High level test plan – SparkWeave™

# Testing types

1. Installation-

Installation testing will be performed for the Outlook plugin only. Installation of the product has to be tested on various combinations of Outlook/OS .

1. Performance-

Performance measures need to be made in order to study the stability and behavior of the product under load/stress conditions. JMeter is the proposed tool for this task.

1. Compatibility-

The web interface needs to be tested against various browser versions.

1. Functional-

The product needs to be tested against the functional requirements.

1. Security-

?

1. User Interface-

There needs to be a clear distinction between User Interface testing and usability testing. By user interface testing we mean, testing so that we know if the user interface designed is according to the laid down requirements and conforms according to standards.

1. Usability-

By usability testing we mean, how easy and understandable the product would be for a novice user/user without formal training on the product.

# Testing approach/methodology

With the growing popularity of Agile teams and Agile working environments, we adopt the same notion for testing too. As we know, there is no specific” testing team” in the Agile world, but to bring this notion into practice is practically impossible. We will have separate testing teams.

Testing teams will focus largely on automation. The need for automation being constant code rework.

Manual testing will be performed for the plugin and other areas where automation is not possible/ yet to been done.

Compatibility & Installation testing

# **Browsers Support for the Web Interface**

Our browser support covers the past one year of releases for Firefox, Chrome, and Safari and the past 5 years of releases for Internet Explorer. Compatibility for these browsers encompasses the end-user and administrative interfaces.

##### Safari- 5.1

##### Firefox- 8.0, 7.0, 6.0, 5.0, 4.0, 3.5.16

##### Chrome- 15.0, 14.0, 13.0, 12.0, 11.0, 10.0, 9.0, 8.0

##### Microsoft Internet Explorer- 7, 8, 9

At this time, we are not supporting any version of Opera nor IE6.

# Compatibility Matrix for Outlook Plugin

The following is the compatibility matrix for Microsoft Outlook version and operating system that we intend to support for the Outlook plugin.

|  |  |  |  |
| --- | --- | --- | --- |
| Outlook vs. OS | Outlook 2007 (32 bit) | Outlook 2010 (32 bit) | Outlook 2010 (64 bit) |
| Windows 7 - 32 bit | yes | yes | N/A |
| Windows 7 - 64 bit | yes | yes | yes |
| Windows Vista – 32 bit | yes | yes | N/A |
| Windows Vista – 64 bit | yes | yes | yes |
| Windows XP – 32 bit | yes | yes | N/A |
| Windows XP – 64 bit | yes | no, installation of Outlook 2010 is not possible\* | |

\* The Outlook installer claims SP3 is required, but Microsoft does not have a SP3 available for this OS.

Microsoft Outlook 2007 is available in a 32-bit version only. We will support all versions of Outlook 2007 and Outlook 2010. Outlook 2003 support is not offered at this time, nor is support for alternative mail clients such as Lotus Notes.

# Pre-Installation Checks

In order for the secure email portion of the Outlook plugin to work, you must have a valid SSL certificate installed on the SparkWeave cluster.

## Windows 7 Only

As of 1/20/2012 there are no known upgrade requirements for Windows 7.

## Windows Vista Only

As of 1/20/2012 there are no known upgrade requirements for Windows Vista.

## XP Only

### All Versions

The best way to ensure plugin compatibility on Windows XP is to run automatic updates. Before installing the SparkWeave Outlook Plugin, please ensure:

1. Windows has been patched to SP2 (Service Pack #2) or higher.
2. All of the latest security updates have been applied.

### 64-bit Only

On Win XP 64 bit, WIC( Windows Imaging Component) needs to be installed.

User Interface testing

**Text Boxes**

Move the Mouse Cursor over all Enterable Text Boxes. Cursor should change from arrow to InsertBar. If it doesn't then the text in the box should be grey or non-updateable. Refer to previous page.Enter text into Box. Try to overflow the text by typing to many characters - should be stopped Check the field width with capital W’s. Enter invalid characters - Letters in amount fields, try strange characters like + , - \* etc. in **All** fields. SHIFT and Arrow should Select Characters. Selection should also be possible with mouse. Double Click should select all text in box. Left and Right arrows should move 'ON' Selection. So should Up and Down. Select with mouse by clicking.

