

TEST CASE Example

Administration

Owner:
Scope: To validate known functions of the Administration features.
Originator:
Status: Active
Document ID: TestCase_Administration.doc
Location:

Change History

Issue	Date	Handled by	Comments
Created.	dd-mmm-yyyy		

Approved by

TypeDateHere

TypeNameHere

Table of Contents

1.	Introduction	4
2.	Scope.....	4
3.	Requirements.....	4
4.	Related Documents.....	4
5.	Traceability Matrix	4
6.	Notes.....	5
7.	Test Cases - Administration.....	5
7.1	Test Item: <i>Registration Request Form</i>	5
7.2	Test Item: <i>Admin Login</i>	9
7.3	Test Item: <i>Create User</i>	10
7.4	Test Item: <i>Edit User</i>	14
7.5	Test Item: <i>Basic Delete and Disable</i>	17
7.6	Test Item: <i>Test user</i>	19
7.7	Test Item: <i>Admin creates user(s) and assign(s) to group(s) using batch scripts</i>	19
7.8	Test Item: <i>Delete and Disable users logged into Company</i>	20
7.9	Test Item: <i>Email to All Users</i>	28
7.10	Test Item: <i>Logout</i>	31
7.11	Test Item: <i>Multiple browser window usage</i>	31
7.12	Test Item: <i>Change Admin password</i>	32
7.13	Test Item: <i>SRC user list</i>	Error! Bookmark not defined.
7.14	Test Item: <i>Sending log files to Company Support</i>	33
7.15	Test Item: <i>Refresh after submits</i>	36
7.16	Test Item: <i>User/Admin Level Privileges</i>	36
7.17	Test Item: <i>Session validation</i>	37
7.18	Test Item: <i>Provision</i>	Error! Bookmark not defined.
7.19	Test Item: <i>Client Admin Function</i>	Error! Bookmark not defined.

1. INTRODUCTION

The purpose of this test case is to test and validate the functionality of the Company Administration features.

2. SCOPE

To validate all known functions of the Company administration features.

3. REQUIREMENTS

Discuss any derived requirements and/or requirements that need clarification or are necessary for completion prior to the execution of the test cases listed in this document.

For example:

The input criteria necessary for the Golden Master CD testing includes:

- Satisfactory completion of the agreed upon QA testing requirements – including project relevant test plans, procedures, and test cases.
- Creation of Golden Master CD from the Software Maintenance Group.
- Software Release Matrix as delivered from the Product Marketing and Software Maintenance Groups.

4. RELATED DOCUMENTS

Add a list and links to supporting documents here, e.g., RFCs, other design documents, standards specifications, etc...

5. TRACEABILITY MATRIX

The following table describes the relation mapping between requirements from the requirement specification documents (Company Extensions Design.v02, UserAdminDesign.v04.doc) and corresponding test items, which validate their implementation

Requirement	Validated by:
# Req-1	Test Item xxx
# Req-2	
# Req-3	
# Req-4	
# PRV-1	Test Item xxx
Bug #1234	Test xxx.xxx
Bug #1235	Test xxx.xxx
Bug #1236	Test xxx.xxx
Bug #1237	Test xxx.xxx
# REQ5	Test Items from xx.xx to xx.xx
# REQ6	
# REQ7	
# REQ8	

# REQ9	
# REQ10	

6. NOTES

Add any generally relevant notes here.

7. TEST CASES - Administration

This section should provide the set of test cases that are used in the process of testing specific features and functions. Include the following checklist.

Follow the checklist in order and perform each test. Check the Pass work Fail box to the left according to the results. Note bugzilla number if/when a bug is submitted resulting from a failed test

Test Run #	Tester Name	Date	Browser & Version	Execution Time	Did all the Tests Pass?

7.1 **Test Item:** *Registration Request Form.*

Scope: *Test various field input on the Registration Request Form.*

Action: *Submit Registration request.*

Test Notes and Preconditions: *Open the user registration request form from the users tab.*

Verification Steps: *Verify the user has been registered with the given details.*

Scenario 1: *Test leaving a required field blank on the new user registration*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.1.1	Submit Registration request...	First Name = blank, Other fields valid	Error message should be displayed for invalid First Name.			
7.1.2	Submit Registration request...	Last Name = blank, Other fields valid	Error message should be displayed for invalid Last Name.			
7.1.3	Submit Registration request...	Email = blank, Other fields valid	Error message should be displayed for invalid Email.			
7.1.4	Submit Registration request...	Phone number = blank, Other fields valid	Error message should be displayed for invalid Phone number.			
7.1.5	Submit Registration request...	Company = blank, Other fields valid	Error message should be displayed for invalid Last Name.			
7.1.7	Submit Registration request...	User ID = blank, Other fields valid	Error message should be displayed for invalid User ID.			
7.1.8	Submit Registration request...	Password = blank, Other fields valid	Error message should be displayed for invalid Password.			
7.1.9	Submit Registration request...	Confirm Password = blank	Error message should be displayed for invalid			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
		Other fields valid	confirmation password			
7.1.10	Submit Registration request...	All fields blank	Error message should be displayed.			

Scope: Test various field input on the Registration Request Form.

Scenario 2: Test minimum number of characters accepted for each field.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.1.11	Submit Registration request...	First Name = max chars Last Name = max chars Email = max chars incl @ Phone # = max chars Company = max chars Reason = max chars User ID = max chars Password = max chars Confirm pwd = max chars	Registration request should be accepted.			
7.1.12	Submit Registration request... (Bug#10224)	First Name = max chars Last Name = max chars Email = max chars incl @ Phone # = max chars Company = max chars Additional information for the admin =Max Chars User ID = max chars Password = max chars Confirm pwd = max chars	Registration request should be accepted.			
7.1.12	Login as Admin and enable this user...	None	User is enabled.			
7.1.13	Login as the new user...	None	User successfully logs in.			
7.1.14	Login as Admin and delete this user...	None	User is deleted.			

Scope: Test various field input on the Registration Request Form.

Scenario 3: Test maximum number of characters accepted for each field.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.1.15	Submit Registration request...	First Name = 1 char Last Name = 1 char Email = a@a.com Phone # = 1 char Company = 1 char Additional information for the admin = 1 char User ID = 1 char Password = 6 chars Confirm pwd = 6 chars	Registration request should be accepted.			
7.1.16	Login as Admin and enable this user...	None	User is enabled.			
7.1.17	Login as the new user...	None	User successfully logs in.			
7.1.18	Login as Admin and delete this user...	None	User is deleted.			

Scope: Test various field input on the Registration Request Form.

Test Notes & Preconditions: Field input should limit the user to some maximum value. If not, an error should be produced.

Scenario 4: Test exceeding maximum field input lengths on the new user registration form.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.1.19	Submit Registration request...	First Name = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.20	Submit Registration request...	Last Name = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.21	Submit Registration request...	Email = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.22	Submit Registration request...	Phone number = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.23	Submit Registration request...	Company = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.24	Submit Registration request...	Additional information for the admin = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.25	Submit Registration request...	User ID = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.26	Submit Registration request...	Password = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.1.27	Submit Registration request...	Confirm Password = max + 1 chars Other fields valid	Field input is limited to max allowable chars.			

Scope: Test various field input on the Registration Request Form.

Scenario 5: Test special characters on the new user registration form. This test is for all fields except Email and User ID, which are addressed separately.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.1.28	Submit Registration request...	Enter all special chars in each field, i.e. `~! @#%&*()-=_+[]{} ;'"<>/? Except for Email and User ID, which should be some valid entry for this test.	Registration request should be accepted.			
7.1.29	Submit Registration request...	Enter pound char in each field, i.e. Asd##ads32%^&*(Except for Email and User ID, which should be some valid entry for this test.	Registration request should be accepted.		Bug #1234567	
7.1.30	Login as Admin and enable this user...	None	User is enabled.			
7.1.31	Login as the new user...	None	User successfully logs in.			
7.1.32	Login as Admin and delete	None	User is deleted.			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	this user...					

Scope: Test various field input on the Registration Request Form.

Scenario 6: Test Email field input.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.1.33	Submit Registration request...	Email = a	Appropriate error should be displayed saying the email is invalid.			
7.1.34	Submit Registration request...	Email = a.b	Appropriate error should be displayed saying the email is invalid.			
7.1.35	Submit Registration request...	Email = a@b	Appropriate error should be displayed saying the email is invalid.			
7.1.36	Submit Registration request...	Email = a@b.c	Email address should be accepted.			
7.1.37	Submit Registration request...	Email = a@b.com	Appropriate error should be displayed saying the email is invalid.			
7.1.38	Submit Registration request...	Email = a @b.com	Email address should be accepted.			
7.1.39	Submit Registration request...	Email = a@.com	Appropriate error should be displayed saying the email is invalid.			
7.1.40	Submit Registration request...	Email = a_b@c.com	Appropriate error should be displayed saying the email is invalid.			
7.1.41	Submit Registration request...	Email = a b@c.com	Email address should be accepted.			
7.1.42	Submit Registration request...	Email = a b @c.com	Email address should be accepted.			
7.1.43	Submit Registration request...	Email = a@b@c.com	Appropriate error should be displayed saying the email is invalid.			
7.1.44	Submit Registration request...	Email = a`~!@#%&*()-=+_[]{} ;':",.<>/?	Appropriate error should be displayed saying the email is invalid.			
7.1.45	Submit Registration request... (Bug #6855)	Email = a`##~!\$%^&*()-=+_[]{} ;':",.<>/?b@c.com	Appropriate error should be displayed saying the email is invalid.			
7.1.46	Submit Registration request...	Email = a`~!#%&*()-=+_[]{} ;':",.<>/?b@c.com	Appropriate error should be displayed saying the email is invalid.			

