Goal:** The user will specify the session's type in the issued request whether public or private

**Preconditions:**

- The user has to specify the session's type ( public or private )
- The user has already requested to reserve a session

**Postconditions:**

- The request will be sent to the expert

**Flow of Events for Main Success Scenario:**

- The user will request to reserve a session with the expert
- The user will specify the type of the session
- The request will be sent to the expert

### Use Case UC-23 :- askQuestion

**Related Requirements:** REQ-43, REQ-63, REQ-67

**Initiating Actor:** Users

**Participating Actors:** Experts

**Actor's Goal:** The user are able to send a question to their desired experts.

**Preconditions:**

- The user selected a list of desired experts to seek help from.

**Postconditions:**

- A question to be sent to the desired experts from the user.

**Flow of Events for Main Success Scenario:**

- User select 1-3 experts then press Ask Question button.
- UI shows the question Form
- User type question then press the send button
- UI Calls AskQuestion method
- Then controller sends the question to the UI with the Experts ID
- Then the UI Shows the Question to the experts

### Use Case UC-24 :- pickSlot

**Related Requirements:** REQ-6, REQ-63, REQ-70, REQ-72.

**Initiating Actor:** User

**Participating Actors:** Database , User interface

**Actor's Goal:** The user will pick one of the received free slots sent by the expert

**Preconditions:**

- The user has already requested a session with an expert
- The expert has already replied with 3 free slots

**Postconditions:**

- A pop-up message will appear to the user to confirm the picked slot
- The picked slot will be added to the user schedule

**Flow of Events for Main Success Scenario:**

- ←
- → The user will press one of the slots sent by the expert .
- ← The user interface will send a pop-up message to the user to confirm the picked slot
- → The user interface will call the pickSlot() method
- The expert will receive a notification about this reservation.

### Use Case UC-25 :- sendURL

**Related Requirements:** REQ-62

**Initiating Actor:** System

**Participating Actors:** Expert,User

**Actor's Goal:** Both users and experts will receive chat room URL .

**Preconditions:**

- There is already a reserved session between the user and the expert.
- The session has already been confirmed by both the user and the expert

**Postconditions:**

- The URL will be sent to both the expert and the user by an email through third party API

**Flow of Events for Main Success Scenario:**

- The expert and the user will confirm the reserved session
- the system will send the URL of the chat room to both the user and the expert

## Use Case UC-26 :- respondToQuestion

**Related Requirements:** REQ-6, REQ-43

**Initiating Actor:** Expert

**Participating Actors:** Main UI

**Actor's Goal:** The expert to respond to a question sent by a user by either accept and answer or reject.

**Preconditions:**

- The Expert is logged in and viewing his questions' requests

**Postconditions:**

- A pop-up message will appear to the user by the name of the expert he asked and the response to his question.

**Flow of Events for Main Success Scenario:**

- →Expert clicks the button View Questions.
- ←System Load the web-page which contains the Question asked to this Expert.



- → Expert browses the requests.
- → Expert clicks a button associated with a specific question.
- ← System Calls the respondToQuestion and log the data to the database and send a confirmation message in a popup to the Expert.

#### **Use Case UC-27 :- sendSlots**

**Related Requirements:** REQ-4, REQ-6, REQ-44, REQ-45

**Initiating Actor:**Expert

**Participating Actors:** User

**Actor's Goal:** The expert to send 3 free slots to the user.

**Preconditions:**

- The user has already asked an expert a question.
- The expert accepted the question.
- The user has sent a request to the expert.

**Postconditions:**

- A message to be sent to the user from the expert with the 3 free slots.

**Flow of Events for Main Success Scenario:**

- Expert sends 3 free slots to the user.
- The message reached to the user to pick one of them.

**Use Case UC-43 :- videoCall**

**Related Requirements:** REQ-22, REQ-62, REQ-64, REQ-65.

**Initiating Actor:** User / Expert.

**Participating Actors:** UI/Database

**Actor's Goal:** Have a real-time video conference with the other user.

**Preconditions:**

- Have a previous reservation assigned with this time logged in the database and linked to a specific URL.

**Postconditions:**

- A pop-up to each user after the conference to rate each other and leave a feedback (if any).

**Flow of Events for Main Success Scenario:**

- ←User opens the URL associated with this reserved session.
- →The session starts after the other party join the chat room.
- The conference happens and then either of the users ends the conference the popup shows up and each user rate the other and give a feedback (if any) and then the data logged in the database.

## Use Case UC-44 :- voiceCall

**Related Requirements:** REQ-62, REQ-64, REQ-68.

**Initiating Actor:** User / Expert.

**Participating Actors:** UI/Database

**Actor's Goal:** Have a real-time Voice call with the other user.

**Preconditions:**

- Have a previous reservation assigned with this time logged in the database and linked to a specific URL.

**Postconditions:**

- A pop-up to each user after the conference to rate each other and leave a feedback (if any).

**Flow of Events for Main Success Scenario:**

- ←User opens the URL associated with this reserved session.
- →The session starts after the other party join the chat room.
- The conference happens and then either of the users ends the conference the popup shows up and each user rate the other and give a feedback (if any) and then the data logged in the database.

## Use Case UC-45 :- chat

**Related Requirements:** REQ-62 ,REQ-69

**Initiating Actor:** Users / Experts

**Participating Actors:** Users / Experts

**Actor's Goal:** The user/expert are able to send/receive text messages from each others.

**Preconditions:**

- The expert accepted the question of a specific user concerning a topic.
- The user/expert has an ongoing session connected over a certain session type.

**Postconditions:**

- The sent messages from each participant will be viewable by the rest of the participants present in the session.

**Flow of Events for Main Success Scenario:**

- User/Expert types message to be sent and presses the send button
- User/Expert receives text messages reached from other participants accordingly.

## Use Case UC-46 :- viewRequest

**Related Requirements:** REQ-73 ,REQ-74

**Initiating Actor:** User

**Participating Actors:** User

**Actor's Goal:** The User are able to view all requests sent/received

**Preconditions:**

- The User has sent/received requests.

**Postconditions:**

- The User is able to view requests

**Flow of Events for Main Success Scenario:**

- User press view requests button
- The UI calles viewRequests method
- Database sends back the requests to UI
- UI Displays the requests for the user