

Designing and Building Applications for the IPAD



Dr. Paul Dorsey
Dulcian, Inc.
www.dulcian.com

IPAD vs. Desktop

- ◆ Great demand for systems to work on IPADs and tablets.
- ◆ Building IPAD applications is a different challenge from building for desktop computers.
- ◆ User experience on a touch screen requires many modifications.



Decisions

- ◆ Build for the browser?
- ◆ Create a native application?
- ◆ Create a hybrid?
- ◆ Depending upon the technology you choose, communication with the database can become another challenge.



Case Study

- ◆ Build a patient check-in application for doctors' offices to support:
 - Patients entering complete medical histories.
 - Patients verifying personal information.
 - Patients correcting errors.
 - Patients paying any fees or co-payments with an attached credit card swiper.



Challenges

- ◆ Users would vary widely in their levels of computer literacy.
 - Elderly patients with no previous exposure to computers at all.
- ◆ Never requiring the patient to ask questions of the doctors' office staff.
 - Very user-friendly
- ◆ Make application engaging and keep the patients' attention through completion



Tasks

- ◆ Use an application generator for the following reasons:
 - Need to support hundreds of different clients.
 - Avoid maintaining hundreds of hand-coded applications
 - Generate to desktops, IPADs, and mobile phones without rewriting application each time.
 - Need ability to make changes to UI without hundreds of changes to application.
 - Implement by a change to a CSS class.

Converting Existing Application

- ◆ Already had a generator to support a desktop application.
 - Cross-tested to ensure that the generated code would work for all major browsers and versions.
- ◆ Converting application to work on IPAD in Safari (not as a native application)
 - Simple? Right?



The Easy Part

- ◆ Everything “mostly” worked.
 - IPAD’s 1024 x 768 resolution is lower than that of most desktop monitors
 - Sufficient for the application.
 - Onscreen keyboard
 - Web widgets



Challenge #1

◆ Support for computer illiterate

- Supporting users who had never used a computer before.
- Supporting users not familiar with touch screens
- No experience with scrolling



Challenge #2

- ◆ Web-based applications have a different feel on tablets than applications designed for a desktop computer.
 - Screens are simpler, with fewer components.
 - Need to be friendly looking rather than efficient.
 - Limit to a single component per row
 - Provides space for help text to the right of the field
 - Sufficient room for a label to the left of the field.
 - The simpler the structure, the easier it is to make the generator work properly.

Challenge #3

◆ Life without a mouse

- Buttons must be large enough to accommodate large fingers.
- More space must be provided between items to prevent selection errors.
- Checkboxes, radio groups, and text boxes must all be redesigned. LOVs do not work at all unless huge fonts are used for the list items.



Alternatives

- ◆ Two alternatives:
 - 1. Use many screens
 - 2. Require scrolling on all screens.
- ◆ Entire user interface must be rethought
- ◆ Affects the entire user experience.



Prepare for the Bizarre

- ◆ Part of our application would not allow users to type a lower case "r".
 - Every other letter worked just fine (upper and lower case).
 - Problem existed in Chrome, but not in Internet Explorer.
 - Using F3 to do something in the application and "r" was being interpreted as F3 in Chrome
- ◆ You just have to love cross-browser compatibility!



Performance Issues

- ◆ Many well-justified complaints about IPAD JavaScript performance.
 - Sub-second performance suddenly required 10 seconds for many common actions.
 - Underpowered IPAD engine



Timing Tests

- ◆ An Acer Chrome book --- 10 seconds
- ◆ A desktop running Chrome --- 15 seconds
- ◆ IPAD Air -- 20 seconds
- ◆ IPAD 4 -- 40 seconds
- ◆ IPAD 2 -- never came back
 - Surprise that an inexpensive Acer Chromebook was much faster than a desktop.
 - Having a system specifically tuned to run a browser has its advantages.



Solving Performance Issues

- ◆ Transitioning a desktop application to the IPAD resulted in many unnecessary objects placed in the Document Object Model (DOM).
- ◆ IPADs are particularly sensitive to many objects in the DOM
- ◆ Tweaking application generator to remove objects markedly improved performance.



Other IPAD Issues

◆ IPAD Tap event

- Device waits 300ms to see second tap comes
- Acts as equivalent of a double mouse-click.
- Translating mouse-click event into a tap event causes a lag on the IPAD interface.
- Solution: Touch event substituted

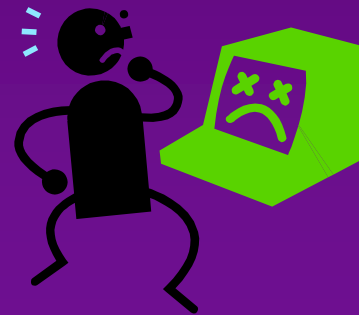


◆ Scrolling slow, jerky, sluggish

- Set `-webkit-overflow-scrolling :touch.` (Internally, refer to as the "goFaster=YES" setting).

Testing Troubles

- ◆ Without a PC browser tool, difficult to determine what actions worked quickly or slowly
- ◆ Breaking down the time lags was a challenge.
- ◆ Necessary to connect the IPAD to an Apple computer and use Mac Developer mode to track performance issues.



Caching Screens



- ◆ Cache screens on client and only refresh renderer at runtime.
 - When application started, all screens were cached resulting in a 10-second performance hit.
 - Page-to-page load time decreased by 50% and loading at approximately 1 second to go from page to page.
- ◆ If DOM is large enough, exceeds device memory and the browser abnormally terminates.

Locking it Down



- ◆ Needed single profile that would not allow patients to bypass the application in any way.
- ◆ Device would open and run the desired application without a security code required to unlock the device.
- ◆ nCLOSE software selected
 - Reasonably inexpensive
 - Easy to set up and configure
 - Met requirements



Which way is up?

- ◆ IPAD users accustomed to turning the device and having the screen rotate as needed.
 - Problematic for application being designed.
- ◆ Alternatives:
 - Turn off screen rotation feature
 - Make screens adapt to both orientations
- ◆ Decision made to turn off screen rotation and Zoom control to simplify application



Novice User Design Issues

- ◆ Forget about menus
- ◆ Use linear flow
- ◆ Test-test-test
- ◆ Nothing is obvious
 - “Ohhhhh, that big thing on the right is a credit card swiper???”
- ◆ Think video game
 - Introduce new skills one page at a time



Conclusions

- ◆ Making a web application work on an IPAD was a lot more work than what we expected.
- ◆ Ended up being possible to solve all the problems encountered
- ◆ Journey was neither as short nor as painless as predicted.



Contact Information

- ◆ Paul Dorsey – paul_dorsey@dulcian.com
- ◆ Website – www.dulcian.com
- ◆ Current published book: *Oracle PL/SQL for Dummies*
- ◆ Upcoming book: *Oracle PL/SQL Performance Tuning Tips & Techniques*, McGraw-Hill (Summer 2014)

