Using the Oracle Mobile Cloud Service Applications

Rapid Development Kit

to Build Oracle Mobile Applications User Experiences

Oracle Applications User Experience

ORACLE
APPLICATIONS
CLOUD

Oracle ® Using the Oracle Mobile Cloud Service Applications Rapid Development Kit to Build Oracle Mobile Applications User Experiences

Contents

Introduction

The Oracle Mobile Application Framework (MAF) is a hybrid mobile framework that enables developers to rapidly develop single-source applications that can be deployed to Apple iOS, Google Android, and Microsoft Windows platforms. Oracle MAF leverages Oracle Java, HTML5, and JavaScript to deliver a complete model-view-controller (MVC) framework with a declarative development experience and integration with native device services. Oracle MAF will appeal to those in an enterprise mobile application development environment.

This eBook supports the Oracle Mobile Cloud Service Applications Rapid Development Kit (RDK) and shows you how to use the RDK to build mobile applications for the Oracle Cloud using the same design patterns and guidelines that Oracle Applications teams use.

Who Should Read This eBook

This eBook is intended for Oracle partners, customers, and the wider Oracle MAF developer community who adapt, design, and build enterprise mobile applications.

Skills You Need

The RDK is not intended to be a learning tool for beginners. It assumes intermediate-level skills for developing mobile applications using Oracle MAF and Oracle JDeveloper.

Software You Need

To use the code in the RDK, you will need this software:

- Oracle JDeveloper 12.2.1.0.0, available on Oracle Technology Network (OTN).
- Oracle MAF 2.4.1 or later, available on OTN.
- For iOS development:
  - OS X 10.11.x or later
  - XCode 8.3.x or later
  - iOS 10.x or later
- See section 3.5 for other platform considerations.

How To Use This eBook

Use this eBook in conjunction with Oracle JDeveloper, the *Oracle Mobile Cloud Service Applications User Experience Design Patterns* eBook, Oracle MAF, and the RDK sample code. The accompanying design patterns eBook outlines the design patterns and guidelines used by Oracle teams to develop and build Oracle mobile applications.

You can use the code in the RDK either as a how-to reference, or you can copy parts of the code into your own application. Alternatively, for quick prototyping, you can write code directly into a copy of the RDK.

Platform Considerations

The initial version of the RDK sample code has been developed and tested exclusively on the iOS platform. While the sample application can be deployed to Android and Windows platforms, you may need to make certain style sheet adjustments, so that screen elements render correctly on these platforms.

Watch our Oracle Usable Apps page (http://tinyurl.com/paas4saas) for later versions of the RDK, which may be updated to include considerations for other platforms.

Support

The RDK contains sample code that is available to anyone who downloads it. No part of the RDK is supported by Oracle Support. The RDK sample code is bound by the UPL License (https://opensource.org/licenses/UPL) for Open Source software.

You might find these additional resources useful:

- *Oracle Mobile Cloud Service Applications User Experience Design Patterns:* http://tinyurl.com/OAUXMobileDesignPatterns
- Oracle MAF site: http://www.oracle.com/maf

- Oracle Applications User Experience blog: https://blogs.oracle.com/oaux/
- Bors, Luc. *Oracle Mobile Application Framework Developer Guide:* https://blogs.oracle.com/emeapartnerweblogic/entry/mobile_application_framework_developer_guide

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility program website at http://www.oracle.com/us/corporate/accessibility/index.html

About the Oracle Applications User Experience Team

User Experience is a strategic differentiator for Oracle Mobile Cloud Service Applications. The Oracle Applications User Experience (OAUX) team invests heavily in researching, designing and delivering a world-class user experience that delights, engages and empowers users of the Oracle Mobile Cloud Service Applications. The OAUX team continues to explore emerging technologies for ways to improve user experience. Learn more about OAUX outreach events and shared resources at http://www.oracle.com/usable-apps

Comments and Suggestions

As you use the resources in the RDK to build, extend, and integrate mobile applications, we'd like to hear about your experiences.

