# Revisions

|  |  |
| --- | --- |
| MOCKUPS FILE | WHAT’S NEW |
| AgentUI-WIP-1.zip | Self-contained Agent UI |

# Workflows

## User activates agent for first time

1. When agent installation completes, user is prompted (with link) to go to agent UI
2. If no admin user exists on the agent yet, the UI shows the activation.html page
3. User enters username and password to create an admin user
4. User enters activation key for the agent  
   OPTIONAL: User can click on the “info” icon, and an explanation + link will appear, prompting user to create an agent in the cloud web app in order to get an activation key
5. User clicks “Activate Agent” button
6. UI validates that all fields are filled, and the passwords entered in both inputs match.
7. Agent creates admin user on itself
8. UI logs in as newly created admin user, and shows the config.html page
9. UI waits for agent to complete registration with cloud (based on activation key), and shows results of registration with cloud in “Cloud Connection” section of config.html page
10. UI indicates in “VX Connection” section of config.html page that a password must be set for the “cloud\_agent” user in VX
11. User clicks on the “edit” icon in the “password” field
12. UI shows editVxSettingsModal.html. All inputs show current values. Agent should ship with default values, such as ipAddress:port = 127.0.0.1:443, pollInterval = 60 secs, but \*no\* default password. If a user clears one of the inputs with a default value, then on blur out of the input, it can take its default value again.
13. User enters password.  
    OPTIONAL: User can click “Test Settings” button to test the current values in the inputs before committing to saving them. Modal will then hide .enable-editing div and show the .show-test-results div. When user clicks “OK” button on test results, the .enable-editing div is shown again and the .show-test-results div is hidden.
14. User clicks “Save” button.
15. UI validates that the password input is not blank. (All other inputs should have been filled with default values if user had cleared them.)
16. Agent saves VX connection settings.
17. UI goes back to config.html page, with info refreshed. There is no longer an “edit” icon in the “password” field; bullet characters show that a password has been saved.

## User updates activation key

1. User goes to agent UI with browser
2. If an admin user exists on the agent, the UI shows the login.html page
3. User enters admin credentials and clicks “Log In” button
4. UI logs in and shows the config.html page
5. User clicks “edit” icon in upper right of “Cloud Connection” section
6. UI shows editActivationKeyModal.html
7. User enters new activation key  
   OPTIONAL: User can click “Test Key” button to test the key before committing to saving it. Modal will then hide .enable-editing div and show the .show-test-results div. When user clicks “OK” button on test results, the .enable-editing div is shown again and the .show-test-results div is hidden.
8. User clicks “Save” button.
9. UI validates that the activation key text area is not blank.
10. Agent saves new activation key.
11. UI goes back to config.html page, with info refreshed.

## User updates VX connection settings

1. User goes to agent UI with browser
2. If an admin user exists on the agent, the UI shows the login.html page
3. User enters admin credentials and clicks “Log In” button
4. UI logs in and shows the config.html page
5. User clicks “edit” icon in upper right of “VX Connection” section
6. UI shows editVxSettingsModal.html
7. User edits desired fields  
   OPTIONAL: User can click “Test Settings” button to test the current values of the inputs before committing to saving them. Modal will then hide .enable-editing div and show the .show-test-results div. When user clicks “OK” button on test results, the .enable-editing div is shown again and the .show-test-results div is hidden.
8. User clicks “Save” button.
9. UI validates that the password input is not blank. (All other inputs should have been filled with default values if user had cleared them.)
10. Agent saves VX connection settings.
11. UI goes back to config.html page, with info refreshed.

# Notes for Specific Pages/Modals

## activation.html

A couple divs are currently commented out:

Show div .activation-key-info-text when .activation-key-info-icon is clicked. It can either be a toggle show/hide for subsequent clicks, in which case the tooltip for the icon should also toggle between title=”Click for more info” and “Hide info”. Or, I’m OK with just once the extra info is shown, it can’t be hidden again.

Div .caption.validation-error contains what I think the two possible errors would be. It can be uncommented to see how the error messages should look if displayed. Also, if a validation error occurs, the border-color of the relevant inputs should change to @alertOrange.

## login.html

Should basically work same as login page for cloud web app. (Div .caption.login-error currently commented out.)

## config.html

In order to keep connection statuses up-to-date on this page, can UI just poll the agent periodically? Like maybe every 10 or 15 secs?

Not sure how fancy to get yet with “Connection Status” messages for Cloud and VX connections.

To start, how about at least:

- green icon + “Last connection” + timestamp  
- red icon + “Last attempt failed” + timestamp  
... plus the initial red icon + “Password required” when there is no VX password set yet.

## editActivationKeyModal.html

Div .caption.validation-error is currently commented out. It can be uncommented to see how the error message should look if displayed.

Div .show-test-results is currently commented out. As previously discussed, having just one modal that toggles between showing .enable-editing and .show-test-results is fine. If we don’t get around to implementing this test capability yet, that is OK. List of possible test result messages is probably TBD with Matt.

Could start with:

- green icon + “Connection successful”  
- red icon + “Connection failed”

... If we start seeing common reasons for failure, we can try to get fancier about diagnostic messages. Note, if connection failed, the “Account > Tenant” and “Agent Name” fields will just be blank.  
The ability to test a key before committing to saving it is relatively low priority at the moment.

## editVxSettingsModal.html

Div .caption.validation-error is currently commented out. It can be uncommented to see how the error message should look if displayed. For all other inputs except password, default values can be filled in if the input is blank when focus leaves the input, so I’m thinking we should not have to anticipate error messages for any of those.

Div .show-test-results is currently commented out. As previously discussed, having just one modal that toggles between showing .enable-editing and .show-test-results is fine. If we don’t get around to implementing this test capability yet, that is OK. List of possible test result messages is probably TBD with Matt.

For ipAddress:port, we could start with:

- green icon + “VxCore found at” + ipAddress:port

- red icon + “VxCore not found at” + ipAddress:port

And for login:

- green icon + “Login successful”

- red icon + “Could not test” (if no VxCore was found)

- red icon + “Authentication failed”

... If we start seeing common reasons for failures, we can try to get fancier about diagnostic messages.

The ability to test settings before committing to saving them is relatively low priority at the moment.

# General UI Patterns

## Modals

For simplicity and ease of plugging into the Angular app, modals are represented as their own pages in the “Clickable Mockups” included in this zip. In the actual UI, they should obviously be displayed as overlaid on the page from which the user reached the modal.

Clicking outside of the dialog box should not dismiss it. User must click a button in the dialog.

## Loading Spinner

For processes that may take a while (> 1s), such as the agent registering with the cloud or testing connection settings, a loading spinner should be displayed.

There should be a 500ms delay before showing the spinner, to avoid a flicker of the spinner if the process completes relatively quickly.

## Tooltips on Icons

Almost all icons should have a title= on them to clarify their meaning – e.g., “Click for more info”, “Successful”, “Failed”.