Scope: Test various field input on the Registration Request Form.

Test Notes & Preconditions: The User ID can only contain numbers and letters.

Scenario 7: Test User ID field input.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.1.47	Submit Registration request...	User ID = all letters	Input should be accepted.			
7.1.48	Submit Registration request...	User ID = all numbers	Input should be accepted.			
7.1.49	Submit Registration request...	User ID = alpha-numeric mix	Input should be accepted.			
7.1.50	Submit Registration request...	User ID = contains a space char	Input should be accepted.			
7.1.51	Submit Registration request...	User ID = contains a period.	Input should be accepted (user id must be begin with an alpha-numeric char).			
7.1.52	Submit Registration request...	User ID = contains a dash -	Input should be accepted (user id must be begin with an alpha			
7.1.53	Submit Registration request...	User ID = contains a plus char +	Input should be accepted (user id must be begin with an alpha			
7.1.54	Submit Registration request...	User ID = contains a slash char /	Input should be accepted (user id must be begin with an alpha			
7.1.55	Submit Registration request...	User ID = contains a pound char twice i.e. ##	Input should be accepted.			
7.1.56	Submit Registration request...	User ID = Duplicate of an already registered ID	Appropriate error message should be displayed.			
7.1.57	Submit Registration request...	User ID = unique alphanumeric entry but all other fields are duplicate of a registered user.	Input should be accepted.			
7.1.58	Submit Registration request... (Bug 10224)	Additional information for the admin = Less than 256 character	Registration request should be accepted. The requested user will get an alert mail from N1bs Admin			
7.1.59	Submit Registration request... (Bug 10224)	Additional information for the admin: 256 character	Registration request should be accepted. The requested user will get an alert mail from N1bs Admin			
7.1.60	Submit Registration request...(Bug 10224)	Additional information for the admin: = More than 256 character	Registration request should be accepted. The requested user will get an alert mail from N1bs Admin			
7.1.61	Submit Registration request...(Bug 10224)	Additional information for the admin: = Maximum characters	Registration request should be accepted. The requested user will get an alert mail from N1bs Admin			

Scenario 8: Password field tests are exercised in the TestCase_TPAS.doc test case document.

7.2 Test Item: Admin Login

Scope: Test valid and invalid login attempts to the Administration page.

Action: User submits Login form.

Test Notes and Preconditions: *Open the admin login page.*

Verification Steps: *Verify the user has been entered in to the admin page after given the correct details.*

Scenario 1: *Test User ID field input.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.2.1	Admin User submits Login form (Bug#10455)	User Name=Invalid Password=Invalid	It should display an error message saying "The User ID or Password you entered is invalid. Please <u>try again</u> "			
7.2.2	Click on 'try again' hyper link (Bug # 10455)		SRC admin login page should be displayed			
7.2.3	Admin User submits Login form (Bug#10455)	User Name=valid Password=Invalid	It should display an error message saying "The User ID or Password you entered is invalid. Please <u>try again</u> "			
7.2.4	Click on 'try again' hyper link (Bug # 10455)		SRC admin login page should be displayed			
7.2.5	Admin User submits Login form (Bug#10455)	User Name=Invalid Password=valid	It should display an error message saying "The User ID or Password you entered is invalid. Please <u>try again</u> "			
7.2.6	Click on 'try again' hyper link (Bug # 10455)		SRC admin login page should be displayed			
7.2.7	Admin User submits Login form	Password = Valid User Name=valid	Admin user should be logged in successfully in the admin page			

7.3 Test Item: *Create User.*

Scope: *Test various field input on the Create User Form.*

Action: *Submit Create User form.*

Test Notes and Preconditions: *Open the User registration form from the users tab.*

Verification Steps: *Verify the user has been registered with the given details.*

Scenario 1: *Test various field input on the Create User Form*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.3.1	Submit Create User form...	First Name = blank, Other fields valid	Error message should be displayed for invalid First Name.			
7.3.2	Submit Create User form...	Last Name = blank, Other fields valid	Error message should be displayed for invalid Last Name.			
7.3.3	Submit Create User form...	Email = blank, Other fields valid	Error message should be displayed for invalid Email.			
7.3.4	Submit Create User form...	User ID = blank, Other fields valid	Error message should be displayed for invalid User ID.			
7.3.5	Submit Create User	Password = blank,	Error message should			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	form...	Other fields valid	be displayed for invalid Password.			
7.3.6	Submit Create User form...	Confirm Password = blank Other fields valid	Error message should be displayed for invalid confirmation password			
7.3.7	Submit Create User form...	All fields blank	Error message should be displayed.			
	Submit Create User form...	Client Version = Standard Client	User successfully Ceated.			
	Submit Create User form...	Client Version = Professional client	User successfully created.			
7.3.8 (Bug 8147 and 10475)	Creating new user account	First Name = valid Last Name = valid Email = valid User ID = valid Password = valid Confirm pwd = valid chars	User successfully created.			

Scope: Test various field input on the Create User Form.

Scenario 2: Test minimum number of characters accepted for each field.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.3.8	Submit Create User form...	First Name = 1 char Last Name = 1 char Email = a@a User ID = 1 char Password = 6 chars Confirm pwd = 6 chars	Create User form should be accepted and the user is created.			
7.3.9	Login as the new user...	None	User successfully logs in.			
7.3.10	Login as Admin and delete this user...	None	User is deleted.			

Scope: Test various field input on the Create User Form.

Scenario 3: Test maximum number of characters accepted for each field.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.3.11	Submit Create User form...	First Name = max chars Last Name = max chars Email = max chars incl @ User ID = max chars Password = max chars Confirm pwd = max chars	Create User form should be accepted and the user is created.			
7.3.12	Login as the new user...	None	User successfully logs in.			
7.3.13	Login as Admin and delete this user...	None	User is deleted.			

Scope: Test various field input on the Create User Form.

Test Notes & Preconditions: Field input should limit the user to some maximum value. If not, an error should be produced

Scenario 4: Test exceeding maximum field input lengths on the create user form.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.3.1 4	Submit Create User form...	First Name = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.3.1 5	Submit Create User form...	Last Name = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.3.1 6	Submit Create User form...	Email = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.3.1 7	Submit Create User form...	User ID = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.3.1 8	Submit Create User form...	Password = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.3.1 9	Submit Create User form...	Confirm Password = max + 1 chars Other fields valid	Field input is limited to max allowable chars.			

Scope: Test various field input on the Create User Form.

Scenario 5: Test special characters on create user form. This test is for all fields except Email and User ID, which are addressed separately.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.3.2 0	Submit Create User form...	Enter all special chars in each field, i.e. `~!@#\$\$%^&*()- =+_[]{} ;':",.<>/? Except for Email and User ID, which should be some valid entry for this test.	Create User form should be accepted and the user is created.			
7.3.2 1	Submit Create User form...	Enter pound char twice in each field, i.e. `~!@#\$\$%^&*()- =+_[]{} ;':",.<>/? Except for Email and User ID, which should be some valid entry for this test.	Create User form should be accepted and the user is created.		Bug#4567	
7.3.2 2	Login as the new user...	None	User successfully logs in.			
7.3.2 3	Login as Admin and delete this user...	None	User is deleted.			

Scope: Test various field input on the Create User Form.

Test Notes & Preconditions: Not certain at this time which of these tests will produce errors. The mechanism for email validation hasn't been specified or documented to my knowledge. Need to get more info from the developer and may need to expand this test scenario. When the email address does not actually exist, the server will write an error to the error_log since it can't actually send a confirmation email to that address.

Scenario 6: Test Email field input.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.3.24	Submit Create User form...	Email = a	Appropriate error message should be displayed.			
7.3.25	Submit Create User form...	Email = a.b	Appropriate error message should be displayed.			
7.3.26	Submit Create User form...	Email = a@b	Appropriate error message should be displayed			
7.3.27	Submit Create User form...	Email = a@b.c	Address should be accepted.			
7.3.28	Submit Create User form...	Email = a@b.com	Appropriate error message should be displayed			
7.3.29	Submit Create User form...	Email = a @b.com	Appropriate error message should be displayed. .			
7.3.30	Submit Create User form...	Email = a@.com	Appropriate error message should be displayed. .			
7.3.31	Submit Create User form...	Email = a b@c.com	Address should be accepted.			
7.3.32	Submit Create User form...	Email = a b@c.com	Appropriate error message should be displayed			
7.3.33	Submit Create User form...	Email = a b @c.com	Appropriate error message should be displayed.			
7.3.34	Submit Create User form...	Email = a@b@c.com	Appropriate error message should be displayed			
7.3.35	Submit Create User form...	Email = a`~!@#%&^&*()-=+_[]{} ;':",.<>/?	Appropriate error message should be displayed.			
7.3.36	Submit Create User form...	Email = a`~!#\$%^&*()-=+_[]{} ;':",.<>/ ?b@c.com	Appropriate error message should be displayed			
7.3.37	Submit Create User form...	Email = wqe##asdf@a23sdsd.com	Address should be accepted.		Bug# 789	

Scope: Test various field input on the Create User Form.

