
This material formed part of the original submission and has been peer reviewed. We post it as supplied by the authors.
Figure 1.

Understanding Navigation Options in Mobile Apps and the Mobile Web

Compiled by: Todd Kasenberg

Additional Resource for Review: http://www.youtube.com/watch?v=ODx8u32lq9A

Introduction

Top and left navigations are typical on large screens and therefore websites, but lack of screen real estate on small screens makes for an interesting challenge for mobile website rendering and app implementation. Mobile navigation must strike a balance between quick access to information and unobtrusiveness. What follows are examples of navigation for mobile apps and the mobile web.

Top Navigation

One option is to render the menu and top level navigation items across the top of the mobile screen, in a manner directly analogous to conventionally accepted navigation at normal websites.

However, this can present with several problems, not the least of which is that where there are many top-level navigation items, valuable screen real estate can be consumed. (This is inconsistent with emerging best practices for mobile design, which suggest that users prefer a content up front approach from a user experience perspective.) As well, it is not uncommon for items to render differently across different mobile devices – so while a line of menu items might look perfectly placed and constrained to a single line on a Galaxy Nexus phone (an Android phone), it might not be as “perfect” on the smaller screen of the iPhone 4.

The Menu Button

A second and simple option is to tuck away from ongoing visibility the main menu, and to rely on the user clicking a button to gain access to next menu options. This can place a range of small buttons across the top (if the number of menu items is smallish), or can float a menu item list down on top of the current screen view or even push the current screen view “down”.

Rev Panam Salud Publica 35(5/6), 2014
This approach can require a bit of coding effort, and may not perform consistently across application platforms (e.g., it may work better on iPhones than Android phones). However, it can be perceived as very classy, and when executed properly, can optimize screen real estate on small devices. Because this button is omnipresent throughout the app, it reduces the need for back buttons or other significant navigation indicators.

The Dropdown Selector

Another menu approach which really is a variation of the menu button is to use a drop-down list box to present navigation options. This nicely transforms to a list of links on activation, and saves precious screen real estate.

This approach, easily recognizable, can allow the phone to serve up native controls, which can be a boon for some devices. However, it really doesn’t offer much styling control, and the management of submenu items, typically through indenting or use of dashes, can make things a bit confusing for the user.

The Left Nav Flyout

This navigational approach has been popularized by Facebook, and usually involves either pressing a button to have a menu list emerge from the left of the screen, or actively swiping across the screen with the finger from left to right to access the menu.
Again, this method provides lots of space to use for content, minimizing onscreen navigation elements. It is sophisticated, and there is a growing awareness of the technique because of the penetration of Facebook. However, some users may be potentially confused by it, and there are some tricky aspects of dealing with a list of menu items too long for the typical lengths of screen afforded by mobile phones.

**Bottom Navigation**

This approach pays homage to the notion that the mobile user wants lots of content on the screen, and provides navigation options through a series of menus, links or a bar that is accessible from the bottom of the screen. This approach may fix the navigation bar to the bottom of the screen, but as likely can require navigation to the bottom of the screen (screen scrolling) to access the menu items.

This approach typically frees up header space. However, depending on its implementation, it can be hard to access because it may require substantial scrolling for the user.
**Gallery**

This approach arrays a series of icons across the main screen, and relies typically on the use of a “back” button, either in a fixed upper bar on the screen, or relying, in the case of Android devices, on the fixed back button. Often, the gallery icons are organized along a theme, and are sideways scrollable.

The main advantage of this method is that a great deal of options may be displayed to the user and accessed rapidly. However, users may not enjoy lateral scrolling, users might be confused by the categories used for organization, and there is a lot of drilling down then coming back to re-access the menu for alternative options. As well, this method is typically quite visual, which relies on universal recognition of the images or icons used.

**Springboard**

The so-called springboard navigation provides a “home screen” which demonstrates a series of icons that serve as springboards to more content.

