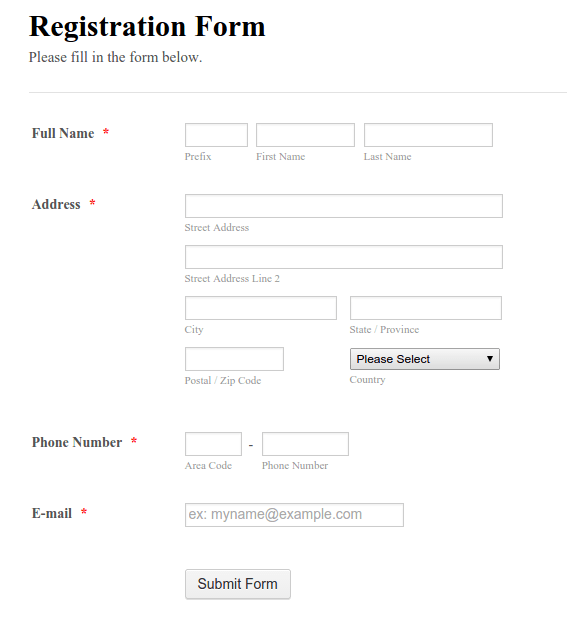
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Priority** | **Preconditions** | **Input test data** | **Steps to be executed** |
| 1 | Test if user is able to login successfully. | A | User must be registered already | correct username,correct password | 1)Enter input(correct )username and password on the respective fields 2)click submit/login |
| 2 | Test if unregistered users is not able to login to the site | A |  | incorrect username,incorrect password | 1)Enter input(incorrect )username and password on the respective fields 2)click submit/login |
|  | Test with valid username and empty password such that login must get failed | B | User must be registered already | valid username and empty password | 1)enter the valid username in the user id and enter no password in the password field |
| 4 | Test with empty username and valid password such that login must get failed | B | registered user's password | empty username and valid password | 1)leave the username empty in the user id and enter a valid user's password in the password field |
| 5 | Test with empty username and empty password and check if login fails | A | - | - | 1)Enter nothing in the mail id and password field 2)click submit button |
| 6 | Check of the password is masked on the screen i.e., password must be in bullets or asterisks | B |  | some password(can be a registered/unregistered) | 1) Enter the password field with some characters |
| 7 | Check if the login function handles case sensitivity | B | registered user's password which is originally in lower case changed to upper case or vice versa | case changed username /password | 1)Enter the case changed username /password in the respective field and 2)click login button |
| 8 | After logging in try to copy/cut the password and paste it on another screen(passwords are usually in \* such that its not visible on the screen) | B |  | Registered user's login id and password | 1)Enter username and password in the respective fields. Copy the password field's content(which is in \*s) 3)paste the content on another screen |
| 9 | Verify account lock | B |  | Registered user's login id and incorrect password | 1)Try to login with a registered user name and incorrect password for more than 3 times |
| 10 | Check if on selecting back button (after logging out) if the user is not signed in | B |  | Registered username and password | 1)Login with registered username and password 2)once your are logged in, sign out of the site 3)now press back button |
| 11 | Verify the url without logging into to the site | B |  | Registered username and password | 1) Login to the site using registered username and password 2)copy and save the url of the logged in page 3)logout of the site 4)now paste the copied url on the browser |
| 12 | Automatic logout of the site when pressing backspace button | B | User must be registered already | Registered username and password | 1) Login to the site using registered username and password 2)now press backspace |

**Sample Test Cases for User Registration Form**

Registration Form Test Cases

**Preconditions:**

1. The zipcode needs to be a number.
2. The phone number is 10 digit and the area code needs to be min. 3 number and maximum 5 numbers.
3. If required fields are not filled or wrong data is inserted then jotform triggers error after submission of the form.

The test cases for the registration form are sorted per field as explained below.

Generic Form Field Tests

* Check the behavior of form by not filling up any data into the form.
* Check the by not filling up the required fields.
* Check the behavior of form by adding random data in the text field.
* Check the by not filling up the full name text field but by filling up rest of the other fields.
* Check the behavior of form by filling other text fields except address text fields.
* Check the form by filling other text fields except phone number text field.
* Check the form by filling other text fields except email text field.