**Command buttons**

If Command Button leads to another Screen, and if the user can enter or change details on the other

screen then the Text on the button should be followed by three dots.

All Buttons except for OK and Cancel should have a letter Access to them. This is indicated by a

letter underlined in the button text. The button should be activated by pressing ALT+Letter. Make

sure there is no duplication.

Click each button once with the mouse - This should activate

Tab to each button - Press SPACE - This should activate

Tab to each button - Press RETURN - This should activate

The above are **VERY IMPORTANT**, and should be done for **EVERY** command Button.

Tab to another type of control (not a command button). One button on the screen should be default

(indicated by a thick black border). Pressing Return in ANY no command button control should

activate it.

If there is a Cancel Button on the screen , then pressing <Esc> should activate it.

If pressing the Command button results in uncorrectable data e.g. closing an action step, there should

be a message phrased positively with Yes/No answers where Yes results in the completion of the

action.

**Drop Down List Boxes**

Pressing the Arrow should give list of options. This List may be scrollable. You should not be able to

type text in the box.

Pressing a letter should bring you to the first item in the list with that start with that letter. Pressing

‘Ctrl - F4’ should open/drop down the list box.

Spacing should be compatible with the existing windows spacing (word etc.). Items should be in

alphabetical order with the exception of blank/none which is at the top or the bottom of the list box.

Drop down with the item selected should be display the list with the selected item on the top.

Make sure only one space appears, shouldn't have a blank line at the bottom.

# Screen Validation Checklist

**AESTHETIC CONDITIONS:**

**1.** Is the general screen background the correct colour?.

**2.** Are the field prompts the correct colour?

**3.** Are the field backgrounds the correct colour?

**4.** In read-only mode, are the field prompts the correct colour?

**5.** In read-only mode, are the field backgrounds the correct colour?

**6.** Are all the screen prompts specified in the correct screen font?

**7.** Is the text in all fields specified in the correct screen font?

**8.** Are all the field prompts aligned perfectly on the screen?

**9.** Are all the field edit boxes aligned perfectly on the screen?

**10.** Are all groupboxes aligned correctly on the screen?

**11.** Should the screen be resizable?

**12.** Should the screen be minimisable?

**13.** Are all the field prompts spelt correctly?

**14.** Are all character or alpha-numeric fields left justified? This is the default

unless otherwise specified.

**15.** Are all numeric fields right justified? This is the default unless otherwise

specified.

**16.** Is all the microhelp text spelt correctly on this screen?

**17.** Is all the error message text spelt correctly on this screen?

**18.** Is all user input captured in UPPER case or lower case consistently?

**19.** Where the database requires a value (other than null) then this should be

defaulted into fields. The user must either enter an alternative valid value

or leave the default value intact.

**20.** Assure that all windows have a consistent look and feel.

**21.** Assure that all dialog boxes have a consistent look and feel.

**VALIDATION CONDITIONS:**

**1.** Does a failure of validation on every field cause a sensible user error

message?

**2.** Is the user required to fix entries which have failed validation tests?

**3.** Have any fields got multiple validation rules and if so are all rules being

applied?

**4.** If the user enters an invalid value and clicks on the OK button (i.e. does

not TAB off the field) is the invalid entry identified and highlighted

correctly with an error message.?

**5.** Is validation consistently applied at screen level unless specifically required

at field level?

**6.** For all numeric fields check whether negative numbers can and should be

able to be entered.

**7.** For all numeric fields check the minimum and maximum values and also

some mid-range values allowable?

**8.** For all character/alphanumeric fields check the field to ensure that there is

a character limit specified and that this limit is exactly correct for the

specified database size?

**9.** Do all mandatory fields require user input?

**10.** If any of the database columns don’t allow null values then the

corresponding screen fields must be mandatory. (If any field which initially

was mandatory has become optional then check whether null values are

allowed in this field.)

**NAVIGATION CONDITIONS:**

**1.** Can the screen be accessed correctly from the menu?

**2.** Can the screen be accessed correctly from the toolbar?

**3.** Can the screen be accessed correctly by double clicking on a list control on

the previous screen?

