User Experience

1 Reviewing Content

1.1 Details Being Addressed

There are usability guidelines called heuristics that are the rules of thumb for User Interface Design. There are 10 heuristics established by Jakob Nielsen. This document will review the various aspects of Crossroads that do not follow the heuristics appropriately. Along with my heuristic evaluation, I will also address the design principles, and then conduct a cognitive walkthrough of the product.

By following these heuristics and principles, we are presenting the user with standards that have been established over the years. The more we follow standards the less our users need to learn how to use the product.

Training is often necessary for a user to be able to use such a complex product like Crossroads, but if it is possible to minimize the amount of training it improves the overall usability of a product immensely. The more a user needs to learn, the more likely they will forget and need to reference directions and this takes time. If we can eliminate one or two seconds it takes for a user to do a specific action, for multiple use cases we minimize the cost (almost always viewed as time) of that action. We all know, time is money!

1.2 Cognitive Psychology

Cognitive Psychology covers a user’s perception, attention, and most importantly memory, which encompasses learning as well. The perception of a user is said to be the UI of the brain. What the user perceives before the information reaches their brain. If the user perceives that a button will not be of use to them, they will not bother clicking on the button and wasting time. In contrast, if a user cannot perceive what a specific button’s functionality is, then they may either opt to: not try to use the button (potentially missing out on functionality that may be more useful to them) or try out the button (potentially wasting time, if it is not what they are looking for). By improving usability, we adjust the perception user’s gain of the products functionality.

All user’s focus on what they believe is relevant to their task, you cannot reach a user’s attention without going through their senses, such as perception.

As for memory, we must focus on the use of recognition rather than recall and where we require the user to use this. Since this is a system that users must be trained to use, we also want to focus on the encoding aspect. We want to facilitate/reduce encoding. This is done through colors, flagging, and position.

1.3 Things to Remember about Usability

- If there is a chance a user can make a mistake or be confused, assume they will.

- Prioritize the design principles: Visibility of user's actions/options; Feedback; Constrain (make certain errors impossible); Consistency; Affordance (make clear how you can things with an object);
- A single application does not create a standard. Multiple applications, programs and software with common practices, over the course of years, create standards. (Operating systems do not apply here. They are an oligopoly, and the use cases of an operating system are vastly different than a web application.)
- If there is one method to complete a task, that can potentially be done incorrectly, provide an alternative method. (Ex: how do you save a word document? At least 3 different ways.)

1.4 Help & Documentation

Heuristic: Help and documentation

There are various areas within the application that do not have clear instructions on what a button does or how to perform an action. Even if users are trained, they still need time to learn.

A truly usable application is one where a user needs no help, or training. The UI teaches the user for you. This is hard to do, even the most user friendly products occasionally need a dialog, or tutorial to teach the user how to do a specific task. In order to make a more usable product, we must reduce the learning curve for the user by following real-world examples and standards that users are most familiar with. We must also help the user's complete tasks in the most efficient way by presenting them with information that should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Suggestion: Provide instructions for ambiguous widgets or child widget actions.

1.5 Assumptions of the Users

- Users are being trained to use this product (helps with more complex use cases)
- Age ranges from early 20's to late 60's
- Computer skills range from moderate-expert
- Frequency of use ranges from fairly often-always
- One user per account

2 General Layout Suggestions

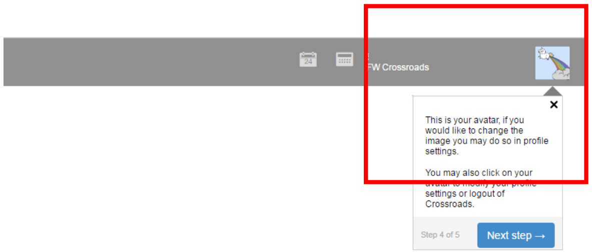
Suggestion: The overall design should deviate from window-like widgets.

Reasoning: In an operating system, you click on an application and a window opens. The contents of that window is the sole purpose as to why the user clicked on the application. The focus is this window. In the context of Cross Roads, and the average use case, the user is encouraged to use various widgets. The widgets load on top of each other covering potentially valuable/important information that the user may be using or want to use along with the current widget they had open. This creates frustration on the user's side, because now the user must take the time to reorganize the information on their screen in order for them to finish the task they are doing. This creates an additional cost for the user.

