Design of tests

Users test cases:

**Signup**

**Test covers:** Graphical User Interface and Data manager.

**Assumption:** User has filled registration form and is on the correct screen.

**Integration Testing**

**Steps:**

* Press the register now button to sign up.
* Data manager receives signup information.

**Expected:** You get redirected to the main page (view) as a logged in user.

**Fails if:**

* User gets a pop-up message saying that there was a problem while trying to register.
* User doesn’t get redirected to the main page.

**Login**

**Test covers:** Graphical User Interface.

**Assumption:** User has entered his login data and is on the correct screen.

**Integration Testing**

**Steps:**

* User presses the login button.

**Expected:** User gets redirected to the main page and His username appears on the upper right side of the screen and a logout button appears.

**Fails if:**

* User is notified that there was a problem while logging in.
* User doesn’t get redirected to the main page (where he can access his other functionalities)

**Logout**

**Test covers:** Graphical User Interface.

**Assumption:** The application is showing the correct screen.

**Integration Testing**

**Steps:**

* User presses the logout button.

**Expected:** User gets redirected to the basic view where he can either signup or login.

**Fails if:**

* User is notified that there was a problem while logging out.
* User doesn’t get redirected to the basic view (where he can either signup or login)

**Editing personal information**

**Test covers:** Graphical User Interface and Data manager.

**Assumption:** User has changed the information he desires and is on the correct screen.

**Integration Testing**

**Steps:**

* User presses the update button.
* Data manager receives modified information.

**Expected:** The same profile page refreshes showing the changes the user entered.

**Fails if:**

* User is notified that there was a problem while updating/editing his profile.

**Sending upgrading request**

**Test covers:** Graphical User Interface.

**Assumption:** User uploaded his portfolio/CV and is on the correct screen.

**Integration Testing**

**Steps:**

* User presses the upgrade button.

**Expected:** A pop-up message appears that the request has been sent successfully.

**Fails if:**

* A pop-up message appears that there was a problem while submitting his request.

Create Tags

**Test covers** : Graphical User Interface

**Assumption**: The application is showing the correct screen

**Steps:**

* Press the button to create a new tag

**Expected**: A pop message is received that a new tag is added

**Fails if:**

* Pressing the create button doesn’t add the tag and no pop messages is received

Search for experts

**Test covers:** Graphical User Interface

**Assumption:** A search bar is found for the tag to be typed in

**Steps:**

* Type in a certain tag
* After typing in the tag press the search button

**Expected:** A list of experts is displayed which are related to the chosen tag

**Fails if:**

* Pressing the search button doesn’t display anything to the user

View expert’s profile

**Test covers:** Graphical User Interface

**Assumption:** list of experts is displayed

**Steps:**

* Choose an expert
* Press the view profile button

**Expected:** The user is redirected to the expert’s profile where all information related to this expert is displayed

**Fails if:**

* Pressing the view profile button doesn’t show the user the selected expert’s profile

Choose experts

**Test covers:** Graphical User Interface

**Assumption:** list of experts is displayed

**Steps:**

* Press the Choose button next to each desired expert
* Choose up to 3 experts

**Expected:** A pop message is sent after choosing each expert and another one if the number of chosen experts is greater than three

**Fails if:**

* Pressing the choose button doesn’t successfully choose the selected expert
* Choosing more than three experts for a specific tag

| **User rates the chosen expert** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** User has just finished a session with an Expert  **Integration Testing**  **Steps:**   * User selects the rating of the Expert * User presses the submit rating button   **Expected:** User receives a pop up message that his rating has been submitted  **Fails if:**   * User presses the submit rating button and does not get a pop up message |

| **User accepts/declines a slot from the suggested ones** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** User has found an expert in the desired field with whom he wishes to have a session  **Integration Testing**  **Steps:**   * User selects desired expert * User presses the send request button   **Expected:** User gets a pop up message that his request has been sent  **Fails if:**   * User presses the send request button and receives no pop up message that his request has been sent |

| **User views users profile** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption: User has searched for another user**  **Integration Testing**  **Steps:**   * User clicks on view profile button   **Expected:** User gets a view of other desired user’s profile  **Fails if:**   * User presses the view profile button yet the view doesn’t change to the desired user’s profile |

| **User sends request to become an expert** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:**User is on his profile view  **Integration Testing**  **Steps:**   * User selects send expert request * User fills the form and attaches his CV * User clicks on send request   **Expected:** User receives a pop up message that his request has been sent  **Fails if:**   * User clicks on send request and does not get a pop up message |

| **User views sessions schedule** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** User is on his profile view  **Integration Testing**  **Steps:**   * User selects view schedule * Expected: view of schedule comes up   **Fails if:**   * User selects view schedule and view does not change |