Test Notes & Preconditions: The User ID can only contain numbers and letters.

Scenario 7: Test User ID field input.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.3.38	Submit Create User form...	User ID = all letters	Input should be accepted.			
7.3.39	Submit Create User form...	User ID = all numbers	Input should be accepted.			
7.3.40	Submit Create User form...	User ID = alpha-numeric mix	Input should be accepted.			
7.3.41	Submit Create User form...	User ID = contains a space char	Input should be accepted (user id must begin with an alpha-numeric char)			
7.3.42	Submit Create User form...	User ID = contains a period.	Input should be accepted (user id must begin with an alpha-numeric char)			
7.3.43	Submit Create User	User ID = contains a dash	Input should be			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	form...	-	accepted (user id must begin with an alpha-numeric char)			
7.3.44	Submit Create User form...	User ID = contains a plus char +	Input should be accepted (user id must begin with an alpha-numeric char)			
7.3.45	Submit Create User form...	User ID = contains a slash char /	Input should be accepted (user id must begin with an alpha-numeric char)			
7.3.46	Submit Create User form...	User ID = Duplicate of an already registered ID	Appropriate error message should be displayed.			
7.3.47	Submit Create User form...	User ID = unique alpha-numeric entry but all other fields are duplicate of a registered user.	Input should be accepted.			
7.3.48	1 .Login to secure shell as admin 2. Add the following line to decuma.local.properties. license_source=@butler 3. Add users 4. Comment out the line from decuma.local.properties #license_source=@butler (bug #11084)		There are x no of users registered on this server, out of a maximum of 0.			

7.4 Test Item: Edit User

Scope: Test various field input on the Edit User Form.

Action: Display Edit User form.

Test Notes and Preconditions: Open the user registration request form from users tab from the existing user list.

Verification Steps: verify the user details have been updated with the given details.

Scenario 1: Test that User ID field is NOT editable on the edit user form.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.1	Display Edit User form...	Attempt to edit User ID field.	User ID field is NOT editable.	(userid field is plain text, not a text input)		

Scope: Test various field input on the Edit User Form.

Scenario 2: Test for empty password filed in the edit user form.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.2	Display Edit User form...	Verify the password field value.	The password field should be empty.			

Scope: Test various field input on the Edit User Form.

Scenario 3: Test leaving a required field blank on the edit user form.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.3	Submit Edit User form...	First Name = blank, Other fields valid	Error message should be displayed for invalid First Name.			
7.4.4	Submit Edit User form...	Last Name = blank, Other fields valid	Error message should be displayed for invalid Last Name.			
7.4.5	Submit Edit User form...	Email = blank, Other fields valid	Error message should be displayed for invalid Email.			
7.4.6	Submit Edit User form...	Password = blank, Other fields valid	Error message should be displayed for invalid Password.			
7.4.7	Submit Edit User form...	Confirm Password = blank Other fields valid	Error message should be displayed for invalid confirmation password			
7.4.8	Submit Edit User form...	All fields blank	Error message should be displayed.			

Scope: Test various field input on the Edit User Form.

Scenario 4: Test minimum number of characters accepted for each field.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.9	Submit Edit User form...	First Name = 1 char Last Name = 1 char Email = a@a Password = 6 chars Confirm pwd = 6 chars	Edit User form should be accepted and the user information is updated.			
7.4.10	Login as the new user...	None	User successfully logs in.			

Scope: Test various field input on the Edit User Form.

Scenario 5: Test maximum number of characters accepted for each field.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.11	Submit Edit User form...	First Name = max chars Last Name = max chars Email = max chars incl @ Password = max chars Confirm pwd = max chars	Edit User form should be accepted and the user info is updated.			
7.4.12	Login as the new user...	None	User successfully logs in.			

Scope: Test various field input on the Edit User Form.

Test Notes & Preconditions: *Field input should limit the user to some maximum value. If not, an error should be produced.*

Scenario 6: *Test exceeding maximum field input lengths on the Edit user form.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.13	Submit Edit User form...	First Name = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.4.14	Submit Edit User form...	Last Name = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.4.15	Submit Edit User form...	Email = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.4.16	Submit Edit User form...	Password = max + 1 chars, Other fields valid	Field input is limited to max allowable chars.			
7.4.17	Submit Edit User form...	Confirm Password = max + 1 chars Other fields valid	Field input is limited to max allowable chars.			

Scope: *Test various field input on the Edit User Form.*

Scenario 7: *Test special characters on Edit user form. This test is for all fields except Email and User ID which are addressed separately.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.18	Submit Edit User form...	Enter all special chars in each field, i.e. `~!@#%\$%^&*()- =_+[]{} ;':",.<>/? Except for Email which should be some valid entry for this test.	Edit User form should be accepted and the user information is updated.			
7.4.19	Login as the new user...	None	User successfully logs in.			

Scope: *Test various field input on the Edit User Form.*

Test Notes & Preconditions: *Not certain at this time which of these tests will produce errors. The mechanism for email validation hasn't been specified or documented to my knowledge. Need to get more info from the developer and may need to expand this test scenario.*

Scenario 8: *Test Email field input.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.4.20	Submit Edit User form...	Email = a	Appropriate error message should be displayed.	Allows it. Probably shouldn't but is innocuous		
7.4.21	Submit Edit User form...	Email = a.b	Appropriate error message should be displayed.			
7.4.22	Submit Edit User form...	Email = a@b	Appropriate error message should be displayed.			
7.4.23	Submit Edit User form...	Email = a@b.c	Address should be accepted.			
7.4.24	Submit Edit User form...	Email = a@b.com	Appropriate error message should be displayed saying the email is invalid.			
7.4.25	Submit Edit User form...	Email = a @b.com	Appropriate error message should be			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
			displayed saying the email is invalid			
7.4.26	Submit Edit User form...	Email = a@.com	Appropriate error message should be displayed saying the email is invalid.			
7.4.27	Submit Edit User form...	Email = a b@c.com	Address should be accepted.			
7.4.28	Submit Edit User form...	Email = a b@c.com	Appropriate error message should be displayed saying the email is invalid			
7.4.29	Submit Edit User form...	Email = a b @c.com	Appropriate error message should be displayed saying the email is invalid	Says it appears to be invalid		
7.4.30	Submit Edit User form...	Email = a@b@c.com	Appropriate error message should be displayed saying the email is invalid	Says it appears to be invalid		
7.4.31	Submit Edit User form...	Email = a`~!@#\$\$%^&*()-=+_[]{} \;':",.<>/?	Appropriate error message should be displayed saying the email is invalid.			
7.4.32	Submit Edit User form...	Email = a`~!@#\$\$%^&*()-=+_[]{} \;':",.<>/ ?b@.com	Appropriate error message should be displayed saying the email is invalid			
7.4.33	Submit Edit User form...	Email = a##dsfa23@asdfb.com	Address should be accepted.		Bug# 9876	

Scope: Test various field input on the Edit User Form.

Scenario 8: Password field tests are exercised in the TestCase_pass.doc test case document.

7.5 Test Item: Basic Delete and Disable

Scope: Test for login credentials of the Disabled user.

Action: Admin disables test user.

Test Notes & Preconditions: An active test user is available for this test and is not currently logged in.

Verification Steps: Verify the user has been disabled or not.

Scenario 1: Test that a disabled user cannot login to Company Access Mobilizer.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.5.1	Admin disables test user...	None.	User's status is changed from 'Active' to 'Disabled'.			
7.5.2	Disabled user attempts to login...	Valid user id and password provided at login	A message should be displayed that the user has been disabled and is not logged in.		Bug# 1234567 89	

Scope: Test for login credentials of the Deleted user.

Test Notes & Preconditions: *An active test user is available for this test and is not currently logged in.*

Scenario 2: *Test that a deleted user cannot login to Company Access Mobilzer.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.5.3	Admin deletes test user...	None.	User is deleted and removed from the registered list of users.			
7,5.4	Deleted user attempts to login...	Original user id and password provided at login	A message should be displayed that the User ID is invalid.			

Scope: *Test for login credentials of the Deleted disabled user.*

- Test Notes & Preconditions: *The user should be a member of a group and the 'Disabled' test user is available for this test.*

Setup: *The user should be a member of a group for this deletion test*

Scenario 3: *Test deleting a disabled user.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.5.5	Admin deletes the disabled test user...	None.	User is deleted and removed from the registered list of users.			
7.5.6	Deleted user attempts to login...	Original user id and password provided at login	A message should be displayed that the User ID is invalid.			

Scope: *Test for login credentials of the Deleted and recreated the same user*

Test Notes & Preconditions: *The user under test has at least one of each of the following:*

- *Registered device(s)*
- *Registered Bookmark(s)*
- *Registered Email account(s)*
- *Registered Sent-Mail folder*
- *Registered Speed message(s)*
- *Registered Nickname(s)*
- *Registered PIM account(s)*

Scenario 4: *Test deleting and recreating a user that has personalized configuration info in the Company database.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.5.7	Admin deletes the test user...	None.	User is deleted and removed from the registered list of users.			
7.5.8	Admin recreates a user with the same user information...	None.	The user is created successfully.			
7.5.9	The user logs into Company...	None.	The user is logged in. There are no			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
			configuration remnants left over from the previously deleted user with the same ID and user info.			

