

Matthew Hughes
2340 Spring 2014
April 6, 2014

UI Evaluation Report

Team 7's application, Mint, is for the most part very naturally set up. Our cognitive walkthrough has shown the application to be fairly easy to pick up and start using. Some of the user interface, especially that attached to the most recently added functionality, could be fairly confusing and potentially misleading to a new and unobservant user of the application.

In presenting our findings, we will move through the different pieces of the application in the order that a user would probably discover them.

Upon opening the app, the user is presented with a bare-bones welcome screen, with two self-explanatory buttons "Login" and "Register." A user who has never used the application before would probably choose to register for the service before their first login (perhaps based on experience with other systems), so this is a reasonable design.

The first issue we have identified is a discrepancy between the registration and the login process. Suppose a user wanted to register with the service using the email "user@somewhere.com" and the password "luv". According to the internal rules of the application, this email address is acceptable, while the password is too short to be valid. However, the registration goes through, taking us through to the login screen. Our wayward user only realizes the depth of her mistake when she attempts to login using these

credentials, which responds with a pointer to the password field, reading “This password is too short”.

Our hypothetical user may return to the Registration page by navigating backwards using the back button (familiar to them from a few minutes using some other application of an Android phone). On the registration page, the fields they had filled in are still filled, so our user changes her password to the slightly longer “lucu” in both password fields, to see if that will work. This time when she attempts to hit the “Register New Account” button, she gets just a Toast popup message reading “User not created - please try again”. The user quits the application in disgust, having forbidden herself from using her own email address to register a usable account.

This believability story is somewhat contrived. A user is unlikely to attempt to register using a password fewer than four characters. As such, we wouldn’t consider it a high priority to fix this issue.

Another issue we have identified is with the dashboard screen, which is presented to the user after a successful login. Suppose a newly registered and logged in user wants to create and view an account. Supposing they create an account successfully, and return to their dashboard where the list of accounts is displayed. Intending to view the first account listed, our unfortunate user accidentally presses the button directly above that entry, labeled “Delete All Accounts”. Their accounts are all immediately removed.

This circumstance is fairly plausible. Users tend not to be the most careful with their fingers when manipulating their phones (and one of our evaluators

actually did do this accidentally while navigating the application). It is a much higher priority that this be fixed.

A final issue: Suppose the user wants to withdraw some amount from a preexisting account and check their remaining balance. She navigates to the account from the dashboard, and selects the account she wants to withdraw from. She selects “Make New Transaction”. She puts in “10” for the amount (although she wanted \$10.50). For the “Type” field she inputs the word “withdrawal”, since she is withdrawing money. After making this withdrawal, the application presents her with a transaction history for the account. Since she wants to see her balance, she hits the back twice to get to the Account summary page. The information there has not been updated since last she saw it, which she notices and navigates back to dashboard and then hits the account again. She notices that the account balance has increased by \$10.00, rather than decreased.

There are two important suggestions we draw from this example. Firstly, the user should not be required to type “withdraw” to make a withdrawal. Using a typing field may lead the user into thinking that any number of words might be used, when really only one will give them their desired effect.

Additionally, there is an issue of relevant feedback. When making a transaction, the user is given ample indication that their action was successful, but that is very probably not the only information they would want to know.