2.1 The Tutorial

Heuristic: Help and documentation; Aesthetic and minimalist design;

The tutorial, is quite wordy and does not encompass an entire use case of the product. This mini-tutorial only goes over very basic components that are mostly like not to be forgotten after initial training, since it is the bases of the entire product.

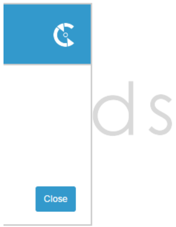


This is your avatar, if you would like to change the image you may do so in profile settings.

You may also click on your avatar to modify your profile settings or logout of Crossroads.

Step 4 of 5

It being broken down into steps is misleading to the user because, this gives the perception of the tutorial going over the steps in a task.

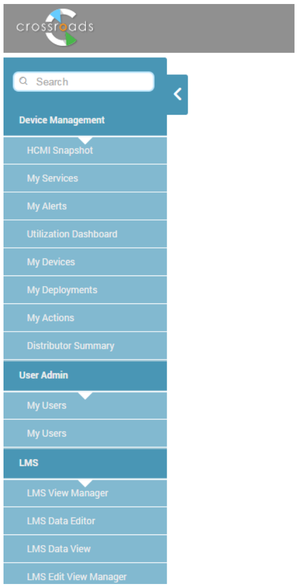


The tour dialog also shows a checkbox to uncheck if you want the tour to not show again upon start up. The tour is referred to a window. This is not a window, this is a dialog box. A window can be moved, and minimized.

Suggestion: Remove this tutorial (or only display during first use of product, after that, have a link where the user can access the tutorial if necessary) and apply tutorials similar to this for use cases in which the user is more likely to forget the workflow.

2.2 Menu

Heuristic: Aesthetic and minimalist design



This view provides information that is no longer necessary to the user based on their current focus.

By leaving the categories open, it extends the length of the menu and does not follow the heuristic of: "Flexibility and efficiency of use". Since this forces the user to scroll through all the previous categories.

Suggestion: Collapse previously opened categories when the user clicks on a different category.

Keep: The search feature. This does a great job at using recognition rather than recall and can be valuable and a time saver for the user's that have many widgets.

2.3 Top Navigation

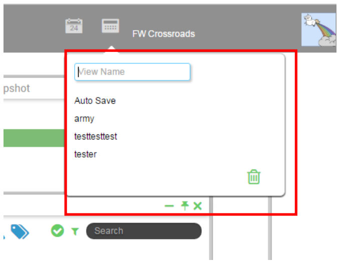
2.3.1 Views

Heuristic: Error prevention; Help and documentation; Flexibility and efficiency of use (follows);

This view is not intuitive enough. The placeholder: “View Name” does not provide enough clarity to the user. If the user was confused about this feature or could not remember what it was for, how could they find the information without deviating or increase the cost of this action?

The use of the “Auto Save” is not clearly documented.

Suggestion: Add a label to this section; Make the use case of it more clear. (See Char’s layout suggestion for this)



2.3.2 Profile Picture

Heuristic: Aesthetic and minimalist design

Keeping in mind the functionality of the product and what the users are able to do within the application. The presence of a profile picture does not provide any help/functionality to the user, other than being where the profile settings and logout is located. Visually, it clutters up the layout of the application.

The space being used can be eliminated or be used for a different purpose.

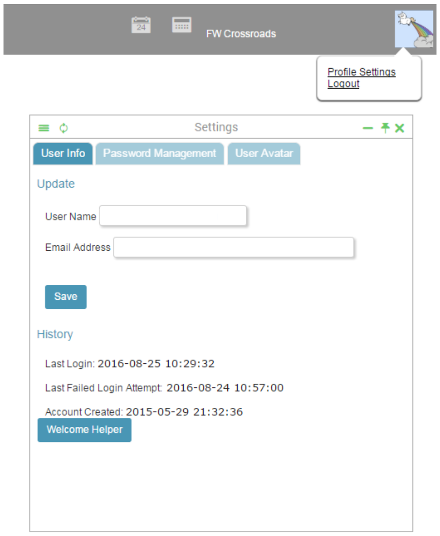
Profile pictures are used for quick recognition of the user, humanizing the workplace and putting faces to names. But, within the scope of this application, there is no use case for a profile picture. There are no discussion boards, no community, or anywhere the user would post content thus, not providing any sort of social business discovery.