Scope: *Test for login credentials of the Deleted user.*

Test Notes & Preconditions: *The user under test has just been created and has performed NO personalized configuration.*

Scenario 5: *Test deleting and recreating a user that has NO personalized configuration info in the Company database.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.5.10	Admin deletes the test user...	None.	User is deleted and removed from the registered list of users.			

7.6 Test Item: Test user

Scope: *Test for test the user Account*

Action: *Admin test user.*

Test Notes & Preconditions: *Open the Test link from list all users tab*

Verification Steps: *Verify the user Configured server Account validation*

Scenario 1: *Test that a Test link for the particular user*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.6.1	Click on Test link for the particular user	User has not created any server Account	Message should display "User has no server account configured"			
7.6.2	Click on Test link for the particular user	User has Valid Server Account	Message should displays" Test successful! "			
7.6.3	Click on Test link for the particular user	User has invalid Server Account	Appropriate error message should display			

7.7 Test Item: Admin creates user(s) and assign(s) to group(s) using batch scripts

Scope: *administrator would use a script to do batch user(s) to group(s) assignments*

Test Notes & Preconditions: *A text file contains users and groups mapping data is existed for this test.*

Action: *Admin runs the batch user creation scripts*

Verification Steps: *Verify that the users are created and assigns to the groups as we specified.*

Scenario 1: Admin creates user(s) and assign(s) to group(s) using batch scripts

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.7.1	Run the batch script and pass the file name as an argument...	File path=invalid Others=valid data	Error message should be displayed			
7.7.2	Run the batch script and pass the file name as an argument...	File format=wrong Others=valid data	Error message should be displayed			
7.7.3	Run the batch script and pass the file name as an argument...	File path=valid User=already existed Others=valid data	Error message should be displayed in log file for the user			
7.7.4	Run the batch script and pass the file name as an argument...	File path=valid User=valid Group=invalid Others=valid data	Error message should be displayed in log file for the user			
7.7.5	Run the batch script and pass the file name as an argument...	File path=valid User=valid Group=valid Others=valid data	Users are created and assigned to the groups we specified in the text file			

7.8 Test Item: Delete and Disable users logged into Company.

Scope: Test for Admin disables use for Desktop Browser.

Action: Admin disables user account of desktop.

Test Notes & Preconditions: For each step below, the user state must be reset to 'Active' and the user must be logged in and display the portion of the UI as indicated.

Verification Steps: Verify the user account has been disabled or not.

Scenario 1: Test the affects of disabling a user that is actively using Company with a desktop browser.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.8.1	UI Location: Email Tab Admin disables user...	Click Home link	Login page should be displayed since user has been deactivated.			
7.8.2	UI Location: Email Tab Admin disables user...	Click	Login page should be displayed since user has been deactivated.			
7.8.3	UI Location: Email Tab Admin disables user...	Click Help Tab	Login page should be displayed since user has been deactivated.			
7.8.4	UI Location: Email Tab Admin disables user...	Click Logout	Login page should be displayed since user has been deactivated.			
7.28.5	UI Location: Email Tab / Access Account(s) Admin disables user...	Click Speed Messages link	Login page should be displayed since user has been deactivated.			
7.28.6	UI Location: Email Tab / Access Account(s) Admin disables user...	Click Compose link	Login page should be displayed since user has been deactivated.			
7.28.7	UI Location: Email Tab / Access Account(s) Admin disables user...	Click Notifications link	Login page should be displayed since user has been deactivated.			
7.28.8	UI Location: Email Tab / Access Account(s) Admin disables user...	Click Configure link	The 'Configure" link should be removed since the user is disabled			
7.28.9	UI Location: Email Tab / Access Account(s)	Click an email account link	Login page should be displayed since user			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	Admin disables user...		has been deactivated.			
7.28.10	UI Location: Email Tab / Access Account(s) Admin disables user...	Click Access Account(s) link	Login page should be displayed since user has been deactivated.			
7.28.11	UI Location: Email Tab / Speed Messages Admin disables user...	Click Add Speed Message link	Login page should be displayed since user has been deactivated.			
7.28.12	UI Location: Email Tab / Speed Messages Admin disables user...	Click edit speed message link	Login page should be displayed since user has been deactivated.			
7.28.13	UI Location: Email Tab / Speed Messages Admin disables user...	Click delete speed message link	Login page should be displayed since user has been deactivated.			
7.28.14	UI Location: Edit Speed Message page displayed. Admin deletes user...	Click Update button	Login page should be displayed since user has been deleted.			
7.28.15	UI Location: Email Tab / Compose Admin disables user...	Click Send button	Login page should be displayed since user has been deactivated.			
7.28.16	UI Location: Email Tab / Notifications Admin disables user...	Click Update button	Login page should be displayed since user has been deactivated.			
7.28.17	UI Location: Email Tab / Configure Admin disables user...	Click an email account link	Login page should be displayed since user has been deactivated.			
7.28.18	UI Location: Configure email account Admin disables user...	Click Update button	Login page should be displayed since user has been deactivated.			
7.28.19	UI Location: Email Account opened Admin disables user...	Click a message to open	Login page should be displayed since user has been deactivated.			
7.28.20	UI Location: Email Account opened Admin disables user...	Click Reload	Login page should be displayed since user has been deactivated.			
7.28.21	UI Location: Email Account opened Admin disables user...	Click Older messages	Login page should be displayed since user has been deactivated.			
7.28.22	UI Location: Email Account opened Admin disables user...	Click Close Mailbox	Login page should be displayed since user has been deactivated.			
7.28.23	UI Location: Email Account opened Admin disables user...	Click Connection Info	Login page should be displayed since user has been deactivated.			
7.28.24	UI Location: Email Message opened Admin disables user...	Click Attachments	Login page should be displayed since user has been deactivated.			
7.28.25	UI Location: Email Message opened Admin disables user...	Click Mailbox	Login page should be displayed since user has been deactivated.			
7.28.26	UI Location: Email Message opened Admin disables user...	Click Full View	Login page should be displayed since user has been deactivated.			
7.28.27	UI Location: Email Message opened Admin disables user...	Click Reply	Login page should be displayed since user has been deactivated.			
7.28.28	UI Location: Email Message opened Admin disables user...	Click Reply to all	Login page should be displayed since user has been deactivated.			
7.28.29	UI Location: Email Message opened Admin disables user...	Click Forward	Login page should be displayed since user has been deactivated.			
7.28.30	UI Location: Email Message opened Admin disables user...	Click Delete	Login page should be displayed since user has been deactivated.			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.28.31	UI Location: Email Message opened Admin disables user...	Click Connection Info	Login page should be displayed since user has been deactivated.			
7.28.32	UI Location: Email Attachment list opened Admin disables user...	Click an attachment	Login page should be displayed since user has been deactivated.			
7.28.33	UI Location: Web Tab Admin disables user...	Click Email Tab	Login page should be displayed since user has been deactivated.			
7.28.34	UI Location: Web Tab / Browse Admin disables user...	Click Go button	Login page should be displayed since user has been deactivated.			
7.28.35	UI Location: Web Tab / Browse Admin disables user...	Click Bookmarks link	Login page should be displayed since user has been deactivated.			
7.28.36	UI Location: Web Tab / Browse Admin disables user...	Click History link	Login page should be displayed since user has been deactivated.			
7.28.37	UI Location: Web Tab / Browse Admin disables user...	Click Settings link	Login page should be displayed since user has been deactivated.			
7.28.38	UI Location: Web Tab / Browse Admin disables user...	Click Browse link	Login page should be displayed since user has been deactivated.			
7.28.39	UI Location: Web Tab / Bookmarks Admin disables user...	Click Add Bookmark link	Login page should be displayed since user has been deactivated.			
7.28.40	UI Location: Web Tab / Bookmarks Admin disables user...	Click Clear Bookmarks link	Login page should be displayed since user has been deactivated.			
7.28.41	UI Location: Web Tab / Bookmarks Admin disables user...	Click Original link	Page should be loaded			
7.28.42	UI Location: Web Tab / Bookmarks Admin disables user...	Click via Company link	Login page should be displayed since user has been deactivated.			
7.28.43	UI Location: Web Tab / Bookmarks Admin disables user...	Click Edit Bookmark link	Login page should be displayed since user has been deactivated.			
7.28.44	UI Location: Web Tab / Bookmarks Admin disables user...	Click Delete Bookmark link	Login page should be displayed since user has been deactivated.			
7.28.45	UI Location: Edit Bookmark page displayed Admin disables user...	Click Update button	Login page should be displayed since user has been deactivated.			
7.28.46	UI Location: Web Tab / History Admin disables user...	Click Clear History link	Login page should be displayed since user has been deactivated.			
7.28.47	UI Location: Web Tab / History Admin disables user...	Click Original link	Page should be loaded			
7.28.48	UI Location: Web Tab / History Admin disables user...	Click via Company link	Login page should be displayed since user has been deactivated.			
7.28.49	UI Location: Web Tab / Caches Admin disables user...	Click Clear Cookies button	Login page should be displayed since user has been deactivated.	(Settings is now called 'caches')		
7.28.50	UI Location: Web Tab / Caches Admin disables user...	Click View Cookies button	Login page should be displayed since user has been deactivated.	(Settings is now called 'caches')		
7.28.51	UI Location: Web Tab / Caches Admin disables user...	Click Clear Auto-completions button	Login page should be displayed since user has been deactivated.	(Settings is now called 'caches')		
7.28.52	UI Location: Web Tab /	Click View Auto-	Login page should be	(Settings is now		

