Going Mobile: From Concept to Delivery

Throughout this magazine you will no doubt see articles from leading experts in a whole range of technologies, using their years of experience to successfully implement solutions on the Oracle stack. Well in this article I want to tell a slightly different story: a story of how an applications DBA, rather than someone from a development background, managed to end up successfully designing and building a production on-device mobile application for Lloyd's Register. If you want to learn the lessons, and maybe start building your own mobile applications, then read on!

Richard Childe, Independent Oracle Applications DBA & ADF Developer

Why Mobile?

I've always thought that whilst the IT industry is quick to announce something as the "next big thing", the reality is that unless the people who hold the IT purse strings have also decided it's the next big thing, then it's going to count for little.

There are plenty of statistics out there to demonstrate the explosion in mobile app usage, such as the research showing that overall app usage grew 115% in 2013. And when you see the number of people at work with smartphones, (and hear the complaints that they can't access anything to do with their jobs on them), then it's clear that the demand for mobile is tangible, real, and it is here today.

Everybody wants mobile apps. People on the move want quick access to their work data on their shiny new tablet, not on an anti-virus laden corporate laptop that takes an age to start.

Working at Lloyd's Register (LR), I'd already dipped my toes in the water with Oracle ADF by re-developing a few of LR's public-facing web pages using Oracle ADF 11*g*. I'd learnt the framework mostly by reading Grant Ronald's "The Quick Start Guide to Fusion Development", and studied a few examples on the web. Some people are little bit wary of a perceived complexity of ADF, but once you find your way around it, you start to appreciate just how quickly and easily you can get an application up and running.

A Mobile Framework

So with only a little ADF knowledge my interest was piqued when ADF Mobile hit the market in late 2012. ADF Mobile is a hybrid mobile framework based on Java and HTML5 which can be written once but deployed to both iOS and Android devices. I decided to take ADF Mobile for a spin and see if I could get a simple piece of data out of a database and onto a phone. Somewhat surprisingly this proved to be very straight-forward. The enthusiastic response to this small achievement from Lloyd's Register management was encouraging. This simple demonstration showed that developing a mobile app could be fairly straightforward, did not have to involve a vast army of developers, could sit happily on an existing Oracle infrastructure, and would not entail the spiralling development costs that sometimes happens when a small, mobile development company comes unstuck when trying to deal with your IT infrastructure.

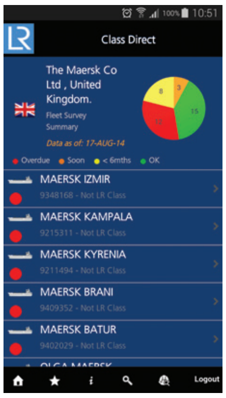
10:51
Class Direct
The Maersk Co Ltd , United Kingdom.
Fleet Survey Summary
Data as of: 17-AUG-14
● Overdue ● Soon ● < 6mths ● OK
MAERSK IZMIR
9348168 - Not LR Class
MAERSK KAMPALA
9215311 - Not LR Class
MAERSK KYRENIA
9211494 - Not LR Class
MAERSK BRANI
9409352 - Not LR Class
MAERSK BATUR
9402029 - Not LR Class

FIGURE 1: INITIAL PAGE DISPLAYING OVERALL SNAPSHOT OF SURVEY DATA FOR A CLIENT, PLUS VESSEL LIST AND SURVEY STATUS INDICATOR

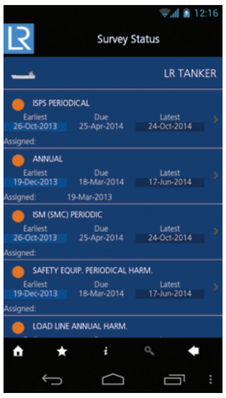
12:16
Survey Status
LR TANKER
ISPS PERIODICAL
Earliest 26-Oct-2013 Due 25-Apr-2014 Latest 24-Oct-2014
Assigned:
ANNUAL
Earliest 19-Dec-2013 Due 18-Mar-2014 Latest 17-Jun-2014
Assigned: 19-Mar-2013
ISM (SMC) PERIODIC
Earliest 26-Oct-2013 Due 25-Apr-2014 Latest 24-Oct-2014
Assigned:
SAFETY EQUIP. PERIODICAL HARM.
Earliest 19-Dec-2013 Due 18-Mar-2014 Latest 17-Jun-2014
Assigned:
LOAD LINE ANNUAL HARM.