Tell us:

- Is there something you would like us to clarify?
- Would you like to see more user interface examples? Do you have an example use case?
- Do you have an idea for a component, design pattern or page type?
- What other information would help improve your development productivity?

Share your experiences with us on Twitter @oauxcloud or through the Oracle Applications User Experience blog: https://blogs.oracle.com/oaux/.

1. Rapid Development Kit Content

The Oracle Mobile Cloud Service Applications Rapid Development Kit (RDK) contains tools to help you develop mobile application user experiences that look good, and are intuitive and user-friendly. The RDK contains:

- Wireframe templates and samples
- Reusable code artifacts
- Sample product flows
- Alta UI CSS
- Images
- User experience design patterns eBook
- RDK technical eBook

These tools help support your mobile application design work and help accelerate the build phase, so that you can build the right thing and build the thing right.

In addition to using these tools, be sure that you work with users and other key stakeholders during the design and wireframing phases to determine the appropriate designs and design patterns for your use cases.

Applications Cloud MAF Kit Workspace

The RDK contains reusable code artifacts and a sample application in a single downloadable ZIP archive. Opening the packaged workspace in Oracle JDeveloper reveals the projects organization.

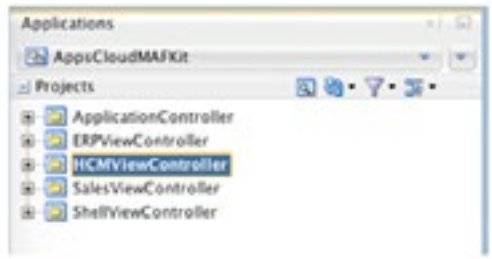

Figure 2-1. Sample projects in the application workspace

Oracle Sales Cloud, Oracle Human Capital Management (HCM) Cloud, and Oracle Enterprise Resource Planning (ERP) Cloud sample features are packaged into separate view controller projects deployed as feature archives. The shell view controller project pulls together these feature archives into a single mobile application that gets deployed on the target device.

Application Controller

The Application Controller project contains application-wide resources, such as the application life cycle listener, custom CSS, launch screen, login screen, and metadata that describes the Oracle MAF application configuration.

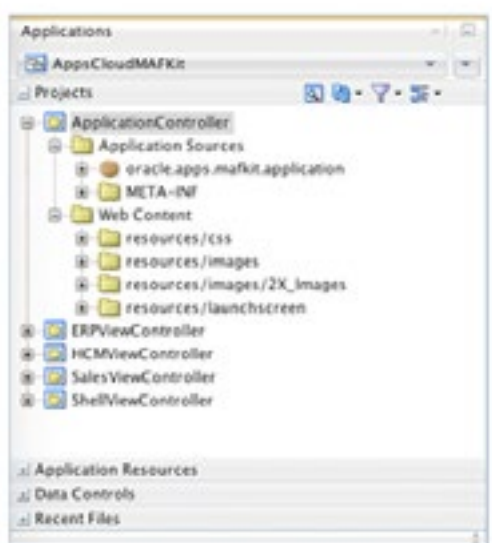

Figure 2-2. Application Controller contents

At the time of writing this eBook, image files in the Application Controller project cannot be directly referenced in the View Controller project, so they need to be physically copied into the View Controller project where they are consumed.

ERP View Controller

The ERP View Controller project contains sample features corresponding to Oracle ERP Cloud functionality, including Self-Service Procurement and Shipments.

Figure 2-3. Sample Oracle ERP Cloud features

The Procurement flow is driven using local data from an Oracle Java bean exposed as a data control. This flow demonstrates a multipart feature home page with search and shopping cart capabilities.

The Shipments flow has been intentionally left blank.

HCM View Controller

The HCM View Controller project contains sample features corresponding to Oracle HCM Cloud functionality, including Expenses, Hiring, and Time Cards.