**4.** Can all screens accessible via buttons on this screen be accessed correctly?

**5.** Can all screens accessible by double clicking on a list control be accessed

correctly?

**6.** Is the screen modal. i.e. Is the user prevented from accessing other

functions when this screen is active and is this correct?

**7.** Can a number of instances of this screen be opened at the same time and is

this correct?

Usability Testing

**USABILITY CONDITIONS:**

**1.** Are all the dropdowns on this screen sorted correctly? Alphabetic sorting

is the default unless otherwise specified.

**2.** Is all date entry required in the correct format?

**3.** Have all pushbuttons on the screen been given appropriate Shortcut keys?

**4.** Do the Shortcut keys work correctly?

**5.** Have the menu options which apply to your screen got fast keys associated

and should they have?

**6.** Does the Tab Order specified on the screen go in sequence from Top Left

to bottom right? This is the default unless otherwise specified.

**7.** Are all read-only fields avoided in the TAB sequence?

**8.** Are all disabled fields avoided in the TAB sequence?

**9.** Can the cursor be placed in the microhelp text box by clicking on the text

box with the mouse?

**10.** Can the cursor be placed in read-only fields by clicking in the field with the

mouse?

**11.** Is the cursor positioned in the first input field or control when the screen is

opened?

**12.** Is there a default button specified on the screen?

**13.** Does the default button work correctly?

**14.** When an error message occurs does the focus return to the field in error

when the user cancels it?

**15.** When the user Alt+Tab’s to another application does this have any impact

on the screen upon return to The application?

**16.** Do all the fields edit boxes indicate the number of characters they will hold

by there length? e.g. a 30 character field should be a lot longer

**DATA INTEGRITY CONDITIONS:**

**1.** Is the data saved when the window is closed by double clicking on the

close box?

**2.** Check the maximum field lengths to ensure that there are no truncated

characters?

**3.** Where the database requires a value (other than null) then this should be

defaulted into fields. The user must either enter an alternative valid value

or leave the default value intact.

**4.** Check maximum and minimum field values for numeric fields?

**5.** If numeric fields accept negative values can these be stored correctly on

the database and does it make sense for the field to accept negative

numbers?

**6.** If a set of radio buttons represent a fixed set of values such as A, B and C

then what happens if a blank value is retrieved from the database? (In some

situations rows can be created on the database by other functions which

are not screen based and thus the required initial values can be incorrect.)

**7.** If a particular set of data is saved to the database check that each value

gets saved fully to the database. i.e. Beware of truncation (of strings) and

rounding of numeric values.

**MODES (EDITABLE READ-ONLY) CONDITIONS:**

**1.** Are the screen and field colours adjusted correctly for read-only mode?

**2.** Should a read-only mode be provided for this screen?

**3.** Are all fields and controls disabled in read-only mode?

**4.** Can the screen be accessed from the previous screen/menu/toolbar in readonly

mode?

**5.** Can all screens available from this screen be accessed in read-only mode?

**6.** Check that no validation is performed in read-only mode.

**GENERAL CONDITIONS:**

**1.** Assure the existence of the "Help" menu.

**2.** Assure that the proper commands and options are in each menu.

**3.** Assure that all buttons on all tool bars have a corresponding key commands.

**4.** Assure that each menu command has an alternative(hot-key) key sequence which

will invoke it where appropriate.

**5.** In drop down list boxes, ensure that the names are not abbreviations / cut short

**6.** In drop down list boxes, assure that the list and each entry in the list can be

accessed via appropriate key / hot key combinations.

**7.** Ensure that duplicate hot keys do not exist on each screen

**8.** Ensure the proper usage of the escape key (which is to undo any changes that have

been made) and generates a caution message “Changes will be lost - Continue

yes/no”

**9.** Assure that the cancel button functions the same as the escape key.

**10.** Assure that the Cancel button operates as a Close button when changes have be

made that cannot be undone.

**11.** Assure that only command buttons which are used by a particular window, or in a

particular dialog box, are present. - i.e make sure they don’t work on the screen

behind the current screen.

**12.** When a command button is used sometimes and not at other times, assure that it is

grayed out when it should not be used.