This method may enable remarkable personalization of the app through allowing the user to configure the order and identities of the icons. However, it is typically a content-limiting approach, and users must invariably back up to this screen to access next options, which is labour-intensive.
Tab Menu

This menu system places a band of iconized buttons either at top or bottom of screen.

This method provides the stability of onscreen menus. However, unless icons used are universal, there can be some confusion on the part of the user. Also, added complexity typically arises when there are more than 5 menu items to depict through a tab menu.

List Menu

A list menu presents the user with a list of items which may be accessed through simple touch. These menus can be activated from a small button at the top of the screen, or may be the primary method of navigation when users will be drawn through lists with sublists which also have sublists.

This method is very conventional, and allows full use of descriptive text for the pages to which each menu item links. However, it must also be considered a content-reducing strategy, since it leaves very little room on the screen for any other content.

MegaMenu

This menu approach is an adaptation of a more recently emerged approach on the web, which incorporates as much of site structure into large-ish menu lists for the purposes of reducing click burden through a website. Typically, a first level set of categories is always available, and when one of the items is selected, a list of the full next options will be presented below the top level navigation items.
This method can be of value when there is a large amount of content that should be easily accessible. However, it does quickly consume screen real estate, even when intelligently implemented so that the second and lower-levels of the navigation scheme “push down” screen content below the menu listing.

**Metaphor Menu**

This approach, unconventional in many respects, leverages imagery to create clickable hot zones which serve as navigation elements.

This approach can create visual interest, and help people understand the flexibility of content navigation through a familiar visual paradigm. However, this method often consumes entire screens of real estate to achieve the visual impact desired, and commonly requires app users to use a Back navigation button to return to the navigation scheme from a drill-down screen.
FIGURE 2

Symptoms

radio buttons

Physical
- Fever
- Rash
- Joint pain
- Pain

Other:
- [ ] Insert here

Emotional
- Depression
- Stress
- Mood
- Overall self-esteem

Like that the 'body' only pops up if you meet a certain criteria
Herschman et al. • Development of a smartphone app for adolescents with lupus

SUPPLEMENTARY MATERIAL

TEXT:

Touch all areas where you have [pain] and hit "done" when you are finished.

A subsequent pop-up prompts for each symptom/area color-coded on previous page.

And when says "touch all areas where you have rash" will also popup "touch to take photo of rash."
Herschman et al. • Development of a smartphone app for adolescents with lupus

SUPPLEMENTARY MATERIAL

Text:
Touch all areas where you have pain and
not "done" when you are finished.

D subsequent pop-up prompts for each
symptom area are continued on
previous page.

↓ and when says "touch all areas where you have rash" it will
also pop-up "touch here" picture camera to add/ take
photo of rash.

Like body

Have magnifying glass to move around body to location
Herschman et al. • Development of a smartphone app for adolescents with lupus

SUPPLEMENTARY MATERIAL

Symptom Tracker

For each of the symptoms, rate the severity:

- Pain
- Fever
- Fatigue

Select the appropriate sliding scales based on the chosen number of symptoms.
Like the visual appeal of pinched

Cool wheel!

CREATIVE!

I like the homepage

Great!
Are you sure
Yes [ ] No [ ]

Back
End
Records
For
Research

Add
New
- Cancel

Notes

Date
Time
Event
is twice recurring?

Reminders
1 Day
2 Day
other

SHARE
contact-list

Send Email
Send SMS
 Lup may

Practical
I like the email.
med: optim!
FIGURE 4

only 6 icons max!
Like that home screen is not too busy.
Supplementary Material: Herschman et al. • Development of a smartphone app for adolescents with lupus

Diagram showing a smartphone interface with the heading "Social" and icons labeled "Discussion," "Message Wall," "Internet," and "Facebook." A heart-shaped note is placed below the phone with the text "Leave the past idea."
Love
The wall!

Great idea!

Social

Love the 'wall' idea.