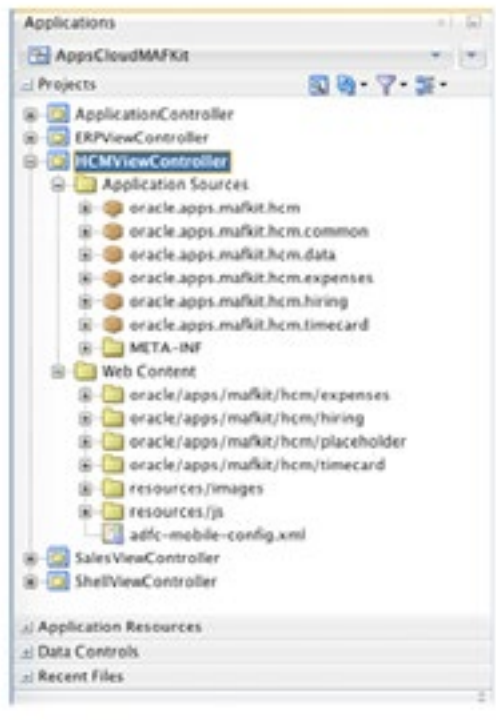

Figure 2-4. Sample Oracle HCM Cloud features

The Hiring flow is driven using local data from an Oracle Java bean exposed as a data control. This flow demonstrates a card view feature home page with context switching capabilities.

The Expenses and Time Cards flows have been intentionally left blank.

Sales View Controller

The Sales View Controller project contains sample features corresponding to Oracle Sales Cloud functionality, including Commissions and Opportunities.

Figure 2-5. Sample Oracle Sales Cloud features

The Opportunities flow is driven using local data from an Oracle Java bean exposed as a data control. This flow demonstrates a simple list view feature home page with transactional capabilities.

The Commissions flow has been intentionally left blank.

Shell View Controller

The Shell View Controller project presents the mobile application shell within which the application features are exposed. The sample shell includes the following capabilities:

- About page
- Home page
- Notifications page (local notification data only)
- Settings page
- Custom springboard

The shell project also includes a login feature, which is another mechanism to present application login. However, a custom login page would typically be packaged in the Application Controller project and automatically served up to the user when using a secured Oracle MAF application.

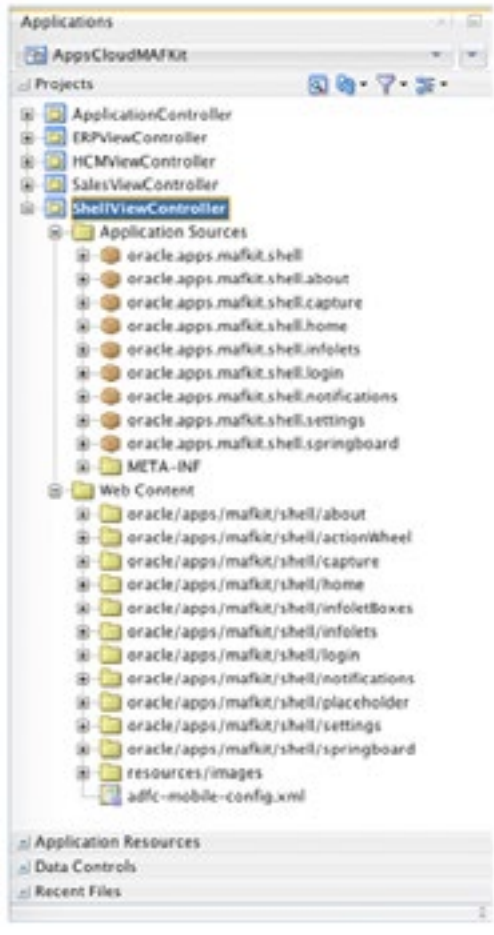

Figure 2-6. Sample Shell view controller

The Shell View Controller project also contains a new interaction paradigm called the action wheel, which is used to capture contextual data. You can interact with this feature in the sample application. For more information about the action wheel, see the design patterns eBook.