**13.** Assure that OK and Cancel buttons are grouped separately from other command

buttons.

**14.** Assure that command button names are not abbreviations.

**15.** Assure that all field labels/names are not technical labels, but rather are names

meaningful to system users.

**16.** Assure that command buttons are all of similar size and shape, and same font &

font size.

**17.** Assure that each command button can be accessed via a hot key combination.

**18.** Assure that command buttons in the same window/dialog box do not have

duplicate hot keys.

**19.** Assure that each window/dialog box has a clearly marked default value (command

button, or other object) which is invoked when the Enter key is pressed - and NOT

the Cancel or Close button

**20.** Assure that focus is set to an object/button which makes sense according to the

function of the window/dialog box.

**21.** Assure that all option buttons (and radio buttons) names are not abbreviations.

**22.** Assure that option button names are not technical labels, but rather are names

meaningful to system users.

**23.** If hot keys are used to access option buttons, assure that duplicate hot keys do not

exist in the same window/dialog box.

**24.** Assure that option box names are not abbreviations.

**25.** Assure that option boxes, option buttons, and command buttons are logically grouped together in clearly demarcated areas “Group Box”

**26.** Assure that the Tab key sequence which traverses the screens does so in a logical

way.

**27.** Assure consistency of mouse actions across windows.

**28.** Assure that the color red is not used to highlight active objects (many individuals

are red-green color blind).

**29.** Assure that the user will have control of the desktop with respect to general color

and highlighting (the application should not dictate the desktop background

characteristics).

**30.** Assure that the screen/window does not have a cluttered appearance

**31.** Ctrl + F6 opens next tab within tabbed window

**32.** Shift + Ctrl + F6 opens previous tab within tabbed window

**33.** Tabbing will open next tab within tabbed window if on last field of current tab

**34.** Tabbing will go onto the 'Continue' button if on last field of last tab within tabbed

window

**35.** Tabbing will go onto the next editable field in the window

**36.** Banner style & size & display exact same as existing windows

**37.** If 8 or less options in a list box, display all options on open of list box - should be

no need to scroll

**38.** Errors on continue will cause user to be returned to the tab and the focus should

be on the field causing the error. (i.e the tab is opened, highlighting the field with

the error on it)

**39.** Pressing continue while on the first tab of a tabbed window (assuming all fields

filled correctly) will not open all the tabs.

**40.** On open of tab focus will be on first editable field

**41.** All fonts to be the same

**42.** Alt+F4 will close the tabbed window and return you to main screen or previous

screen (as appropriate), generating "changes will be lost" message if necessary.

**43.** Microhelp text for every enabled field & button

**44.** Ensure all fields are disabled in read-only mode

**45.** Progress messages on load of tabbed screens

**46.** Return operates continue

**47.** If retrieve on load of tabbed window fails window should not open

**Specific Field Tests**

**Date Field Checks**

Assure that leap years are validated correctly & do not cause errors/miscalculations

Assure that month code 00 and 13 are validated correctly & do not cause

errors/miscalculations

Assure that 00 and 13 are reported as errors

Assure that day values 00 and 32 are validated correctly & do not cause

errors/miscalculations

Assure that Feb. 28, 29, 30 are validated correctly & do not cause errors/

miscalculations

Assure that Feb. 30 is reported as an error

Assure that century change is validated correctly & does not cause errors/

miscalculations

Assure that out of cycle dates are validated correctly & do not cause

errors/miscalculations

**Numeric Fields**

Assure that lowest and highest values are handled correctly

Assure that invalid values are logged and reported

Assure that valid values are handles by the correct procedure

Assure that numeric fields with a blank in position 1 are processed or reported as an

error

Assure that fields with a blank in the last position are processed or reported as an error

an error

Assure that both + and - values are correctly processed

Assure that division by zero does not occur

Include value zero in all calculations

Include at least one in-range value

Include maximum and minimum range values

Include out of range values above the maximum and below the minimum

Assure that upper and lower values in ranges are handled correctly

**Alpha Field Checks**

Use blank and non-blank data

Include lowest and highest values

Include invalid characters & symbols

Include valid characters

Include data items with first position blank

Include data items with last position blank

Functional Testing

# User states and desired on-boarding behavior

Action- Any page that links back to the sparkweave product should auto-detect whther the user trying to access content is registered or is the logged in user. If not registered, the user should be redirected to the registration page. If another user is logged in, that user should be logged out.