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	Caches Admin disables user...	completions button	displayed since user has been deactivated.	called 'caches')		
7.28.53	UI Location: Web Page opened Admin disables user...	Click Next Part	Login page should be displayed since user has been deactivated.			
7.28.54	UI Location: Web Page opened Admin disables user...	Click Previous Part	Login page should be displayed since user has been deactivated.			
7.28.55	UI Location: Web Page opened Admin disables user...	Click Top	Login page should be displayed since user has been deactivated.			
7.28.56	UI Location: Web Page opened Admin disables user...	Click Reload	Login page should be displayed since user has been deactivated.			
7.28.57	UI Location: Web Page opened Admin disables user...	Click Full View	Login page should be displayed since user has been deactivated.			
7.28.58	UI Location: Web Page opened Admin disables user...	Click Mark	Login page should be displayed since user has been deactivated.			
7.28.59	UI Location: Web Page opened Admin disables user...	Click Connection Info	Login page should be displayed since user has been deactivated.			
7.28.60	UI Location: My Accounts Tab Admin disables user...	Click Devices link	Login page should be displayed since user has been deactivated.			
7.28.61	UI Location: My Accounts Tab Admin disables user...	Click Password link	Login page should be displayed since user has been deactivated.			
7.28.62	UI Location: My Accounts Tab / Devices Admin disables user...	Click Update button	Login page should be displayed since user has been deactivated.			
7.28.63	UI Location: My Accounts Tab / Devices Admin disables user...	Click Show all link	Login page should be displayed since user has been deactivated.			
7.28.64	UI Location: My Accounts Tab / Devices Admin disables user...	Click Delete device link	Login page should be displayed since user has been deactivated.			
7.28.65	UI Location: My Accounts Tab / Password Admin disables user...	Click Change Password button	Login page should be displayed since user has been deactivated.			
7.28.66	UI Location: Help Tab Admin disables user...	Click Troubleshooting link	Login page should be displayed since user has been deactivated.			
7.28.67	UI Location: Help Tab Admin disables user...	Click About link	Login page should be displayed since user has been deactivated.			
7.28.68	UI Location: Help Tab Admin disables user...	Click Introduction link	Login page should be displayed since user has been deactivated.			

Scope: Test for Admin disables user in WAP browser.

Test Notes & Preconditions: For each step below, the user state must be reset to 'Active' and the user must be logged in and display the portion of the UI as indicated.

Scenario 2: Test the affects of disabling a user that is actively using Company with a WAP browser.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.28.69	UI Location: Company Home	Click Email link	Login page should be displayed since			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	Admin disables user...		user has been deactivated.			
7.28.70	UI Location: Company Home Admin disables user...	Click Bookmarks link	Login page should be displayed since user has been deactivated.			
7.28.71	UI Location: Company Home Admin disables user...	Click Go to web link	Login page should be displayed since user has been deactivated.			
7.28.72	UI Location: Company Home Admin disables user...	Click Preferences link	Login page should be displayed since user has been deactivated.			
7.28.73	UI Location: Company Home Admin disables user...	Click Info link	Login page should be displayed since user has been deactivated.			
7.28.74	UI Location: Company Home Admin disables user...	Click Log out link	Login page should be displayed since user has been deactivated.		Bug #5579	
7.28.75	UI Location: Email accounts list Admin disables user...	Click Email account link	Login page should be displayed since user has been deactivated.			
7.28.76	UI Location: Email accounts list Admin disables user...	Click Reload button	Login page should be displayed since user has been deactivated.			
7.28.77	UI Location: Bookmarks list Admin disables user...	Click Bookmark link	Login page should be displayed since user has been deactivated.			
7.28.78	UI Location: Bookmarks list Admin disables user...	Click Home button	Login page should be displayed since user has been deactivated.			
7.28.79	UI Location: Go to Web options Admin disables user...	Click Visit URL link	Login page should be displayed since user has been deactivated.			
7.28.80	UI Location: Go to Web options Admin disables user...	Click History link	Login page should be displayed since user has been deactivated.			
7.28.81	UI Location: Preferences page Admin disables user...	Click Update	Login page should be displayed since user has been deactivated.			
7.28.82	UI Location: Info page Admin disables user...	Click Help	Login page should be displayed since user has been deactivated.			
7.28.83	UI Location: Info page Admin disables user...	Click Device Info	Login page should be displayed since user has been deactivated.			

Scope: *Test for Admin deletes user.*

Test Notes & Preconditions: *For each step below, the user state must be reset to 'Active' and the user must be logged in and display the portion of the UI as indicated.*