Running the Sample Application

To run the RDK sample application on an Apple Mac computer, make sure that you have the latest version of Xcode installed. Xcode will make the iOS Simulator available to Oracle JDeveloper. You can then use the iOS deployment profile defined on the sample application to deploy the application to the iOS Simulator.

To run the RDK sample application for Android, you will need the latest version of the Android SDK installed on your machine. You can then use the Android deployment profile defined on the sample application to deploy it to the Android emulator.

When deployed to the iOS simulator or Android emulator, the application will launch and display the login screen. To demonstrate the authentication user experience, the login page is driven by hard-coded user name and password values. (Note: This type of hard-coded authentication mechanism should not be used in any real-world application.)

Figure 2-7. Login page

Sign in with these credentials:

User Name: lisa.jones
Password: oracle

The home page is displayed. This page contains infolets, is scrollable, and can be viewed in both portrait and landscape orientations. You can tap and hold individual infolets to put them in drag-and-drop state. In this state, you can drag them to reorder the infolets on the page. The page displays a global header at the top that contains the feature name, a shuttle icon on the left, and a notifications icon on the right.

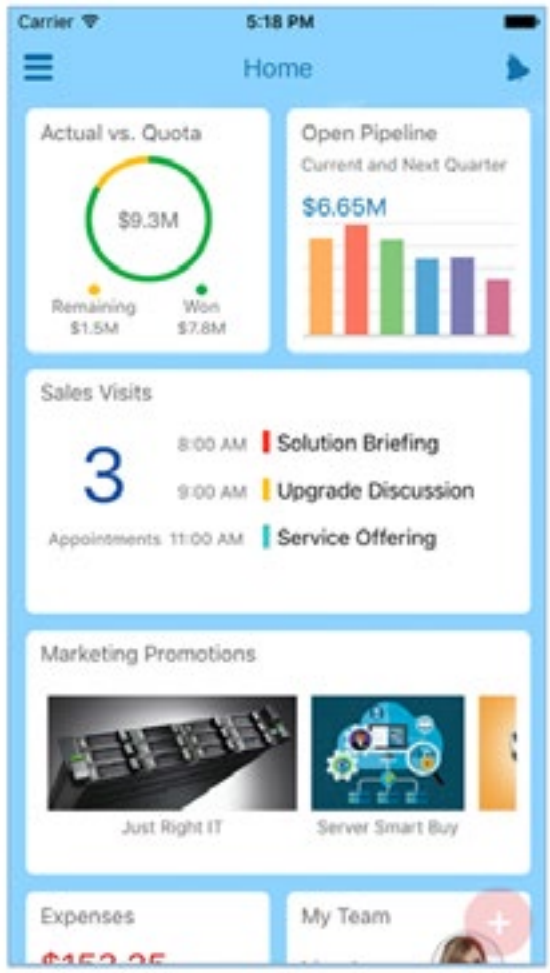

Figure 2-8. Home page displaying infolets

At the bottom right corner of the home page, the capture icon is displayed. Tap this icon to open the action wheel.

The action wheel provides quick access to contextual actions that are relevant to the page. The action wheel lets you capture images and notes, and then stores them as notifications for later retrieval and action.

Figure 2-9. Action wheel displaying camera and notes icons

The shuttle, which is implemented as a custom springboard in Oracle MAF, is the launch point for all of the features available in this application. The shuttle icon is displayed at the left top corner of the home page and on all work area landing pages.

Tapping the shuttle icon opens the shuttle, which displays a scrollable list of features available in this sample mobile application. Tapping an individual feature will navigate the application to that feature.