2.3.3 Settings

Heuristic: Consistency; Error prevention; Flexibility and efficiency of use; Aesthetic and minimalist design; Visibility of system status;

The header for the settings widget reads “Settings” but the link that opens this widget is titled “Profile Settings”.

There is a distinction between these two titles. One directly relates to the user and the other relates to the application as a whole. By providing the user with a link for profile settings, they expect to see settings specific to the user rather than the entire application. If a user is look for the overall settings of the application, they may see the Profile Settings but not click on it because, to the user, that’s not where the general settings would most likely be.



Last Login: 2016-08-25 10:29:32

Last Failed Login Attempt: 2016-08-24 10:57:00

Account Created: 2015-05-29 21:32:36

Suggestion: Go with a more general settings title. It gives more flexibility to adding more to the settings page later on if necessary.

Under the section “History”, there is a “Welcome Helper” button. This buttons does not relate to the title it is under. This button is meant to show you the initial tour of Crossroads shown upon start up, but this tour is categorized under history. The title given to this tour is not clear enough what it is. Earlier it referred to as a tour, yet it contains steps that a tutorial would contain, not a tour and then later the Welcome Helper. The naming convention for this feature is not consistent and causes confusion to the user. This button is also available on all the setting tabs. It has not been properly categorized and causes confusion to the user as to why that piece of information is presented there.

Suggestion: Rename the tour button and find a single, properly categorized location for it.

Under “User Avatar” the only way for the user to be able to use their own profile picture is by dragging and dropping. By only letting the user perform this action one way, you limit the user. It is not always convenient or efficient for the user to drag and drop.

Suggestion: Provide the user with an alternative for uploading an avatar. Such as a simple upload. This will also help in the development of the mobile interface. Drag and drop functionality for uploading pictures cannot be done on a mobile device.

The password management tab, provides a way to change your password. The new password confirmation does not give the instant feedback on whether or not their passwords matched.

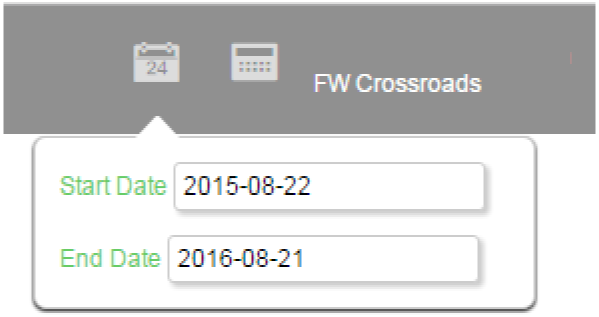
The refresh button on this button, serves no purpose. It is best to remove anything that is not being used, otherwise, it causes confusion and/or unnecessary use of real-estate that could otherwise be used differently.

Suggestion: Remove refresh button. Provide proper feedback on password management tab.

2.3.4 Calendar

Heuristics: Help and documentation; Recognition rather than recall;

There is no clear directions or labeling as to what this calendar feature is for, or what it does. If there is some external form to get the instructions or information, that is extra time the user has to take to figure out what this feature is for. That is a significant amount of cost to the user and they cannot assess whether the cost is worth the benefit since there is no clear explanation readily available.



Start Date 2015-08-22

End Date 2016-08-21

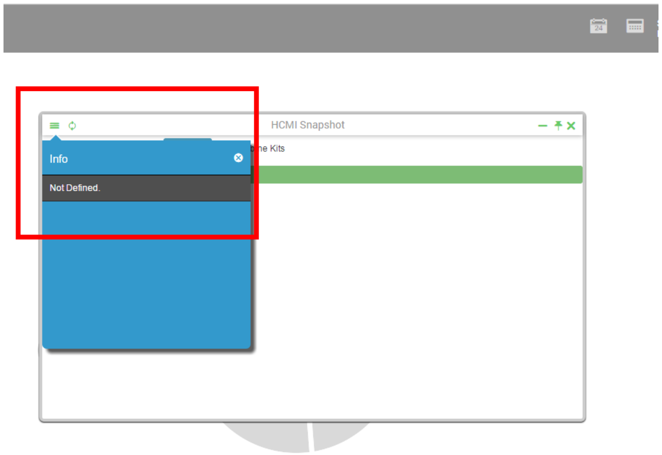
Suggestion: Provide a more information, on how to use the feature, which is readily available to the user.