This behavior SHOULD be seen for the following cases:

1. Secure email read receipt, with personalized ON
2. Secure email read receipt, with generic ON
3. LFT download receipt, with personalized ON
4. LFT download receipt, with generic ON
5. Secure mail , with personalized ON
6. LFT email , with personalized ON
7. Secure email a/c creation prompt, with generic ON
8. LFT a/c creation prompt, with generic ON

The above behavior should **NOT** be seen for the following cases:

1. Secure email, with generic ON

2. LFT email, with generic ON

3. A/c confirmation, with personalized ON

4. A/c confirmation, with generic ON

The following actions need to be checked with the corresponding user states mentioned-

# ACTION = user attempts to see email content through a personalized link

#### STATE = non-existent

Show a simple page with an error message: “The user <username> could not be found on the system.”

#### STATE = unregistered

Show the registration page. Then if confirmations are off, show the content; else if confirmations are on, show the user a message telling them to check their email for an account creation confirmation.

#### STATE = unconfirmed

Show a simple page with an error message: “*Sorry, your account is not activated. In order to confirm your identity, you must open the “Account Confirmation” email sent to* [*foo@bar.com*](mailto:foo@bar.com) *and click the “Confirm” link. I lost my confirmation email; send me another.”*

#### STATE = active

If the user is logged in, show them the content. If not, show them the login page until the log in and then show them the content.

#### STATE = locked

Show a simple page with an error message: “This account has been locked down by someone – probably you. If this account should be unlocked, contact an administrator.”

# ACTION = user attempts to see email content through a generic link

If generic links are being used, then we cannot tell which user is clicking on the link form their inbox, we can only tell which user is currently logged in. If no user is currently logged in, the person who clicked the link should be redirected to the login page in which case the “login attempt” action scenarios can be followed. The user should be redirected to the desired content after they log in assuming they have permission.

If a user *is* logged in, the result will differ based on whether or not the logged in account was a sender or recipient of the given email. If they are (if they have permission), then the email content will be rendered. If they do not have permission, then the email content will not be rendered. They will be shown a “You do not have permission to this content” message instead.

#### STATE = non-existent

See above

#### STATE = unregistered

See above

#### STATE = unconfirmed

See above

#### STATE = active

See above

#### STATE = locked

See above

# ACTION = Login attempt w/ wrong password

A login attempt with an incorrect password should never reveal information about the users that do or do not exist on the system. For example, if someone attempts to log in as an unregistered, unconfirmed, or locked user, the system should not reveal through an error message that the user is unregistered, unconfirmed, or locked out unless the person making the login attempt can actually prove they are the user they claim they are (either by typing in the password, using a private link, or having access to the user’s email account).

#### STATE = non-existent

Show the error message “Invalid Credentials”.

#### STATE = unregistered

Show the error message “Invalid Credentials”.

#### STATE = unconfirmed

Show the error message “Invalid Credentials”.

#### STATE = active

Show the error message “Invalid Credentials”.

#### STATE = locked

Show the error message “Invalid Credentials”.

# ACTION = Login attempt w/ correct password

#### STATE = non-existent

This action is impossible in this state. There is no correct password.

#### STATE = unregistered

This action is impossible in this state. There is no correct password.

#### STATE = unconfirmed

Show a simple page with an error message: “*Sorry, your account is not activated. In order to confirm your identity, you must open the “Account Confirmation” email sent to* [*foo@bar.com*](mailto:foo@bar.com) *and click the “Confirm” link. I lost my confirmation email; send me another.”*

#### STATE = active

Forward the user to their homepage (inbox or admin homepage – either way)

#### STATE = locked

Show them an error message “This account has been locked down by someone – probably you. If this account should be unlocked, contact an administrator.”

# ACTION = “forgot my password” pressed

#### STATE = non-existent

Right now the system tells the user in the browser that a user by the given username cannot be found on the system. This behavior is acceptable for now since most other sites I tested behaved this way, but in the future, we should probably change this to email the typed-in email addr whether or not we find it on the system – that way people will not be able to gather data about accounts that are not associated with one of their email addresses.