**Expert Test Cases**

| Expert Adds/Edits tags |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** User has been approved as an Expert and is adding expert functionalities.  **Integration Testing**  **Steps:**   * Expert presses the add/Edit Tag Button * Expert selects add tag button and the tag that represents his area of expertise or selects edit tag button and edits previously selected tags   **Expected:** Tag will be added to expert’s profile  **Fails if:**   * Pressing the buttons does not add/edit tags |

| **Expert rates user** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** Expert has just finished a session with a user  **Integration Testing**  **Steps:**   * Expert selects the rating of the user * Expert presses the submit rating button   **Expected:** Expert receives a pop up message that his rating has been submitted  **Fails if:**   * Expert presses the submit rating button and does not get a pop up message |

|  |
| --- |
| **Create an appointment with the user** |
| **Test Covers:** Graphical User Interface and Data Manager.  **Assumption:** User accepted a slot from the suggested ones.  **Integration Testing**  **Steps:**   * Press on the book button and provide the information of the session with the agreed on slot. * The Graphical User Interface passes this information to data manager to be saved in the session table.   **Expected:** The appointment should be created and saved.  **Fails if:**  **•** The session record was not saved. |

|  |
| --- |
| **Receive Notifications** |
| **Test Covers: Graphical User Interface.**  **Assumption: Expert received a notification.**  **Integration Testing**  **Steps:**   * **Press to view the notifications.**   **Expected: Notification can be viewed.**  **Fails if:**  **• No notification is available to be viewed.** |

|  |
| --- |
| **Suggesting slots for the appointment** |
| **Test Covers:** Graphical User Interface.  **Assumption:** User confirms on the session’s slot.  **Integration Testing**  **Steps:**   * Expert suggests 3 slots. * Send email to the user with the slots.   **Expected:** A message appears that the email was sent successfully.  **Fails if:**  **•** The message says that email failed to be sent. |

|  |
| --- |
| **Receive a confirmation and the Url for the session** |
| **Test Covers:** Graphical User Interface.  **Assumption:** User confirms on the session’s slot.  **Integration Testing**  **Steps:**   * Open inbox.   **Expected:** A confirmation and the URL are received in the inbox.  **Fails if:**  **•** No confirmation is sent. |

|  |
| --- |
| **URL opens chat room** |
| **Test Covers:** Graphical User Interface.  **Assumption:** URL is received.  **Integration Testing**  **Steps:**   * Open the confirmation email in the inbox. * Press on the URL hyperlink.   **Expected:** URL opens the chat room/video conference.  **Fails if:**  **•** The URL doesn’t redirect to the session’s platform. |

Admin test cases:

| **Admin blocking tags** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** Admin has found a tag that he does not believe should be visible to people  **Integration Testing**  **Steps:**   * Admin selects unwanted tag * Admin presses the block tag button   **Expected:** Tag gets removed from the list of tags  **Fails if:**   * Admin presses the block tag button and tag does not get removed from the list of tags |

| **Admin editing tags** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** Admin has found a tag that he does not believe should be edited  **Integration Testing**  **Steps:**   * Admin selects wanted tag * Admin presses the edit tag button * Admin types id the new tag name * Admin presses save tag button   **Expected:** Tag gets edited in list of tags  **Fails if:**   * Admin presses the edit tag button and types in then saves the new data and tag does not get updated in the list of tags |

| **Admin views potential experts CV** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** Admin has received a request from a user to become an expert  **Integration Testing**  **Steps:**   * Admin goes to requests * Admin chooses the request * Admin clicks on view CV   **Expected:** User’s CV opens  **Fails if:**   * Admin clicks on the view CV button and the user’s CV does not open |

| **Admin bans user** |
| --- |
| **Test covers:** Graphical User Interface  **Assumption:** Admin has found a account that has a very low rating or has been performing miscellaneous activity  **Integration Testing**  **Steps:**   * Admin goes to user’s profile page * Admin clicks on ban user button   **Expected:** Admin gets a pop up message that the user has been banned  **Fails if:**   * Admin clicks on the ban user button and gets no pop up message |