2.4 Widgets

2.4.1 Widget Information

Heuristic: Match between system and the real world; Consistency;

Suggestion: Change the information icon to an 'i' icon instead. This follows a standard and exercises the design principle of affordance. Affordance provides strong clues to the operation of things. Using a standard icon for information will allow the user to perceive that they will be receiving information when this icon is clicked. Following standards, the hamburger icon (icon used for info now) is used for menus.



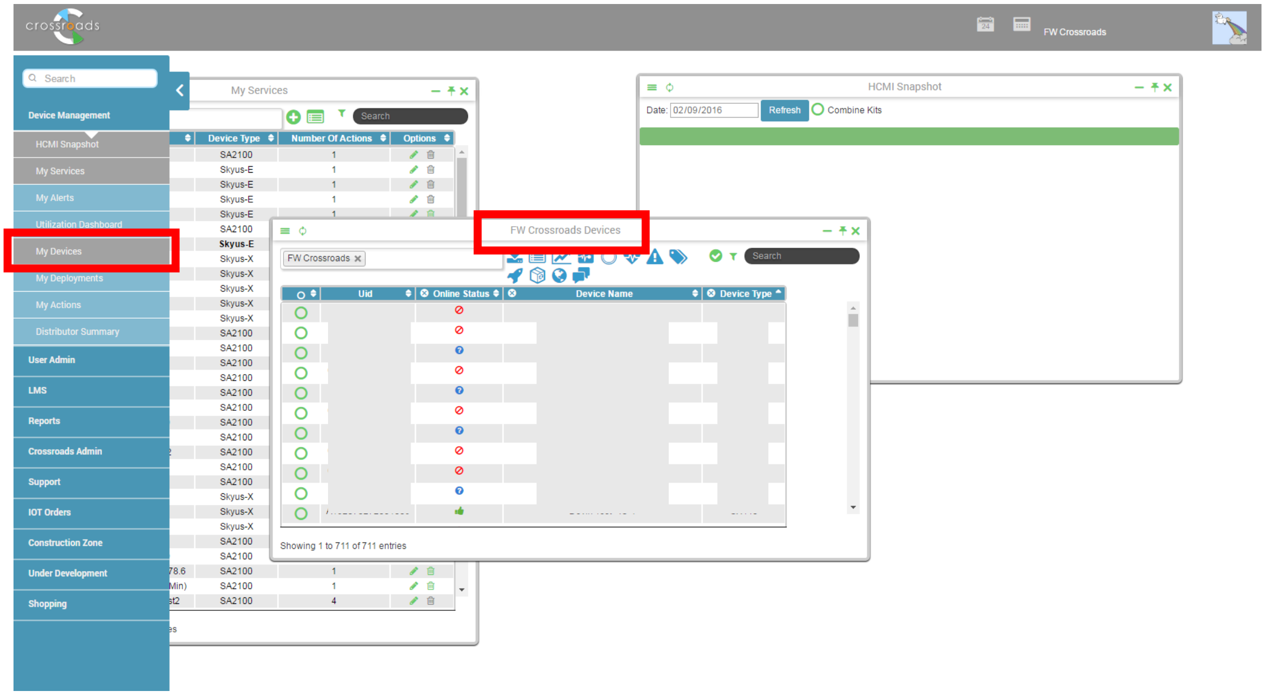
Not Defined.

This feature has no information and is unnecessary for this widget's view. If there is no use for the icon it is best to remove it, as to not confuse the user and clutter the interface.

2.4.2 "My Devices"

Heuristic: Recognition rather than recall**;** Consistency and standards;

Suggestions: The widget name in the menu and the header of the widget should match. User's must know at all times, where they are. By changing the name of the widget window, you force the user to use recall rather than recognition to remember what they had clicked on, and potentially misleading the user to search the wrong name, if it is a widget they are do not use often.