FIGURE 2: VESSEL SURVEY CATEGORIES AND STATUS

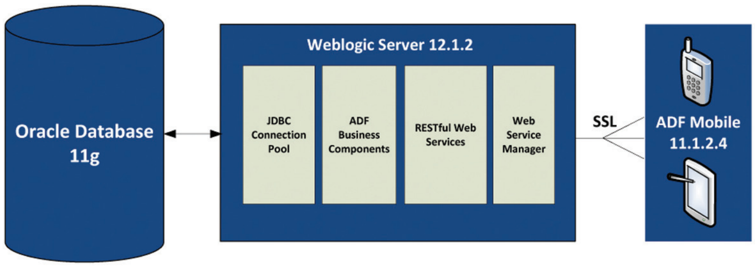

FIGURE 3: SIMPLIFIED ARCHITECTURE VIEW

The Mobile Application

We already had a definite idea of what we wanted for one particular production app: something that would allow clients of the company in the shipping industry to monitor the status of the inspection surveys which are regularly carried out on their vessels. This on-device application would be based on a sub-set of data already exposed through Lloyd's Register website. However if the same functionality was to be presented on the phone we had an immediate challenge - I couldn't see how I could realistically employ any sort of on-device caching strategy, as the data volumes to be searched were very large, specific to a particular user, and constantly changing. I took the decision that so long as the mobile app wasn't pulling gigabytes of data over the network, and the back-end was running database queries that took less than a second to complete, then it should be perfectly workable to simply request and deliver data to the mobile application as and when it was needed. For these web service calls, REST seemed to be the way to go due to its well-documented advantages for mobile access.

One immediate advantage of writing this mobile app is that we weren't committing to some horrendously complex application.

Mobile apps, by their nature, tend to be simple and easy to use. When planning an app, it's always worth remembering that the average time a person will spend using a mobile app is just over 1 minute, so I decided I probably shouldn't be looking at trying to shovel a re-write of E-Business Suite on to people's phones.

When it came to actually coding the app, I switched to using a Mac, as the iPhone simulator proved to be an essential item in the toolkit, allowing me to quickly view the fruits of my labours without loading it on to an actual phone.

However now and again I did need to see how it behaved on a physical device, so I deployed my web services into the integrated Weblogic server, and put the laptop on a WiFi network, so that a phone or tablet could talk to it. My app uses a number of backing beans, and using the debugger was a must, as it can talk to the simulator or a physical device and halt the execution at a breakpoint in the usual way. I made extensive use of JDeveloper's built in HTTP Analyser to test my web service calls, which are basically RESTful web services which hook into an ADF application module to obtain the data from ADF Business Components view objects. Each row being returned from the view object is loaded into a POJO, an array of which is returned from the web service call in REST/XML format. A simplified view of the architecture looks a little like Figure 3.

Branding and look and feel tend to be hot topics when it comes to mobile apps and luckily enough, LR were undergoing a major re-branding exercise during the development phase, so fortunately the in-house brand management team gave me plenty of ideas and direction on a look and feel. With ADF Mobile it's pretty easy to change the look and feel as I was using skinning, which allows you to create a style-sheet to be applied to all your mobile pages in the same way a website does, rather than burdening each page with styling directives. Determining how the users would navigate their way through the app was initially based on some very simple wire-framing of pages, together with a basic page flow diagram. The app is essentially one ADF task flow – a series of pages linked together with control flows. Mostly the design process evolved slowly over time, and in retrospect I learned the importance of nailing-down the whole UX from the start and not trying to deviate from it too much. The initial beta-testing we did with users made me re-think the design a little, and taught me that maybe I didn't get enough user-input from the outset.

Lesson learned. Know your audience! Take a look through Oracle's "Fusion Applications User Experience Patterns and Guidelines" website for some very good guidance on getting this right from the off.