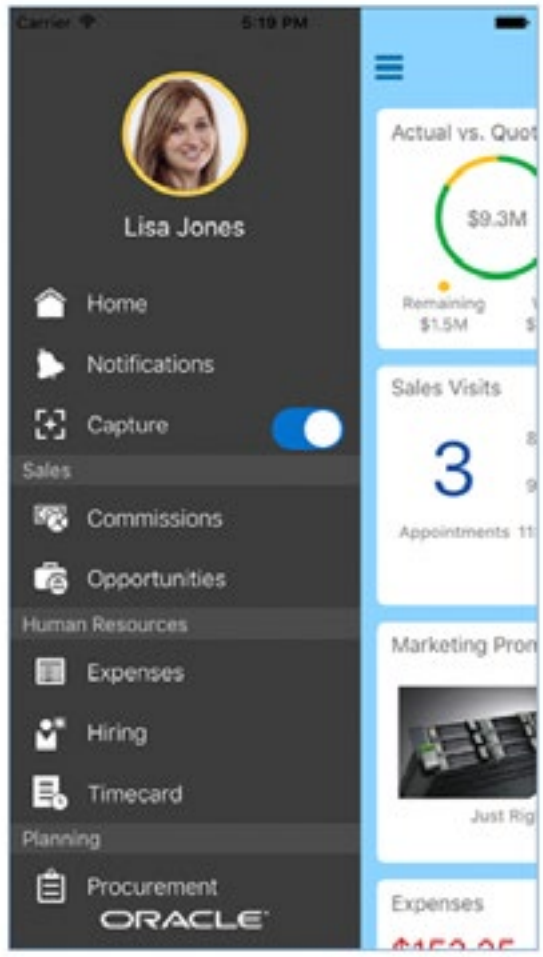

Figure 2-10. Shuttle implemented as a custom springboard

While this eBook introduces you to the layout and navigation features within the sample mobile application included with the RDK, it does not cover every available feature and design pattern. You should explore the application in greater detail to familiarize yourself with the available sample flows before starting extensive design and development activities.

When deployed to the simulator, the application runs automatically. At any time thereafter, you can rerun the application by tapping the launch icon on the iOS simulator or Android emulator.

Figure 2-11. RDK sample application launch icon

Oracle Sales Cloud Sample Feature

The Oracle Sales Cloud Opportunities feature displays opportunities in a list view. Tap anywhere in the list item to drill to the details of a specific opportunity.

For more information about list views and list items, see List View in the design patterns eBook.

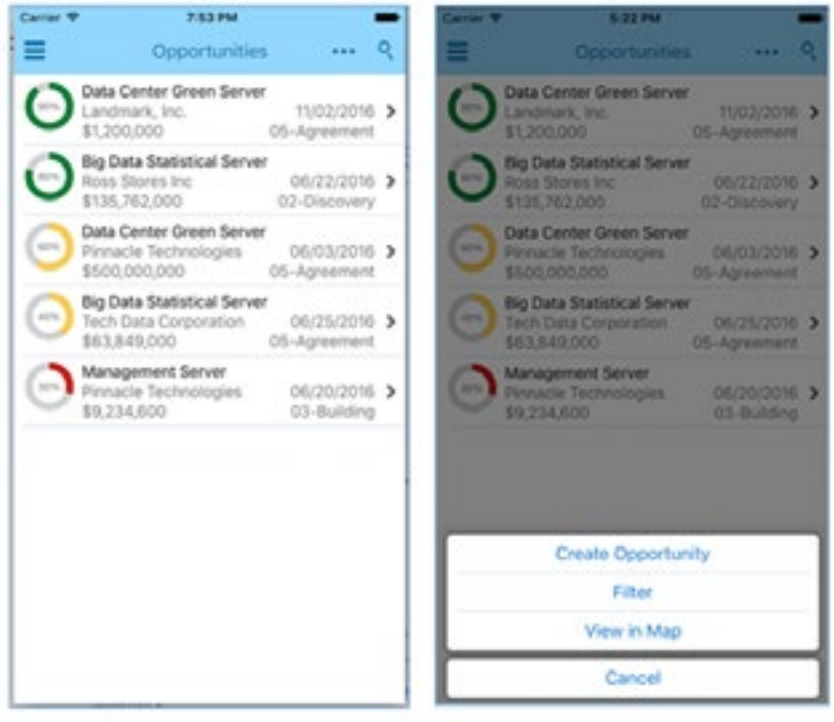

Figure 2-12. (Left) Oracle Sales Cloud opportunities displayed in a list view and (right) with an action sheet

