Tiki 3

Menu

Home
Search
Categories
📁 **Wiki**
Wiki Home
Last Changes
Rankings
List Pages
Orphan Pages
Sandbox
📁 **File Galleries**
List Galleries
Upload File
📁 **Admin**
Admin Home
Admin Mods
Cache
Categories
Comments
Content templates
Cookies
DSN
External wikis
Groups
Hotwords
Links
Mail Notifications
Menus
Modules
QuickTags
RSS Modules
Security Admin
Tiki Cache/Sys Admin
Tiki Logs
Users
phpinfo

Search

in:

Entire Site

Advanced search: ☑

✏️ Backlinks…

Login

Help

Logged in as: mchung⭐

Logout

Switch User

User: [   ]

Switch

Recently visited pages

1. Functional Testing with GUI for QA
2. Functional Testing
3. Testing
4. UserPagemchung
5. HomePage

Top users

1. 12086 mchung⭐
2. 4771 kchung⭐
3. 4764 aau⭐
4. 4097 tcheung⭐
5. 3981 ckwong⭐
6. 3439 dlam⭐
7. 2434 aleung⭐
8. 2368 wlo⭐
9. 1680 tyip⭐
10. 1249 kfung⭐

Last changes

1. Clients
2. Internships
3. SunTrust Credit Card
4. Company Credit Cards
5. Answering MSP Tickets
6. Markdown
7. UserPagemchung
8. Java Code Review Checklist
9. Code Review

Go

Quick Edit a Wiki Page

Create/Edit

Tikiwiki Assistant

Thank you for installing Tikiwiki!

▶ **To configure your Tiki**:
Select Admin > Admin Home from the menu.

Read the configuration documentation.

Watch the demo movies.

▶ **To remove this module**:
Select Admin > Modules and remove the assistant module. You can also add other modules.

▶ **To customize the menu**:
Select Admin > Menus and edit menu ID 42.
Or, create your own menu and add it to a module.

To learn more, visit:
http://tikiwiki.org.

For help, visit
http://doc.tikiwiki.org.

10. Develop a Proposal Theme

...more   **Help**

Functional Testing with GUI for QA

All QA should use this checklist for the screen. Regression testing may not need to cover everything, depending on the changes.
Prerequisites

- Ask the developer and analyst to show you what is the change and how to use the system with changes.
- Ask for use case document if any.
- Make sure there is a SharePoint test request before you start testing.

Security Related

1. Make sure that all Personal Identifiable Information (PII) in the database is encrypted such as SSN, birthday, and credit card number
2. After we use the testing program for a while, do a javascript:alert(document.cookie) in a web browser and find out what values are stored in the cookie.
3. We want to make sure that the logout button will terminate the session. One way to test it is to copy the URL after login, click the logout button, and paste the URL and see if it still can load the previous protected page. We may test it by using the back button after logout.

Functional Testing Related

1. Enter valid values for all required fields, save it, re-open the screen and see if it retrieves the same values back.
2. Click all buttons and links.
3. Make sure you understand what is the meaning of each field and message on the screen and if they make sense. Ask the analyst if you have questions.
4. Check Access Control. There are different types of users in the system. Some users can edit. Some users can only view. Some users can access all the web pages and functionality. Other users can't. Login as different users and see if they can access what they are supposed to access. **If regular users know the URL of a privileged page, make sure the system doesn't allow them to access the page**.
5. For screens that allow you to insert data, normally certain fields are unique fields, such as username, ssn, different types of ID. Try to insert data that has the same unique field as existing data. The system is supposed to return error messages.
6. Check grammar and spelling.

Data Validation

Help

1. If couples values have group restriction, check if individual will violates the restriction. For example, a + b + c = 100%, can "a" = 30%, "b" = 80% and "c" = -10% be accepted?
2. Ask the analyst if there are valid data range for an input field. It is better if this info is in a document. Enter the first and last value in valid range of input field. For example, if you have 10,000 users, retrieve the first one and last one.
3. Enter valid values for all required fields except one at a time, save it, and see if the system warns you.
4. Enter incorrect values for each field one or more at a time, and see if the system warns you.Invalid values means invalid data type, valid data type with invalid range, valid data type with invalid length, or same data type with invalid context (such as 2/30/2008).
5. for numeric fields or string field that accepts number, such as phone numbers or zip codes, try to enter leading zeros.
6. identify fields that are not supposed to have spaces in the prefix or suffix, and then see if the system would complain or save it. For example, " mchung " is not supposed to be a valid username because of the space in front and at the end. The system either strips off these spaces automatically or returns an error message. I am sure there are other fields like this besides usernames.
7. Test if you can delete data that is used by other forms
8. Test if the system returns an error message for any blank required fields.
9. Validation not needed for a blank, non-required field.
10. Some fields are required only after a checkbox is checked. For example, "Others" text input field is needed only if the "Others" checkbox is checked. Validation is needed if the checkbox is checked and not needed if the checkbox is not checked.
11. If there is a textbox, enter as many lines of characters as possible and save it. See if the system can handle that.
12. If there is a list with "add" button, add as many items as possible, and save it. See if the system can handle that. Ask the developer whether this list is used in a report or another page. Use that page or report to view the list if there is any layout problems.
13. If there is an input field, type as many characters as possible and see if the system breaks when you try to save it.