Date: 02/09/2016

	Device Type	Number Of Actions	Options
	SA2100	1	
	Skyus-E	1	
	Skyus-E	1	
	Skyus-E	1	
	Skyus-E	1	
	SA2100		
	Skyus-E		
	Skyus-X		
	Skyus-X		
	Skyus-X		
	Skyus-X		
	Skyus-X		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	SA2100		
	Skyus-X		
	Skyus-X		
	Skyus-X		
	SA2100		
	SA2100		
78.6	SA2100	1	
Min)	SA2100	1	
st2	SA2100	4	

○	Uid	Online Status	Device Name	Device Type
○		⊘		
○		⊘		
○		?		
○		⊘		
○		?		
○		⊘		
○		?		
○		⊘		
○		⊘		
○		?		
○		👍		

Showing 1 to 711 of 711 entries

2.4.3 Repeated Icons

Heuristics: Consistency and standards**;** User control and freedom;

By providing two icons with different functionality, you are encoding the icons functionality as one thing and then changing it in the eyes of the user. Even though these icons are labeled, a user may still not remember which one of the two displays they want. This means that, at most, the user would need to hover over both icons to find which one they need. Increasing the cost of the task. This is an instance in which the cost can be minimize.

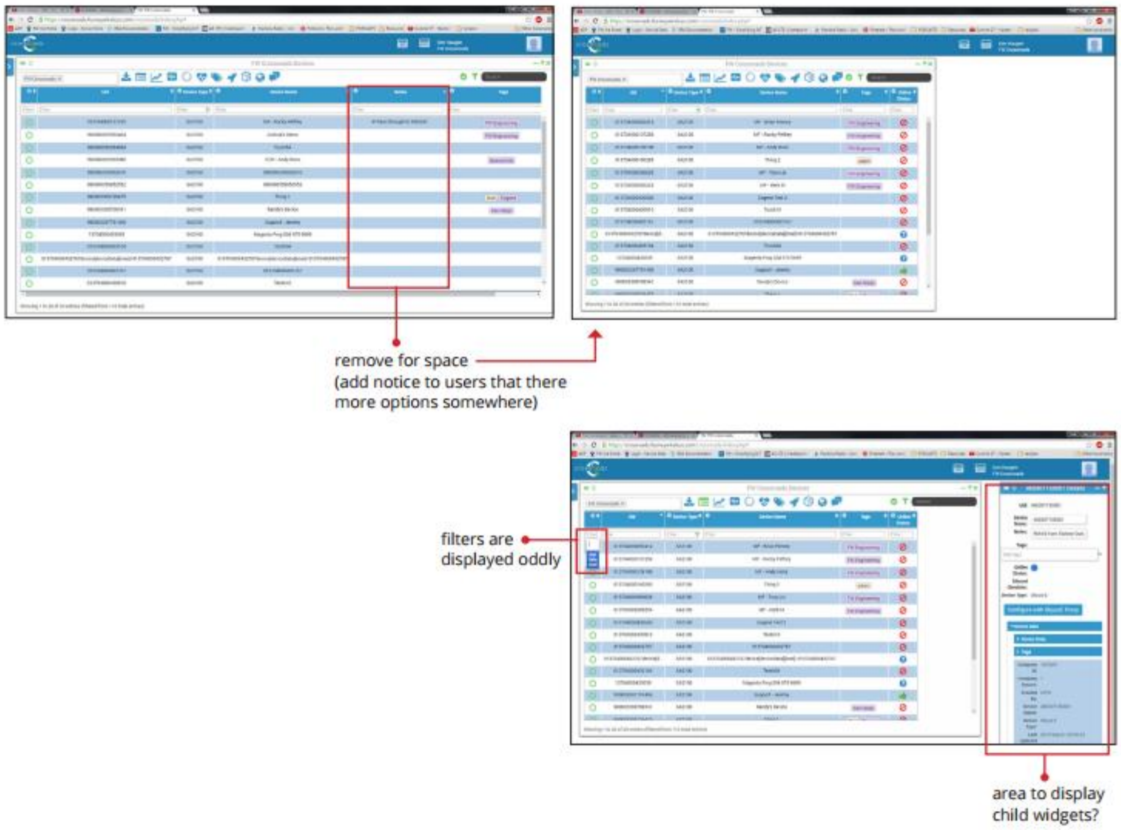
2.4.4 Crossroads Devices

Heuristic: Aesthetic and minimalist design

By providing extra columns such as the one in the screen shot, we are displaying information that is potentially irrelevant/rarely needed. This clutters the interface and increases the time the user takes to find the information they are looking for. User's eyes would still need to pan over the columns they don't need, to get to the information they want.