Scenario 3: Test the affects of deleting a user that is actively using Company with a desktop browser.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.28.84	UI Location: Email Tab Admin deletes user...	Click Web Tab	Login page should be displayed since user has been deleted.			
7.28.85	UI Location: Email Tab Admin deletes user...	Click My Account Tab	Login page should be displayed since user has been deleted.			
7.28.86	UI Location: Email Tab Admin deletes user...	Click Help Tab	Login page should be displayed since user has been deleted.			
7.28.87	UI Location: Email Tab Admin deletes user...	Click Logout	Login page should be displayed since user has been deleted.			
7.28.88	UI Location: Email Tab / Access Account(s) Admin deletes user...	Click Speed Messages link	Login page should be displayed since user has been deleted.			
7.28.89	UI Location: Email Tab / Access Account(s) Admin deletes user...	Click Compose link	Login page should be displayed since user has been deleted.			
7.28.90	UI Location: Email Tab / Access Account(s) Admin deletes user...	Click Notifications link	Login page should be displayed since user has been deleted.			
7.28.91	UI Location: Email Tab / Access Account(s) Admin deletes user...	Click Configure link	Login page should be displayed since user has been deleted.			
7.28.92	UI Location: Email Tab / Access Account(s) Admin deletes user...	Click an email account link	Login page should be displayed since user has been deleted.			
7.28.93	UI Location: Email Tab / Access Account(s) Admin deletes user...	Click Access Account(s) link	Login page should be displayed since user has been deleted.			
7.28.94	UI Location: Email Account opened Admin deletes user...	Click a message to open	Login page should be displayed since user has been deactivated.			
7.28.95	UI Location: Email Account opened Admin deletes user...	Click Reload	Login page should be displayed since user has been deactivated.			
7.28.96	UI Location: Email Account opened Admin deletes user...	Click Older messages	Login page should be displayed since user has been deactivated.			
7.28.97	UI Location: Email Account opened Admin deletes user...	Click Close Mailbox	Login page should be displayed since user has been deactivated.			
7.28.98	UI Location: Email Account opened Admin deletes user...	Click Connection Info	Login page should be displayed since user has been deactivated.			
7.28.99	UI Location: Email Message opened Admin deletes user...	Click Attachments	Login page should be displayed since user has been deactivated.			
7.28.100	UI Location: Email Message opened Admin deletes user...	Click Mailbox	Login page should be displayed since user has been deactivated.			
7.28.101	UI Location: Email Message opened Admin deletes user...	Click Full View	Login page should be displayed since user has been deactivated.			
7.28.102	UI Location: Email Message opened Admin deletes user...	Click Reply	Login page should be displayed since user has been deactivated.			
7.28.103	UI Location: Email Message opened Admin deletes user...	Click Reply to all	Login page should be displayed since user has been deactivated.			
7.28.104	UI Location: Email Message opened Admin deletes user...	Click Forward	Login page should be displayed since user has been deactivated.			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7. 28.105	UI Location: Email Message opened Admin deletes user...	Click Delete	Login page should be displayed since user has been deactivated.			
7. 28.106	UI Location: Email Message opened Admin deletes user...	Click Connection Info	Login page should be displayed since user has been deactivated.			
7. 28.107	UI Location: Email Attachment list opened Admin deletes user...	Click an attachment	Login page should be displayed since user has been deactivated.			
7. 28.108	UI Location: Email Tab / Speed Messages Admin deletes user...	Click Add Speed Message link	Login page should be displayed since user has been deleted.			
7. 28.109	UI Location: Email Tab / Speed Messages Admin deletes user...	Click edit speed message link	Login page should be displayed since user has been deleted.			
7. 28.110	UI Location: Email Tab / Speed Messages Admin deletes user...	Click delete speed message link	Login page should be displayed since user has been deleted.			
7. 28.111	UI Location: Edit Speed Message page displayed. Admin deletes user...	Click Update button	Login page should be displayed since user has been deleted.			
7. 28.112	UI Location: Email Tab / Compose Admin deletes user...	Click Send button	Login page should be displayed since user has been deleted.			
7. 28.113	UI Location: Email Tab / Notifications Admin deletes user...	Click Update button	Login page should be displayed since user has been deleted.			
7. 28.114	UI Location: Email Tab / Configure Admin deletes user...	Click an email account link	Login page should be displayed since user has been deleted.			
7. 28.115	UI Location: Configure email account Admin deletes user...	Click Update button	Login page should be displayed since user has been deactivated.			
7. 28.116	UI Location: Web Tab Admin deletes user...	Click Email Tab	Login page should be displayed since user has been deleted.			
7. 28.117	UI Location: Web Tab / Browse Admin deletes user...	Click Go button	Login page should be displayed since user has been deleted.			
7. 28.118	UI Location: Web Tab / Browse Admin deletes user...	Click Bookmarks link	Login page should be displayed since user has been deleted.			
7. 28.119	UI Location: Web Tab / Browse Admin deletes user...	Click History link	Login page should be displayed since user has been deleted.			
7. 28.120	UI Location: Web Tab / Browse Admin deletes user...	Click Settings link	Login page should be displayed since user has been deleted.			
7. 28.121	UI Location: Web Tab / Browse Admin deletes user...	Click Browse link	Login page should be displayed since user has been deleted.			
7. 28.122	UI Location: Web Tab / Bookmarks Admin deletes user...	Click Add Bookmark link	Login page should be displayed since user has been deleted.			
7. 28.123	UI Location: Web Tab / Bookmarks Admin deletes user...	Click Clear Bookmarks link	Login page should be displayed since user has been deleted.			
7. 28.124	UI Location: Web Tab / Bookmarks Admin deletes user...	Click Original link	Login page should be displayed since user has been deleted.			
7. 28.125	UI Location: Web Tab / Bookmarks Admin deletes user...	Click via Company link	Login page should be displayed since user has been deleted.			
7. 28.126	UI Location: Web Tab / Bookmarks	Click Edit Bookmark link	Login page should be displayed since user			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	Admin deletes user...		has been deleted.			
7. 28.127	UI Location: Web Tab / Bookmarks Admin deletes user...	Click Delete Bookmark link	Login page should be displayed since user has been deleted.			
7. 28.128	UI Location: Edit Bookmark page displayed Admin disables user...	Click Update button	Login page should be displayed since user has been deactivated.			
7. 28.129	UI Location: Web Tab / History Admin deletes user...	Click Clear History link	Login page should be displayed since user has been deleted.			
7. 28.130	UI Location: Web Tab / History Admin deletes user...	Click Original link	Login page should be displayed since user has been deleted.			
7. 28.131	UI Location: Web Tab / History Admin deletes user...	Click via Company link	Login page should be displayed since user has been deleted.			
7. 28.132	UI Location: Web Tab / Settings Admin deletes user...	Click Clear Cookies button	Login page should be displayed since user has been deleted.			
7. 28.133	UI Location: Web Tab / Settings Admin deletes user...	Click Clear Auto-completions button	Login page should be displayed since user has been deleted.			
7. 28.134	UI Location: Web Page opened Admin deletes user...	Click Next Part	Login page should be displayed since user has been deactivated.			
7. 28.135	UI Location: Web Page opened Admin deletes user...	Click Previous Part	Login page should be displayed since user has been deactivated.			
7. 28.136	UI Location: Web Page opened Admin deletes user...	Click Top	Login page should be displayed since user has been deactivated.			
7. 28.137	UI Location: Web Page opened Admin deletes user...	Click Reload	Login page should be displayed since user has been deactivated.			
7. 28.138	UI Location: Web Page opened Admin deletes user...	Click Full View	Login page should be displayed since user has been deactivated.			
7. 28.139	UI Location: Web Page opened Admin deletes user...	Click Mark	Login page should be displayed since user has been deactivated.			
7. 28.140	UI Location: Web Page opened Admin deletes user...	Click Connection Info	Login page should be displayed since user has been deactivated.			
7. 28.141	UI Location: My Accounts Tab Admin deletes user...	Click Devices link	Login page should be displayed since user has been deleted.			
7. 28.142	UI Location: My Accounts Tab Admin deletes user...	Click Password link	Login page should be displayed since user has been deleted.			
7. 28.143	UI Location: My Accounts Tab / Devices Admin deletes user...	Click Update button	Login page should be displayed since user has been deleted.			
7. 28.144	UI Location: My Accounts Tab / Devices Admin deletes user...	Click Show all link	Login page should be displayed since user has been deleted.			
7. 28.145	UI Location: My Accounts Tab / Devices Admin deletes user...	Click Delete device link	Login page should be displayed since user has been deleted.			
7. 28.146	UI Location: My Accounts Tab / Password Admin deletes user...	Click Change Password button	Login page should be displayed since user has been deleted.			
7. 28.147	UI Location: Help Tab Admin deletes user...	Click Troubleshooting link	Login page should be displayed since user has been deleted.			
7. 28.148	UI Location: Help Tab	Click About link	Login page should be			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	Admin deletes user...		displayed since user has been deleted.			
7. 28.149	UI Location: Help Tab Admin deletes user...	Click Introduction link	Login page should be displayed since user has been deleted.			

Scope: *Test for Admin deletes user.*

Test Notes & Preconditions: *For each step below, the user state must be reset to 'Active' and the user must be logged in and display the portion of the UI as indicated.*

Scenario 4: *Test the affects of deleting a user that is actively using Company with a WAP browser.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7. 28.150	UI Location: Company Home Admin deletes user...	Click Home tab	Login page should be displayed since user has been deleted.			
7. 28.151	UI Location: Company Home Admin deletes user...	Click Server tab	Login page should be displayed since user has been deleted.			
7. 28.152	UI Location: Company Home Admin deletes user...	Click Phone tab	Login page should be displayed since user has been deleted.			
7. 28.153	UI Location: Company Home Admin deletes user...	Click sync tab	Login page should be displayed since user has been deleted.			
7. 28.154	UI Location: Company Home Admin deletes user...	Click password tab	Login page should be displayed since user has been deleted.			
7. 28.155	UI Location: Company Home Admin deletes user...	Click Log out link	Login page should be displayed since user has been deleted.			

7.9 Test Item: *Email to All Users*

Scope: *Test the functionality of admin email.*

Action: *Send email with input values mentioned below*

Test Notes & Preconditions: *Make sure admin has valid and existing email id. Click on Email All Users button in List all users page, in order to compose a mail.*

Verification Steps: *Verify that the messages are successfully sent and opened correctly.*

Scenario 1: *Test with blank components of email*

Scope: *Test various input values for subject and body.*

S#	Action	Input Value	Expected Results	Actual Results	P/F
7.29.1	Send a message with specified input values.	Subject = Blank Message = Blank	Message is sent successfully.		
7.29.2	Open the message received from admin from	Subject = Blank Message = Blank	Message is opened successfully.		

S#	Action	Input Value	Expected Results	Actual Results	P/F
	the active user account				
7.29.3	Send a message with specified input values.	Subject = Non blank Message = Blank	Message is sent successfully.		
7.29.4	Open the message received from admin from the active user account	Subject = Non blank Message = Blank	Message is opened successfully.		
7.29.5	Send a message with specified input values.	Subject = Blank Message = Non Blank	Message is sent successfully.		
7.29.6	Open the message received from admin from the active user account	Subject = Blank Message = Non Blank	Message is opened successfully.		
7.29.7	Send a message with specified input values.	Subject = Non Blank Message = Non Blank	Message is sent successfully.		
7.29.8	Open the message received from admin from the active user account	Subject = Non Blank Message = Non Blank	Message is opened successfully.		

Scenario 2: *Some string length and special character tests*

Scope: *Test short and long strings and special characters in various email components.*

S#	Action	Input Value	Expected Results	Actual Results	P/F
7.29.9	Send the message of specified char length for subject and body.	Subject =1 char Body = 1char	Message is sent successfully		
7.29.10	Open the message received from admin from the active user account	Subject =1 char Body = 1char	Message is opened successfully.		
7.29.11	Send the message of specified char length for subject and body.	Subject = Maximum allowed length or simply very long. Body = Maximum allowed length or simply very long.	Message is sent successfully.		
7.29.12	Open the message received from admin from the active user account	Subject = Maximum allowed length or simply very long. Body = Maximum allowed length or simply very long.	Message is opened successfully.		
7.29.13	Send the message of specified char length for subject and body.	Subject =Contains all special character. Body = Contains all special characters.	Message is sent successfully.		
7.29.14	Open the message received from admin from the active user account	Subject = Contains all special character Body = Contains all special characters	Message is opened successfully.		
7.29.15	Send the message of specified char length for subject and body.	Subject =Contains only "###". Body = Contains only "###".	Message is sent successfully.		
7.29.16	Open the message received from admin from the active user account	Subject =Contains only "###". Body = Contains only "###".	Message is opened successfully.		