GUI Programing Related

1. Please make sure all debug messages are gone
2. Turn off cookies in the browser for the domain name of the system. Test and see if the system still works. If the system doesn't work, either ask the developer to make it work with cookies turned off or the system should display the message that "cookies must be turned on" if the browser turned

Help

off cookies.

3. Turn off scripts for the domain name of the system. Test and see if the system still works. If the system doesn't work, either ask the developer to make it work with scripts turned off or the system should display the message that "scripts must be turned on" if the browser turned off scripts. We can install firefox with the "no scripts" plugin to turn off the scripts.

4. For each field in a form, click the field, enter some valid values, and then hit the "Enter" key on the keyboard (instead of the submit button). Check if the form behaves as if the submit button is clicked, including the result after submitting the form.

5. For string field, ask the analyst and developer whether the field is case sensitive. Try a combination of both uppercase and lowercase values and see if the system returns exceptions. For example, we usually convert email address and username into lowercase.

6. After the user navigates from the current page by clicking any buttons or links, if the user navigates back to this page, the system should save the original settings. For example, if the user sets a search criteria and clicks search, then views a detailed record, and if the user clicks the back button in the browser or cancel in the application to go back to the previous page, the system should save the search criteria.

7. Check to see if the system can handle unhandled exception gracefully. Normally the system should display something like "The system encounters an unhandled exception error. You can click here and email the error log to the system administrator". The system should automatically send the administrator an error log too.

GUI Security Related

1. Enter values with sql symbols to prevent the system from sql injection attack, for example use =, <, >, ', ", and #.

2. Ask the developer to show you a system log or database log that indicated the time you login and logout the system. This includes logout by session timeout.

3. Ask the developers which tables and actions are being audited. Then ask the developers to check whether those actions are actually being audited after you perform those actions.

4. Login must use https

5. Page error must not show Technical Information or Debug Information, debug mode must be turned off.

GUI Design

1. Must have a title and a header.

2. The Title, Header, and Breadcrumbs should match unless the developer has a good reason for the mismatch. Otherwise, website users will find it

Help

confusing.

3. Must have a brief instruction & maybe a purpose to the page.
4. Website layout should be able to re-size properly. Our standard is pretty low. As long as the screen doesn't look like it screws up, then the page passes this resize test.
5. Check if we can remove blank lines so that users do not have to scroll to click a button at the bottom. Also check if we can combine two lines into one line.
6. Check sorting by whether a list is sorted properly and if the page has the same set of records
7. Check if the site has a consistent web page style, such as color & fonts.
8. Check the labels to make sure that all required fields have an (*)
9. It is good to have the footer at the same position on pages if the content can fit into one page.
10. All lists or tables should be sorted.
11. Every time users click a button, there should be a way to notify the users the data was saved successfully, Users either see the new data on the screen, or a popped up window was closed, or the system display a message on the screen.
12. Test the system using IE 7, latest Firefox, and IE 8.
13. Make sure there is a logout button.
14. All field labels must be right aligned with ":".
15. The font for labels must be bigger or the same size as the input field
16. The field label and the field should align to the middle vertically
17. Input field should not have more than one padding on the left or on the right
18. View the source of different web pages and see if there are comments or info about the software used. For example, the source of www.wasoftware.com may contain references to Joomla.
19. Make sure the meta tags in keyword and description describes the website.
20. View source of different web pages and see if the meta tags "keywords", "description", and "title" are setup properly.
21. Make sure favicon is either empty or show the website's favicon instead of the plesk favicon.

Section 508 Related

1. Make sure "Skip Navigation" is in "View Source" if the site is supposed to be 508 compliant.
2. Make sure all images have description in ALT tags.
3. Download this document for detail Section 508 Solicitation Documentation for detail.

Configuration Related

1. Configure web server so that it restricts URL access based on user access control
2. Use TLS in login page

3. For security sake, make sure that there are no use of redirects and forwards.
   If it's used, try a mapping instead of a real URL.
4. In the source code, the program needs to authenticate each page that the
   user is allowed to view that page.

Help

| Edit | Source | Remove | Rename | Perms | History | Similar |

Wiki File Galleries

Powered by TikiWiki CMS/Groupware | Theme: Coelesce

[Execution time: 0.21 secs]   [Memory usage: 11.22MB]   [28 database queries used in 0.0 secs]   [Server load: 0.01]