Testing Full Name Field

* Check the Full name text field without Prefix.
* Check the Full name text field without adding First name.
* Check the Full name text field without adding Last name.
* Check the Full name text field with special characters in Prefix.
* Check the Full name text field with special characters in First name.
* Check the Full name text field with special characters in Last name.
* Check by adding numbers instead of string in the full name text field.

Testing Address Field

* Check if all the text fields in the address field are mandatory or not.
* Check if the second line of the street field can be kept blank.
* Check the first line of the street text field in address by adding nothing into it.
* Check the city text field by adding numbers into it.
* Check the city text field by adding no data.
* Check the state text field by adding no data.
* Check the state text field by adding numbers into it.
* Check the state text field by adding data more than the field limit.
* Check the behavior of form by not selecting the country from the drop-down menu of the country field.
* Check the behavior of the form by selecting wrong country to that of address field content.
* Check the zipcode text field with string instead of numbers.
* Check the zipcode text field with numbers shorter than required.
* Check the zipcode text field with numbers larger than required for the specific state or place.

Testing Phone Number Field

* Check the Phone text field with area code being string instead of numbers.
* Check the Phone text field with phone number being string instead of numbers.
* Check the Phone text field with both area code and phone number both being string instead of numbers.
* Check the Phone text field with no data added.

Testing Email Field

* Check the Email text field that has Email address without @ symbol.
* Check the Email text field that has random string instead of real email.
* Check the Email text field that has @ symbol written in words.
* Check the Email text field that has missing dot in the email address.
* Check the Email text field as “name@gmail”
* Check the Email text field as “@gmail”
* Check the Email text field as “name@gmail..com”
* Check the Email text field as “name@192.168.1.1.0”
* Check the Email text field as “name.. @gmail.com”

I1 Enter empty value for First Name Display - “Name required” Display error message “Name required” √ 2 Enter empty value for Last Name Display - “Please enter your Last Name”

3 Enter empty value for User ID Display error message “User ID required”

4 Enter empty value for /Passport Display error message “NRIC/Passport required.”

5 Enter empty value for Password and Confirm Password Display error message “Password required.”

6 Enter empty value for either Password or Confirm Password Display -“Password does not match.”

7 Enter empty value for Contact Number Display error message “Enter valid contact number”

8 Enter nonnumeric value in Contact Number field abcde Display error message “Invalid contact number” √

9 Enter empty value for Email Address Display error message “Enter your email address”

10 Enter empty value for Confirm Email Address Display error message “Email address does not match” 11 Enter empty value for Address No error No error √

12 Enter password less than 8 characters long 123456 - “Password length must be 8 characters”

13 Enter different email format instead of xxx@xxx.xxx Yahoo.com.sg “Invalid email address”

14 Enter duplicate User ID Display error message “User already exist.”

**General**

* Activate all logs on the server (PHP, MySQL, Magento, mail, etc)
* Check meta tags in HTML
* Check favicon
* Check merging of CSS
* Check merging of JS
* Check Google Analytics
* Check for Javascript errors in Console
* Check header - all blocks present?
* Check footer - all blocks present?
* Check homepage
* Test newsletter subscription block
  + For new subscriber
  + For existing subscriber

**Catalog**

* Click through all category pages and check for;
  + Layered navigation showing and working correctly
  + Sorting working correctly
  + Pager working correctly
  + Check breadcrumbs
* Open at least 30 product pages and check for;
  + Correct price
  + Add to cart button
  + Configurable options, if applicable
  + Images
* Check working of search
* Check working of Advanced Search

**Checkout**

**Cart**

* Add item to cart
* Add multiple items to cart
* Test Remove item button
* Test Update shopping cart button
* Test Clear shopping cart button
* Check correct price
* Check correct totals (discount/tax/etc)
* Test Proceed to checkout button
* Check pop-out shopping cart in top right, if applicable

**Onepage**

* Check if login works
* Place a lot of orders;
  + Multiple products
  + With discounts
  + Let the transaction fail to see how Magento handles canceled orders
  + Try all combinations of payment methods & shipping methods
  + Try guest checkouts as well as logged-in checkouts
* Check if all fields that should be collapsed are collapsed

**OneStepCheckout**

* All shipping methods available?
* All payment methods available?
* All enabled countries correct?
* Are the AJAX filled fields loading correctly?
* Does the login link work?
* No license warning showing?