Scenario 5: *Basic Send and Receive*

Scope: *Test sending mail to all users*

Verification Steps: *Verify that the message is sent only to active users.*

S#	Action	Input Value	Expected Results	Actual Results	P/F
7.29.35	Admin sends an email to all users with valid values in subject and body.	No active user is present.	Message is only sent to admin, since admin is in Bcc by default.		
7.29.36	Check for new messages and open those messages in Admin's 'Reply to' email account.		Messages are opened successfully		
7.29.37	Admin sends an email to all users with valid values in subject and body.	Active and Inactive users are present.	Message is only sent to admin and all the active users. Inactive users should not receive this message.		
7.29.38	Login and check the inactive user's primary email account for new messages, after enabling the inactive user by admin.		No Messages were received from admin, since the user was disabled.		
7.29.39	Admin sends an email to all users with valid values in subject and body.	Some of the active users do not have primary email accounts.	Message is sent to active users email addresses that were used for SRC account creation.		
7.29.40	Check for new messages received from admin from users email accounts that were used for SRC account creation.	The active user does not have primary email account.	Message is received from admin successfully.		
7.29.41	Admin sends an email to all users with valid values in subject and body.	Active and Inactive users are present.	Message is sent to admin and all the active users. And the header of the received message should correctly display the admin's mail address.		
7.29.42	Admin composes and sends a message with valid values in subject and body.	No connection between SRC and SMTP server.	The error page should be displayed with the "Try again" and "Cancel" buttons.		
7.29.43	Admin clicks cancel button of the error page.		The main "Users" page should be displayed. No mail is sent.		
7.29.44	Check for new messages in Admin's reply to email account.		No mails from admin were received since the email sending was cancelled.		
7.29.45	Admin clicks on the "Try again" button of the error page.	Connection between SRC and SMTP is fine.	Email will be sent to the active users. The success screen should be displayed with the message "Mail sent to all enabled users".		
7.29.46	Check for new messages and open those mails in Admin's reply to email account.		Message are opened successfully		

Scenario 6: *Administrator changes "reply to" address*

Scope: Test for changing administrator's reply to address.

S#	Action	Input Value	Expected Results	Actual Results	P/F
7.29.47	Admin runs the reconfig script from command line and change the reply to address.	Save changes and restart SRC	The new 'Reply to' email address is used when admin sends an email message to active users.		
7.29.48	Admin sends an email to all users with valid input		Message is sent successfully with new 'Reply-to' email address		
7.29.49	Login to SRC as an active User. Open the email received from admin and click on 'reply to' link.		The email address in the 'To' header should be the new 'Reply-to' email address.		
7.29.50	Admin runs the reconfig script from command line and change the reply to address.	Abort the process before saving.	The old 'Reply to' email address is used when admin sends an email message to active users.		
7.29.51	Admin sends an email to all users with valid input		Message is sent successfully with the old 'Reply-to' email address		
7.29.52	Login to SRC as an active User. Open the email received from admin and click on 'reply to' link.		The email address in the 'To' header should be the old 'Reply-to' email address.		

7.10 Test Item: Logout

Scope: Test that no operations can be performed after Logout.

Action: Admin Logs out, chooses the browser's <Back> option, clicks any command function.

Test Notes & Preconditions: Admin user is logged into the administration page and has performed some admin functions.

Verification Steps: Verify the Admin is accessing the options of Company even if Admin selects logout option.

Scenario 1: Test using the browser's <Back> option after logout.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.30.1	Admin Logs out, chooses the browser's <Back> option, clicks any command function...		Message should be displayed that session is invalid or expired and provides a link to log back in.			
7.30.2	Click login link and log back in...		Admin is successfully logged back in.			

7.11 Test Item: Multiple browser window usage

Scope: Test simultaneous and conflicting operations from multiple browser windows.

Action: Test simultaneous and conflicting operations from multiple browser windows.

Test Notes & Preconditions: Admin is logged in from two separate browser windows.

Verification Steps: *Verify the status of the 2 users depends on the operations performed on them.*

Scenario 1: *Test simultaneous and conflicting operations from multiple browser windows.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.31.1	Enable 2 different registered users at the same time.		Both users are enabled successfully.			
7.31.2	Enable the same registered user in both browser windows at the same time.		The user is enabled. No error results from the conflict.			
7.31.3	Disable 2 different users at the same time.		Both users are disabled successfully.			
7. 31.4	Disable the same user in both browser windows at the same time.		The user is disabled. No error results from the conflict.			
7. 31.5	Create 2 different users at the same time.		Both users are created successfully.			
7. 31.6	Create the same user in both browser windows at the same time.		The create request that is submitted first creates the user. The second request will give an error that the user already exists.			
7. 31.7	Edit and update 2 different users at the same time.		Info for both users is updated successfully.			
7. 31.8	Edit and update the same user in both browser windows at the same time.		The user info is updated. No error results from the conflict.			
7. 31.9	Delete 2 different users at the same time.		Both users are deleted successfully.			
7. 31.10	Delete the same user in both browser windows at the same time.		The delete request that is submitted first deletes the user. The second request will give an error that the user does not exist.			
7. 31.11	Delete a user in one window, attempt to edit the user in the other window...		A message should be displayed that the user does not exist.		F – bug #5652	
7. 31.12	Delete a user in one window, attempt to disable the user in the other window...		A message should be displayed that the user does not exist.		F – bug #5653	

7.12 Test Item: *Change Admin password*

Scope: *Test the ability to change the Admin's password.*

Action: *User submits Change Password form from the password tab.*

Test Notes & Preconditions: *Admin is logged into the system.*

Verification Steps: *Verify the Admin password to enter in to the admin page with the new password.*

Scenario 1: *Test changing the Admin's password.*

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.32.1	User submits Change Password form...	Old Password = Blank New Password = Blank Confirm Password = Blank	Error message should be displayed because original password is invalid.			
7. 32.2	User submits Change Password form...	Old Password = Invalid New Password = any Confirm Password = any	Error message should be displayed because original password is			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
			invalid.			
7. 32.3	User submits Change Password form...	Old Password = Valid New Password = Blank Confirm Password = Blank	Error message should be displayed because password cannot be blank.			
7. 32.4	User submits Change Password form...	Old Password = Valid New Password = any Confirm Password = Does NOT match	Error message should be displayed because confirmation password must match the newly types password.			
7. 32.5	User submits Change Password form...	Old Password = Valid New Password = Any Confirm Password = Matches	Message should be displayed stating that the password has been changed.			
7. 32.6	Logout and attempt to login with OLD password	Password = OLD password	Admin cannot login with old password. Appropriate error message should be displayed saying that password is invalid.			
7. 32.7	Attempt to login with new password.	Password = valid new password	Admin is logged in successfully.			
7. 32.8	User submits Change Password form...	Old Password = Valid New Password = All special chars Confirm Password = Matches	Message should be displayed stating that the password has been changed.			
7. 32.9	Logout and attempt to login with OLD password	Password = OLD password	User cannot login with old password. Appropriate error message should be displayed saying that password is invalid.			
7. 32.10	Attempt to login with new password.	Password = valid new password	User is logged in successfully.			
7. 32.11	User submits Change Password form...	Old Password = all special chars New Password = Maximum characters that input field permits Confirm Password = Maximum characters that input field permits	Message should be displayed stating that the password has been changed.			
7. 32.12	Logout and attempt to login with OLD password	Password = OLD password	User cannot login with old password. Appropriate error message should be displayed saying that password is invalid.			
7. 32.13	Attempt to login with new password.	Password = valid new password	User is logged in successfully.			

7.13 Test Item: Sending log files to Company Support.

Scope: Test sending log files to Company support or to specified email address

Action: Send log files to Company support or other Email ID with specified size, logs of specified date range and with given comments.

Test Notes: This option available only in desktop browsers and only in administrator site. This ID can be modified by the admin through desktop. The duration of the error logs in the log file, the maximum size of the log files, selection of log files can also be configured before sending logs.

Verification Steps: *The default values are always be set automatically when the page is revisited.*

Verification Steps: *Verify sending system logs as per the configuration set up by the admin. Specified interval, specified log files registers the new start time and roll time.*

Scenario 1: *Test sending logs for different configuration.*

S#	Action	Input Value	Expected Result	Actual Result	P/F	Automated
7.43.1	Click on 'System logs' tab		N1 should display the System logs page, which has a link of 'Sending logs' at the top of the page.			
7.43.2	Click on 'Sending logs' link		N1 should display the 'sending logs' page with log file names, email input field, maximum size of logs/email, comments input field and date fields.			
7.43.3	Verify the default maximum no of bytes		By default this should be 2MB only.			
7.43.4	Verify the default email address in the recipient email address		By default it should be def.support@Company.com			
7.43.5	Click on 'Submit' button	Log files= one or all Other fields= default	N1 should send the selected files as per the default values given.			
7.43.6	Modify the maximum number of bytes per email, and click on 'Submit' button	Log file= any of the available or all log files Input = NULL	N1 should display an error of "Max size per email is not entered. Please enter a size" and should have a link to abort the current process			
7.43.7	Modify the maximum number of bytes per email, and click on 'Submit' button	Log file= any of the available or all log files Input = Zero	N1 should display an error of "Max size per email should have a valid data" and should have a link to abort the current process			
7.43.8	Modify the maximum number of bytes per email, and click on 'Submit' button	Log file= any of the available or all log files Input = negative value	N1 should display an error of "Max size per email should have a valid data" and should have a link to abort the current process			
7.43.9	Modify the maximum number of bytes per email, and click on 'Submit' button	Log file= any of the available or all log files Input = positive decimal value	N1 should allow the max size given and send the log files			
7.43.10	Modify the maximum number of bytes per email, and click on 'Submit' button	Log file= any of the available or all log files Input = any positive value.	N1 should allow the max size given and send the log files			
7.43.11	Modify the email address and click on 'Submit' button.	Log file = any Email Address= Empty	N1 Should display an error saying that "Please enter a valid email address" and should have a link to abort the current process			
7.43.12	Modify the email address and click on 'Submit' button.	Log file = any Email Address= syntactically Invalid	N1 Should display an error saying that "Please enter a valid email address" and should have a link to abort the current process			
7.43.13	Modify the email address and click on 'Submit' button.	Log file = any Email Address= Invalid (But Syntactically Valid)	N1 should accept the email id given and should send the logs to the same. Later the admin should get an undelivered mail.			
7.43.14	Modify the email address and click on 'Submit' button.	Log file = any Email Address= Valid (But Syntactically Valid)	N1 should accept the email id given and should send the logs to the same.			
7.43.15	Click on the submit button	Comments = Empty All other field=Valid	N1 should send the selected log files to the email address specified.			
7.43.16	Click on 'Submit' button	Log file = None All other fields = Valid	N1 should display an error saying "No log file is selected. Please select one or more log files to send" and should have a link to abort the current process			
7.43.17	Click on 'Submit' button	Log file = One All other fields = Valid	N1 should send the selected log file only to the email specified.			