#### STATE = unregistered

User is sent a reset password link.

#### STATE = unconfirmed

User is sent a reset password link.

#### STATE = active

User is sent a reset password link.

#### STATE = locked

User is sent a reset password link.

# ACTION = “reset password” email link pressed.

#### STATE = non-existent

Show a simple page with an error message: “The user <username> could not be found on the system.”

#### STATE = unregistered

Unregistered users should not be able to get password reset emails, but they may have an email in their inbox from before their account was deleted and then re-created through by being on-boarded. If so, when they press it, it should take them to the registration page with a warning message explaining why.

#### STATE = unconfirmed

The user will not be shown the reset password page; they will be show the error message: “*Sorry, your account is not activated. In order to confirm your identity, you must open the “Account Confirmation” email sent to* [*foo@bar.com*](mailto:foo@bar.com) *and click the “Confirm” link. I lost my confirmation email; send me another.”*

#### STATE = active

After they reset their password, they will be logged in and redirected to their homepage.

#### STATE = locked

The user will not be shown the reset password page; they will be show the error message: “This account has been locked down by someone – probably you. If this account should be unlocked, contact an administrator.”

# ACTION = “Help! – I didn’t create this account” pressed to lock account

#### STATE = non-existent

Show a simple page with an error message: “The user <username> could not be found on the system.”

STATE = unregistered  
Unregistered users should not be able to get account confirmation emails, but they may have an email in their inbox from before their account was deleted and then re-created through by being onboarded. If a user clicks this link, even though the user should not be able to have it, the account should transition to the “locked” state with an appropriate success message.

STATE = unconfirmed

The account should transition to the “locked” state with an appropriate success message.

#### STATE = active

The account should transition to the “locked” state with an appropriate success message.

#### STATE = locked

The account should remain in the locked state; they will be show the warning message: “This account has already been locked.”

# ACTION = “Confirm – Create Account” pressed

#### STATE = non-existent

Show a simple page with an error message: “The user <username> could not be found on the system.”

#### STATE = unregistered

An informational message saying “You cannot confirm the activation of your account before you register” should be shown. The registration page should be shown and the registration process should be followed in the normal manner.

STATE = unconfirmed  
A success message telling them that their account is now active should be shown on the login page where their email address is pre-populated.

#### STATE = active

An informational message should appear telling the user that their account is already activated – shown on the login page with the username prepopulated if the user is logged out, else on the user’s homepage.

#### STATE = locked

The account should remain in the locked state; they will be show the error message: “This account has been locked down by someone – probably you. If this account should be unlocked, contact an administrator.”

# ACTION = “Register Now” link pressed

#### STATE = non-existent

Show a simple page with an error message: “The user <username> could not be found on the system.”

#### STATE = unregistered

The user should be taken to the registration page and allowed to register.

#### STATE = unconfirmed

The user should be shown the message “*This account has already been registered and now needs to be activated. In order to confirm your identity, you must open the “Account Confirmation” email sent to* [*foo@bar.com*](mailto:foo@bar.com) *and click the “Confirm” link. I lost my confirmation email; send me another.”*

#### STATE = active

The user should be shown the message “*This account has already been registered” on their homepage if logged in and the login page otherwise.*

#### STATE = locked

The user will not be shown the registration page; they will be show the error message: “This account has been locked down by someone – probably you. If this account should be unlocked, contact an administrator.”

Time zones

Time zones can be set /modified at the following locations in our product-

1. Admin interface- domain and guest user settings

2. Admin interface- dates & time settings

3. End –user interface –account settings page

1. The default time zone for a new user (domain or guest) should match the time zone saved in the local user templates. If the local users template has “—use system default—“ saved , then the time zone saved in the admin interface under date & time should be used.
2. Once a user is created, he can adjust the time zone according to his preference on the account settings page and this will override all other higher levels of time zone settings in the product.
3. Time zone set on the local users’ template page takes higher priority than the admin-dates & time setting page.
4. The date and time stamp needs to be tested once a time zone has been saved for all incoming/outgoing mails.