An action sheet is displayed at the bottom of a landing page or detail page when the number of actions a user can take exceeds the number of actions that can be displayed on the global header. The page actions in the action sheet relate to the page content and are accessed by tapping the Actions menu icon on the right side of the global header.

For more information about action sheets, see Action Sheet in the design patterns eBook.

The detail page for a specific object is typically a read-only display of the object data. A detail page can be presented as a:

- Simple detail page: A single scrollable page of details for an object
- Tabbed detail page: A series of detail pages that comprise a summary or profile page and a number of sub-tabs with secondary information

For more information about detail pages, see Detail Page in the design patterns eBook.

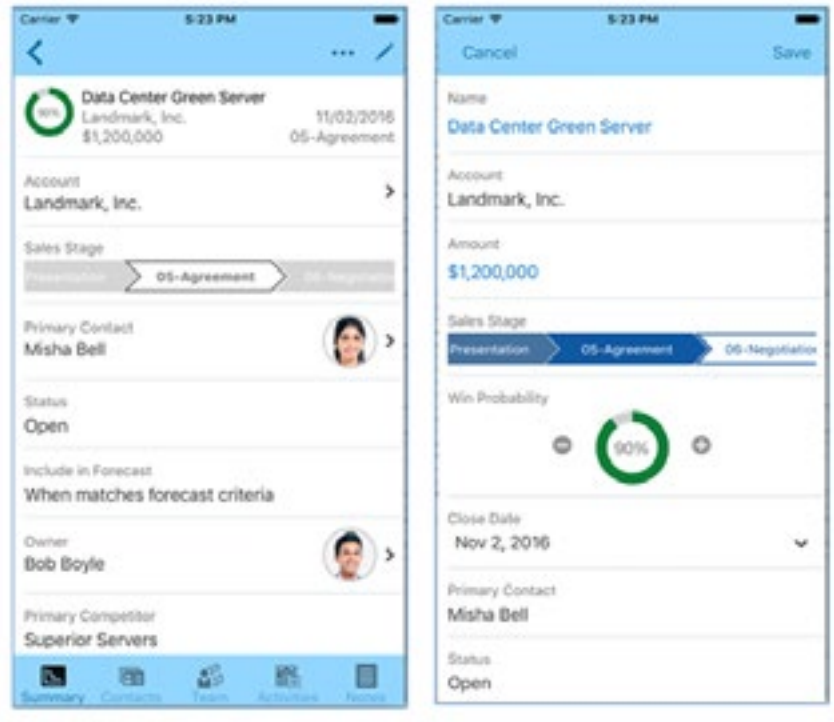

Figure 2-13. (Left) a read-only detail page and (right) an editable page

To open a simple or tabbed detail page, the user taps an object on the originating page. To open an Edit page, the user taps the Edit

icon on the right side of the global header. The edit page has page-level Save and Cancel buttons to commit or rollback changes made on the page. For more information about create/edit pages, see Create/Edit Page in the design patterns eBook.

Oracle ERP Cloud Sample Feature

The Oracle ERP Cloud Self-Service Procurement home page is relatively more complex than the simple list view home page seen in Oracle Sales Cloud Opportunities. The Self-Service Procurement home page supports vertical page scrolling and horizontal content region scrolling actions (for example, Top Categories).

While it may be tempting to pack a large amount of information on a mobile application page, you should minimize the number of data sources and attributes used with a view to accommodate acceptable performance and page simplicity.