S#	Action	Input Value	Expected Result	Actual Result	P/F	Automated
7.43.18	Click on 'Submit' button	Log file = More than One All other fields = Valid	N1 should send the selected log files only to the email specified.			
7.43.19	Mark the 'Select all logs' check box then click on 'Submit' button	All other fields = valid	N1 should send the selected log files only to the email specified.			
7.43.20	Unmark the 'Date range' check box and click on 'Submit' button	All other fields = Valid	The date range fields should be disabled and N1 should send the full log files selected without any date range filtering.			
7.43.21	Mark the 'Date range' check box and click on 'Submit button	All other Fields = Valid	N1 should send the log files for the selected date range only to the specified date range.			
7.43.22	Select the date range and click on 'Submit' button	From= current day Through= current day All other fields = Valid	N1 should send the log files for the current day only to the specified date range.			
7.43.23	Select the date range and click on 'Submit' button	From and Through date = same All other fields = Valid	N1 should send the log files for the day specified only to the specified date range.			
7.43.24	Select the date range and click on 'Submit' button	From= later than through date Through= current All other fields = Valid	N1 should display an appropriate error stating that the date range selected is invalid.			
7.43.25	Select the date range and click on 'Submit' button	From and Through= invalid Mm=45, dd=55,yyyy=2004 All other fields = Valid	N1 should correct the date to the nearest or previous date ranges and proceed to send the log files.			
7.43.26	Select the date range and click on 'Submit' button	From= Valid Through= Valid later than the 'From' date All other fields = Valid	N1 should send the selected log files for the specified date range to the email specified.			
7.43.27	Click on 'Submit' button	All fields = Valid data	N1 should start the process of sending logs by displaying a message contains the options to abort and continue waiting. And it should auto refresh every 10 sec.			
7.43.28	Select 'continue waiting' option while N1 processing the sending logs		N1 should refresh the screen and continue sending logs.			
7.43.29	Select 'abort' option while N1 processing the sending logs		N1 should abort the sending logs process and should show the process completion screen.			
7.43.30	Select 'abort' option in the middle of the process or a bit later		N1 should abort sending logs, but it might have sent some log files.			
7.43.31	In system logs page click on 'Submit button'	All fields = Valid data	N1 should start sending log files to the specified email ID.			
7.43.32	When the sending logs is in progress go to other screens (web, email or any other tab) and re-visits the 'System logs'>'Sending logs' screen		N1 should show the progress screen again and sending logs process still be continued.			
7.43.33	Click on 'Submit' button...	Max bytes/email = 1MB or other Selected log file > above max value set	N1 should show an appropriate error stating that log files selected is more than the specified byte size and give the following options: 1. Abort sending all files 2. Abort sending only the files that are too large 3. Send files anyway			
7.43.34	Select 'Abort sending all files' option and click on		N1 should abort sending log files and should display process completion			

S#	Action	Input Value	Expected Result	Actual Result	P/F	Automated
	'Submit' button...		screen.			
7.43.35	Select 'Abort sending only the files that are too large' option and click on 'Submit' button...		N1 should abort sending the larger files as specified and send the other log files selected if any and should display process completion screen.			
7.43.36	Select ' Send all files anyway and click on 'Submit' button...		N1 sends all the selected log files and display the process completion screen.			
7.43.37	Verify the process completion screen		It should display the list of files if any, which are successfully sent to the specified email and list of log files if any, which are failed to send.			
7.43.38	If any errors occurs while sending logs...		N1 should show the progress page with the list of log files, which are failed to send.			

7.14 Test Item: Refresh after submits

Scope: Test clicking refresh in the desktop browser after submitting a form. The reason for this is at one point in time, the system would resubmit the same form and cause an error. Refresh should not do this.

Action: Enable a new user registration request. Choose Refresh.

Test Notes & Preconditions: Login to Admin page.

Verification Steps: Verify the display results are not changing even after selecting the refresh option from the browser's window.

Scenario 1: Test browser refresh after submitting forms.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.44.1	Enable a new user registration request. Choose Refresh.		The page is simply redisplayed.			
7.44.2	Create a new user. Refresh.		The page is simply redisplayed.			
7.44.4	Delete a user. Refresh.		The page is simply redisplayed.			

7.15 Test Item: User/Admin Level Privileges

Scope: Test the ability to check admin level privileges

Action: Open 'List all users' link from the Users tab.

Test Notes & Preconditions: Admin is logged into the system and click on the Users Tab and click on list all users.

Verification Steps: Verify that the Admin level privilege options are displayed as mentioned below.

Scenario 1: Test user / admin level privileges.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.45.1	Display configure option		Admin should be able to see the configure hyperlink in Action Items for every user.			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.45.2	Display user pages to Admin	Click on configure option hyper link from action items for particular user.	It should take admin to particular user's account, just as if the user would have logged in (Admin will have full functionality except that he unable to access the user's mail system like view messages or compose messages)			

7.16 Test Item: Session validation

Scope: Test the admin session validation.

Action: Test the admin session validation for new user creation.

Test Notes & Preconditions: Admin is logged into the system and click on the Users Tab.

Verification Steps: Verify that the session is expired after some time.

Scenario 1: Test the admin session validation.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.46.1	Admin session validation	After creating an Online request, Login into Admin UI screen select "Enable/Delete" button under "Request-Registration Request" after the session expires.	It should display the following message "Session invalid or expired. Click here to login"			
7.46.2	Admin session validation	Do an "upgrade request" from Mobile Device. Login into Admin UI screen select "Enable/Delete" button under "Request-Upgrade Request" after the session expires.	It should display the following message "Session invalid or expired. Click here to login"			
7.46.3	Admin session validation	Let the system sit idle until the user session had expired and then click on "List all Users" link.	It should display the following message "Session invalid or expired. Click here to login"			
7.46.4	Admin session validation	Admin displays create user form, fill out the form but does not hit "Submit" button until session timed out. Admin then hits "submit" button on the form. Error is displayed that session has timed out. Admin logs back in.	User should not be added to the List of Users.			
7.46.5	Admin session validation	Admin displays "Search" user screen, enter valid data in the search field but does not hit "Search" button until session timed out. Admin then hits "search" button on the screen.	It should display the following message "Session invalid or expired. Click here to login"			
7.46.6	Admin session validation	Let the system sit idle until the user session had expired and then click on "List all Administrators"	It should display the following message "Session invalid or expired. Click here to			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
			login"			
7.46.7	Admin session validation	Admin opens Edit/Configure/Delete/Test user form, fill out the form but does not hit "Delete" button until session timed out. Admin then hits "Delete" button on the form. Error is displayed that session has timed out. Admin logs back in.	Selected user should not be Edited/Configured/deleted/tested from the list of users.			
7.46.8	Admin session validation	Let the system sit idle until the admin user session had expired and then click on "Clear Cache" button under Provision.	It should display the following message "Session invalid or expired. Click here to login"			
7.46.9	Admin session validation	Admin selects "Client Admin" tab under which "Status/Sessions/Users/Usage/Account Admin" should be selected after the session had expired.	It should display the following message "Session invalid or expired. Click here to login"			
7.46.10	Admin session validation	Admin selects "Licensing" tab let the system sit idle after which select "List Users".	It should display the following message "Session invalid or expired. Click here to login"			
7.46.11	Admin session validation	Admin selects "Logs" tab let the system sit idle after which select "Submit" button after entering all the details.	It should display the following message "Session invalid or expired. Click here to login"			
7.46.12	Admin session validation	Admin displays Email form, fill out the form but does not hit "Send" button until session timed out. Admin then hits "Send" button on the form. Error is displayed that session has timed out. Admin log back in.	The message should not be sent to all the active users.			
7.46.13	Admin session validation	Let the system sit idle until the admin user session had expired and then click on "Logout" .	It should display the following message "Session invalid or expired. Click here to login"			

Scope: Login failures

Action: Click on Client admin tab and click on users link and click on client settings

Test Notes & Preconditions: Admin is logged into the system.

Scenario 1: Admin can configure the number of failed login attempts before the client data is wiped.

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
7.19.13	Admin sets "Login failures before data wipe"	Login failure before data wipe=4 or10, or 20 Attempts	User is locked out and client and client data are wiped from the device.			
7.19.13	Admin sets "Login failures	Login failure	Client data should			

S#	Action	Input Value	Expected Results	Actual Results	P/F	Automated
	before data wipe"	before data wipe=No= limit	never wiped during invalid login attempts.			