If you're thinking of developing an app with Oracle ADF Mobile, here's what you (and your team if you're lucky enough to have one), need to arm yourself with:

Key skills:
ADF Mobile *– Learn how to build a small app. There are plenty of good videos out there from the Oracle development team to get you started with Mobile.*
ADF BC *- For database queries, learn how to build a simple application module with a view object to query some data.*
Java *- Basic syntax and structure. Look for examples on how to code a RESTful web service and access your ADF BC components, and learn how to use the debugger and the HTTP Analyser.*
SQL Tuning *- Again, if you are going to be running database queries, your queries need to finish pronto. You may get away with a SELECT statement taking 5 seconds to complete in an OLTP system, but not in a mobile app. Learn to read an explain plan. Or ask your friendly DBA to do it for you. (Good luck with that).*

And remember that materialised views are your friend.

The tools you will need:
An Apple Mac or Macbook *- worth it for the iPhone simulator alone. All the Android emulators I tried were as much use as a chocolate teapot.* ***Oracle JDeveloper*** *&* ***Lots of coffee****.*

General Tips:
Speak to your users and ask them exactly they want out of the app. Make sure they'll want to download it and use it on a regular basis. Keep them engaged during the development process to make sure everything is heading in the right direction. Make the app intuitive to use, and easy to navigate. It shouldn't require any sort of user manual. All this might seem obvious but it's important not to lose sight of it. ***Good luck!***

When it came to some initial beta testing, we used TestFlight (https://testflightapp.com) to allow us to distribute the app to a selected user base under controlled conditions.

We supplied the download details to our users along with a questionnaire concerning the app's usefulness and usability, the results of which allowed us to make some pertinent changes.

This was made easier by the Agile approach we'd taken with this project, which seems particularly suited to mobile app development. We were able to make regular presentations of the app to the business stakeholders. You don't need the full, functioning, finished application to demonstrate something meaningful to them. And the iPhone simulator, along with some devices handed around during the meetings, made an ideal way to present the work I'd done so far.

When the development work was complete, it was time to add security. That meant securing the web-services with Oracle Web Services Manager (OWSM), and SSL enabling the web service traffic. We placed a load balancer in front of our Weblogic servers and terminated SSL at that point, which allows us to spread the load across multiple servers, and provides for a much simpler URL naming scheme and SSL certificate setup to facilitate any future applications sited in the same infrastructure.

Deploying on the Apple iStore and Google Play

Once I'd put the supporting production database and application server infrastructure in place, it was time to make the app available on Apple's iStore and Google's Play. You'll see a lot of bad press on the web around how picky Apple can be with allowing apps to go on the iStore, but actually, we found the process to be fairly stress free. The ADF Mobile framework meets all the stipulated technical requirements, so unless you've shoe-horned something into your app that the Apple reviewers aren't happy with, you should find the submission process goes smoothly. Make sure your accompanying documentation is up to scratch, and make sure the description of the app accurately reflects what it does.

Looking to the future, there's a number of possibilities to take this further.

A surveyor carrying out a vessel survey may find it useful to have the vessel's current GPS location available. Using the device's camera to capture images of damaged or worn components and upload them to a database may also be a handy capability. All of these are capabilities the framework already supports, and I don't think it's an exaggeration to say that a well-written mobile app has the potential to make life much easier for people out in field and change the way they work. The possibilities are huge. Take a look around your workplace and have a think about how mobility can improve existing processes. ■

Sources:
Mobile-App Use Increased 115% in 2013 [http://mashable.com/2014/01/14/mobile-app-use-2013]
Oracle Fusion Applications User Experience Patterns and Guidelines [http://www.oracle.com/webfolder/ux/applications/fusiongps/mobile/index.htm]
Know Your Users Guideline [http://www.oracle.com/webfolder/ux/applications/fusiongps/mobile/index.htm]
Study: Average App Session Lasts About 1 Minute [http://readwrite.com/2012/01/17/study_average_app_session_lasts_about_1_minute#awesm=~ovcbifLYharQyn]

ABOUT THE AUTHOR

Richard Childe
Independent Oracle Applications DBA & ADF Developer

An independent Oracle Applications DBA and ADF Developer. Has worked with Oracle technologies for over 20 years, for a variety of companies including AT&T, AHL Trading & Lloyd's Register. To learn more about developing a mobile app with your existing Oracle stack, contact me through LinkedIn or talk to Nymad at www.nymad.co.uk (info@nymad.co.uk)