Simple search enables users to quickly find objects within a work area or a list. The page type determines where the Search field is displayed, the search behavior, and the scope of the search that will be performed:

- On a landing page, the Search field is displayed in the global header and all records in the work area are searched.
- On a detail page, the Search field is displayed in the global subheader and only the records displayed in the list on the detail page are searched.

To access a search page, the user taps the Search icon in the global header on the landing page or detail page.

For more information about simple search, see Simple Search in the design patterns eBook.

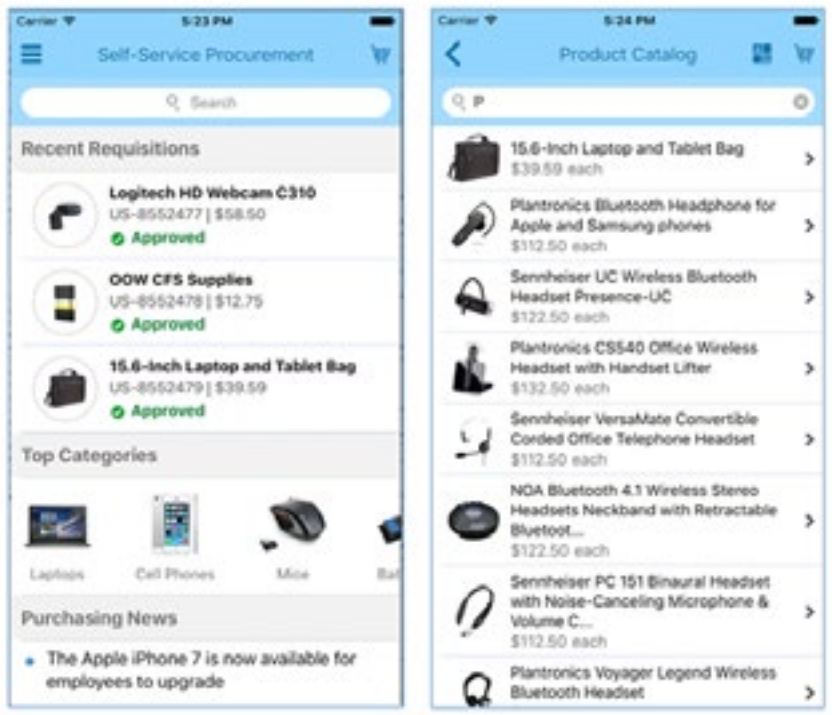

Figure 2-14. (Left) Oracle ERP Cloud Procurement home page and (right) a simple search with results

A variation of the Simple Search design pattern can be seen in the Oracle ERP Cloud Procurement sample feature. When the user types search criteria into the Search field, the product catalog replaces the Procurement home page and displays results based on the search criteria. If the user clears the search criteria, the Procurement home page replaces the product catalog.

To add an item to the shopping cart, the user can:

- Swipe an individual product in the results list and add this item to the shopping cart.
- Tap the item in the results list to see the item details, and then tap the Add to Cart button on the detail page.