Appropriate visibility of filter options are cruicial to the usability of this feature.

crossroads WIDGETS - crossroads devices



remove for space (add notice to users that there more options somewhere)

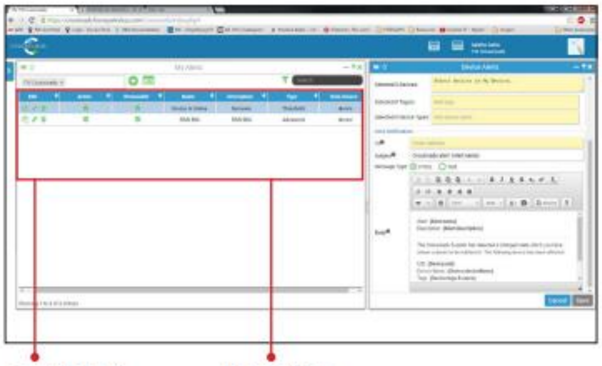
filters are displayed oddly

area to display child widgets?

Suggestions:

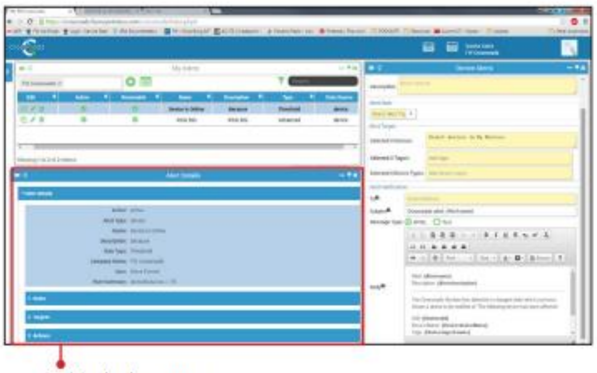
Provide the user with a default view in widgets that contain only the necessary columns and icons needed for the basic use case of that widget.

Simultaneously (excluding icons) still providing the user the ability to add more columns and making this availability of more columns clear and presenting them additional column options with appropriate descriptions of the data each column provides.



copy and edit
do not w ork

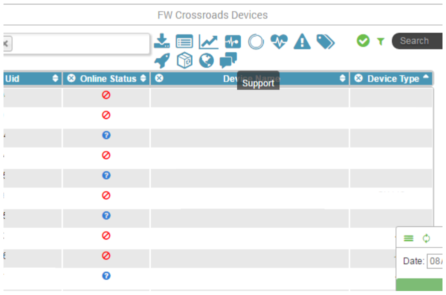
analyze table
do wo need all these rows?
data source?



is this the best way to
display this?

2.4.5 Hovering Tip Labels

With hovering labels, the user does not immediately know what a button does. This action now has a higher cost in order for a user to be able to know which button they want to use (the payoff). Icons are used by many applications and some of the more seldom used icons will have different meanings over a multitude of different applications. This means that by providing no static label, the user does not immediately know what each icon means nor the potential options they have. Having to hover over all the icons in order to see all their options, increasing the cost further.



Suggestion: Remove all hover tips labels from application. Hover tips should be used for additional, helpful information to the user. But not for essential information.

2.4.6 General Labels

Heuristic: Consistency and standards

There are many graphs and child widgets that can be generated and many of them do not have static labels within the content they display. The lack of static labels deprives the user from knowing their options/information and the ability to learn how to complete their task within the application faster, creating a larger learning curve.

Suggestion: Provide appropriate labels within the content generated. In order to better inform the user of what they are viewing.