**My Account**

* Login & logout
* Are the correct My Accounts options showing?
* Reorder an order
* Add/edit address
* Subscribe / unsubscribe for newsletter

**Extensions**

* Walk through all extensions and find out what their purpose is. This takes up the most of the work, but most extensions will have test cases that coincide with above test cases. For example; by placing an order you'll probably also test whether the order is sent to the warehouse or financial management system.

**Backend**

* Clear cache
* Reindex all indexers
* Place an order through the backend
* Create a customer through the backend
* Open all pages to see whether exceptions occur
* Check all configuration options for all extensions in the backend
* Check tax setting
* Check transactional emails
* Check attributes and attribute sets

**Misc**

* Check if crontab is set, start cron.php manually and check cron\_schedule for errors
* Shell scripts
* After all previous test assignments, check all log files (Magento logs, PHP logs, MySQL logs, email logs)

**Installer/Uninstaller**

* Install ownCloud
  + expected: files to be found in System-wide app directory.

**Setup**

* Check if hostnames are accepted correctly by the input field
* Check if self-signed certificates are being accepted

**Syncing**

**User stop syncing**

(By pausing or logging out) Expected: Client stops syncing Error: Client crashes, Client stays in sync state (check with --logwindow

* Stop syncing while doing local discovery
* Stop syncing while doing remote discovery
* stop syncing while uploading a file
* stop syncing while downloading a file
* stop syncing while deleting files
* stop syncing while renaming files

**Test resuming of uploads**

* Add a big files locally (100Mb or more) and look at the server logs: it should start sending chunks
* Kill the client while syncing.
* Restart the client: it should resume from the last sent chunk, and stop when all the chunk are transfered (observe that on the server log)
* Do the same but instead of killing the client, shut the server off or turn off the network

**Test resuming download**

* Add a big file on the server
* Look at the progress bar of the sync and kill the client when it is almost over
* Restart the client
  + The sync should resume from what it already downloaded.
  + Known bug: the progress still show it has too download it full
  + But after the remaining megabytes have been downloaded, the progress bar should jump to the end.

**Modifying files when the sync is running**

* Add a new directories full of files and let it sync.
* While syncing, rename directory on the client.
* After two sync, the directory fust have been renamed on the server and must not re-appear on the client, and all the files should be there.
* Now do the same, but delete the directory instead of renaming
* Do the same, but rename the directory on the server when syncing
* The same but delete the directory on the server

**Test shared folder**

* Have a read only Shared/ folder
* Adding a file there should cause an error. The error should be blacklisted after a few sync.
* Editing a file should cause an error saying that a conflict file has been created and the original file restored. Verify it worked
* Removing a file should restore the file
* Renaming a file should restore the original, and show an error for the new one.
* Rename Shared/ to something else : Syncing should move Shared/ back
* Move a subfolder of Shared outside of Shared : The original files should be restores at their original location, and the new directory uploaded to the server
* Remove a subfolder : The original files hsould be restored

**Network-related**

* Network goes away
  + Expected: Client recovers
  + Error: Client crashes, client stays in error state
* Network goes online again:
  + Expected: the sync resumes

**Crash Recovery**

* client crashes (kill -9, force end via task manager)
  + Expected: client recovers
  + Error: client shows error message, DB corrupts, settings are forgotten
  + Important: check behavior also with clean settings directory. Does the client forget its credentials?

