



UX CHECKLIST FOR WALLET DEVELOPERS

We believe that improving the user experience for wallets will be a huge step in making cryptocurrencies more usable, as users need to interact with a wallet in order to store, send, and use any digital currency.

We have compiled a checklist of good practices that can be applied to any cryptocurrency wallet with a graphic user interface. If you're building a wallet that supports Zcash, we encourage that you use this checklist to catch common usability problems before launch or user testing.

USER INTERACTION

Progressive disclosure:

Things should progress from simple to complex; only the necessary or requested information is displayed at a given time. This helps manage complexity without becoming confused, frustrated, or disoriented.

Feedback:

Every action should produce a visible, understandable, and immediate reaction. Failing to acknowledge an interaction can lead to unnecessary repetition of actions or errors (i.e. clicking "send" multiple times).

Priming:

Tell people what they can expect and what they should do. For example, explaining that a camera is needed to scan QR codes before you ask for camera permissions is likely to have users who want that feature to accept it.

Communication:

Be context-aware of what the user is doing and the nature of the message. For instance, notify of events like transaction confirmations with push notifications, since they're probably not waiting on the app for the confirmation.

Error handling:

The best way to handle errors is to prevent them. But if one occurs, put next to the relevant input field (not just at the top or bottom of the screen) to show users what they need to fix without searching for it. It should describe what happened, why it happened, suggest a fix, and not blame the user.

USER INTERFACE

Hierarchy:

Information is presented in order of importance and the visual hierarchy of actions on a screen matches what the user expects to do first, second, third, etc.

Simplification:

Limit the choices that a user is presented with per screen. Provide appropriate filters if there is a large data set.

Consistency:

Components with a similar behavior should have a similar appearance. For example, all buttons that send a transaction should be blue, square, and labeled 'send.'

Predictability:

Set good expectations. From looking at your interface, with no previous use, users should be able to answer things such as "where am I?," "what can I do here?," "where can I go from here?," and "what does this button do?."

Visibility:

Discoverability shouldn't involve luck or chance. If a page requires scrolling, hint that more content is below the screen by showing half of an image. If there are some screens you want users to find, the menu that links to those pages persists everywhere.

NAVIGATION

Persistence.

The navigation bar should always be visible on every screen. If it isn't, users don't know what to do next or don't know how to do the next thing.

Uniformity:

Similarly styled navigational elements should behave similarly. Additionally, elements of navigation should never appear and disappear, rearrange in order, or move to a different location.

Method:

Choose the method that most easily lets the users find what they want. This is specific to the use case. Method include browsing via a navigation system, searching with keywords, or filtering to narrow large lists.

Sorting:

Alphabetical sorting is avoided unless necessitated by many navigational choices (7+). Sort by relevance, related groups, or anything else instead.

Labeling:

Use meaningful labels and icons for navigation menu items, links, and buttons. Don't force people to chase information they need.

VISUAL DESIGN

Alignment:

Every element in the UI should be aligned with one or more other elements. Alignment provides cognitive stability and creates visual relationships. In this same vein, left-align large blocks of text as users need to expend more energy to track the lines. Eyes fatigue faster, comprehension slows, but the users may not be aware why.

Proximity:

Group certain elements (navigation, header, articles, etc.) contextually to form a perceived whole. For the same reason, visually separate unrelated items.

Repetition:

Use repetition to create a hierarchy of visual styles. This principle applies to fonts but also colors, textures, and graphical elements. (For instance, all titles should be of one size, all buttons are square, all colors are in a color palette, etc.) Reusing elements of visual styles in visual elements creates cohesiveness.

Contrast:

Text is easily readable when stark, complementary colors are used. A lack of contrast between text and background strains the eyes because they don't know which color to focus on.

CONTENT

Market information:

Provide an up-to-date crypto to FIAT currency conversion, along with current exchange rates between cryptocurrencies.

Network information:

Tell users if their transaction is likely to be processed quickly or not, based on mempool congestion.

Fee information:

Show how much the fee is, what % of the transaction it is, and if it's added on top or included.

Simplify jargon:

Translate what a concept or event affects the user, rather than exposing or explaining what it is technically. For instance, say if the transaction has been confirmed or not, instead showing the number of confirmations or how many confirmations is considered safe.

Let's Talk!

<https://chat.zcashcommunity.com>

Get updates!

[@zcashco](#)