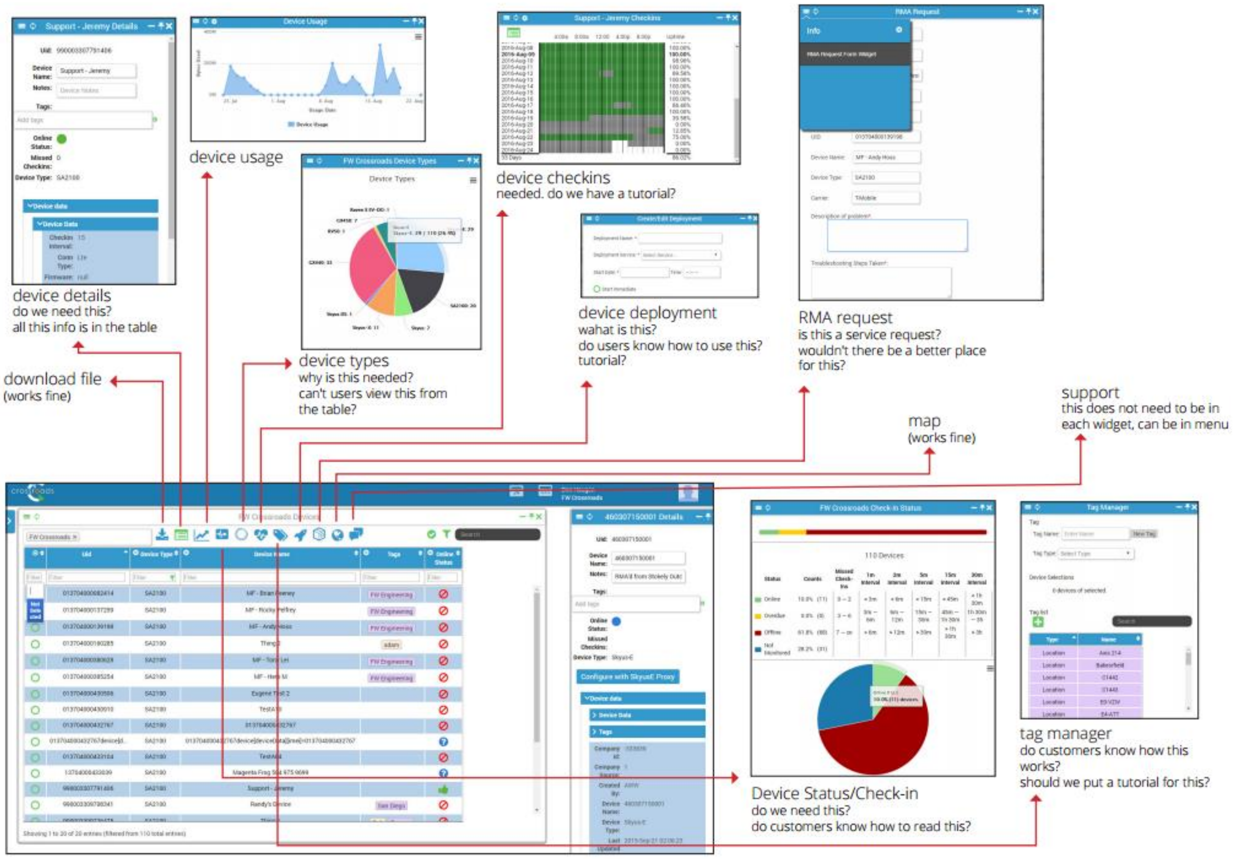
2.4.7 Support Icon

Heuristic: Aesthetic and minimalist design

This icon is displayed on almost every widget. But, the information is the same. Since the icon is in each specific widget, this leads the user to assume that this button is specific to the widget they are using and that it will provide information specific to it.

The support icon currently used is commonly perceived as a chat icon. This is also misleading to the user because they expect to be able to immediately chat with someone about an issue, rather than have to call.

Suggestion: Remove support icons from all widgets. Place support information in a more general location. Re-organize/display differently information presented. Such as bolding the information that is of most interest to the user.



device details
do we need this?
all this info is in the table

device usage

device types
why is this needed?
can't users view this from
the table?

device checkins
needed. do we have a tutorial?

device deployment
wahat is this?
do users know how to use this?
tutorial?

RMA request
is this a service request?
wouldn't there be a better place
for this?

download file
(works fine)

map
(works fine)

support
this does not need to be in
each widget, can be in menu

tag manager
do customers know how this
works?
should we put a tutorial for this?

Device Status/Check-in
do we need this?
do customers know how to read this?

3 Jakob Nielsen's Heuristics for User Interface Design

Visibility of system status (Feedback)

> *The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.*

Match between system and the real world (Metaphor)

> *The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.*

User control and freedom (Navigation)

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards (Consistency)

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention (Prevention)

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall (Memory)

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use (Efficiency)

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design (Design)

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors (Recovery)

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation (Help)

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

4 DESIGN PRINCIPLES

- Visibility of user’s options/actions
- Feedback

- Constrain
    - Make certain errors impossible
        - Ex: Menu’s vs typing to prevent syntax errors.
- Consistency
- Affordance
    - Make clear *how* you can do things with an object.