**Server interaction checks**

* Client must not initiate new session per request, make sure cookie gets passed along.
* Remove DB.
  + Expected results: no changes after initial sync.
  + Tolerated: some conflicts when server and client are not time-synced
  + Error: Conflict files get uploaded to server

**Upgrade (where applicable)**

**Upgrade from Previous Client Version**

For this test, the server stays the same.

On Windows: Install the previous version of the client. Configure some acconts and folder and other settings. Download the new installer and start it while the previous is running. It will ask if it should remove the older installation, test with both choices. Reboot after every update and see if the system start works fine. Check that the settings and folder configuration are kept from one version to the next.

**Upgrade from previous server version**

For this, the client version is the one that should be tested and stays the same. Install on older server version, sync. After that, update the server and let it perform its update business. The client needs to handle that gracefully.

**Credential Management**

The following test cases should be checked.

**First startup**

This is the situation new users have: There is no former installation with credentials stored somewhere.

* Check if the account has set credentials in the password storage which is Registry on Windows, Wallet system on Linux and the System Password storage on MacOS. Remove all ownCloud related entries there. Do that after ever test here.
* Test if the user can connect to a http server with correct credentials.
* Test if the user can connect to a https server with correct credentials.
* Test what happens if user connects with wrong user.
* Test what happens if user connects with wrong password.
* Test what happens if user presses "Cancel" without having connected.
* Verify that user can not press "Finish" before a successful connection was established.
* Verify that if we bo back and enter different URL everything works correctly

**Update**

Test authentication after the user updated from a previous version. Basically repeat the tests from before but with existing credentials in the credential store.

* Check if the client starts without asking for password on first start after update.

**Password change, without URL change.**

Test the case that the user wants to change her password. For that, run the configuration dialog and just change the users password. For that, first change the password on the server using the web interface.

* After the password change on the webinterface, check the client's reaction when it tries to connect with the old, now wrong credentials. Start the client with the wrong credentials.
* Change the credentials on the server while the client is running. Check the reaction.
* Test if the new password is accepted and appears in the credential store.
* Test if the client connects to the server and starts syncing after user pressed "Finish".
* Test if the old configuration remains unchanged after user pressed "Cancel".
* Test if the proxy settings are the same as before after reconfiguration.
* Go back in the configuration wizard and check if the password is still consistently used.
* Test with a wrong password.

**Password change, with URL change**

Repeat the tests above, but also with the url changing.

**Wrong password**

* Sing out. (from the menu in the systemtray)
* Sign in: It should ask for the password
* Press cancel: it should stay disconnected and not ask for the password
* Sign in, and enter the wrong password: it should ask for the password again
* Enter the wrong password a couple of times: it must always ask for the password
* Enter the right password: It should do the sync and connect and not ask for the password anymore.

**Shibboleth**

* Change the account to use an account on the shibboleth test server
* Connect and start a sync.
* Try sign in / sign out
* Try wrong password, try closing the shibboleth window without loggin in.
* Try loggin with a different user. This should not be allowed

**Multiples folders**

* Add multiples folder : Check that they sync properly
* Remove a folder : There should not be crashes.

**Activity log**

* Look at the activity log.
* Click or double click on items. It should open the explorer with the file selected.

**Quota**

* Check that the quota displayed on every account page is correct compared to what the server says it is.
* Change the quota on the server. Make sure that the quota is updated within 30 seconds on the quota indicator on the client. This need to be tested if the ui is visible when the change happen, or hidden when the change happen.
* After a sync that uploaded files, the quota should be updated within 30 seconds.

**Windows Office Files**

* Save an office file under windows into the sync dir. Keep the file open in the office app. Every time the file is saved in the office app, the file should be synced.

**Big File modified during Upsync**

* Put a big file into the sync dir. While it is synced up, change it. The client should realize the change, stop syncing and redo it later.

**Command Line Client**

* Check that no username/password is logged in cleartext during operation.

Bottom of Form

* [HOME](http://onecore.net/)

* [ABOUT](http://onecore.net/about)

* [ADVERTISE](http://onecore.net/advertise)

* [ARCHIVES](http://onecore.net/archives)

* [PRIVACY POLICY](http://onecore.net/privacy-policy)