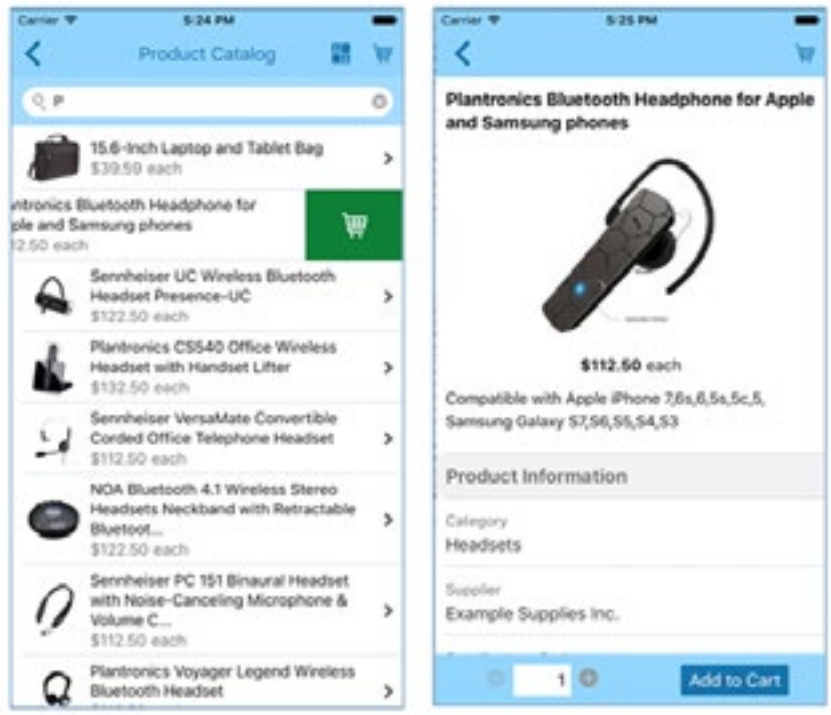

Figure 2-15. (Left) Adding an item to the shopping cart from a list and (right) adding an item to the shopping cart from a detail page

When an item is added to the shopping cart, a confirmation banner is displayed at the top of the page. The total number of items in the shopping cart is displayed on a badge, which overlays the shopping cart icon in the global header.

The user can tap the shopping cart icon on the global header to see the items in the shopping cart. The user can alter the quantity of each item in the shopping cart or remove items from the shopping cart, and then tap Submit to purchase the items.

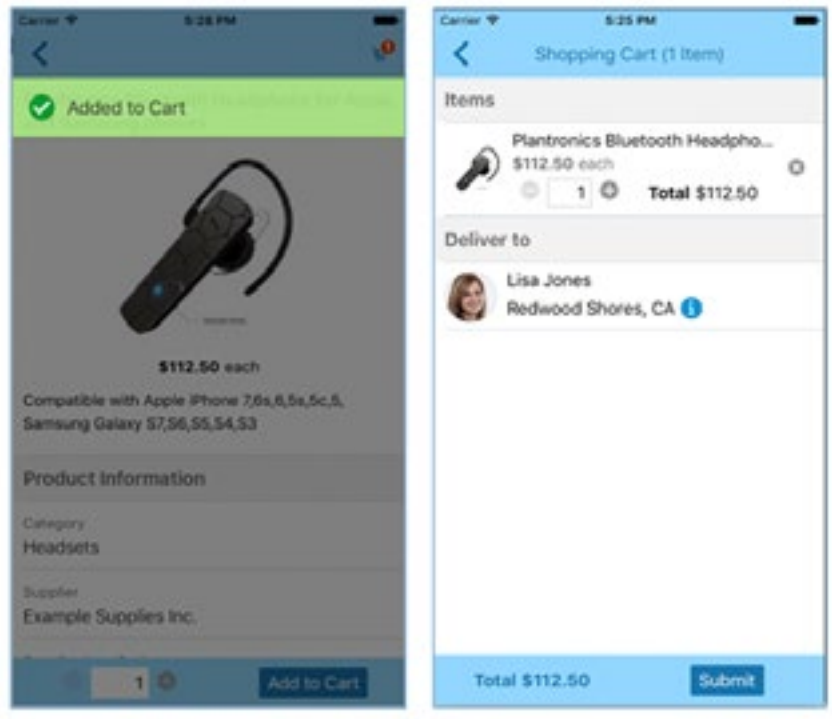

Figure 2-16. (Left) A confirmation banner and (right) the shopping cart

Oracle HCM Cloud Sample Feature

The Oracle HCM Cloud Hiring sample feature demonstrates the use of cards in combination with lists. Cards are used to display hiring requisitions in the Hiring work area. Each card displays a minimal amount of relevant data that enables users to glance at the data and determine next actions.

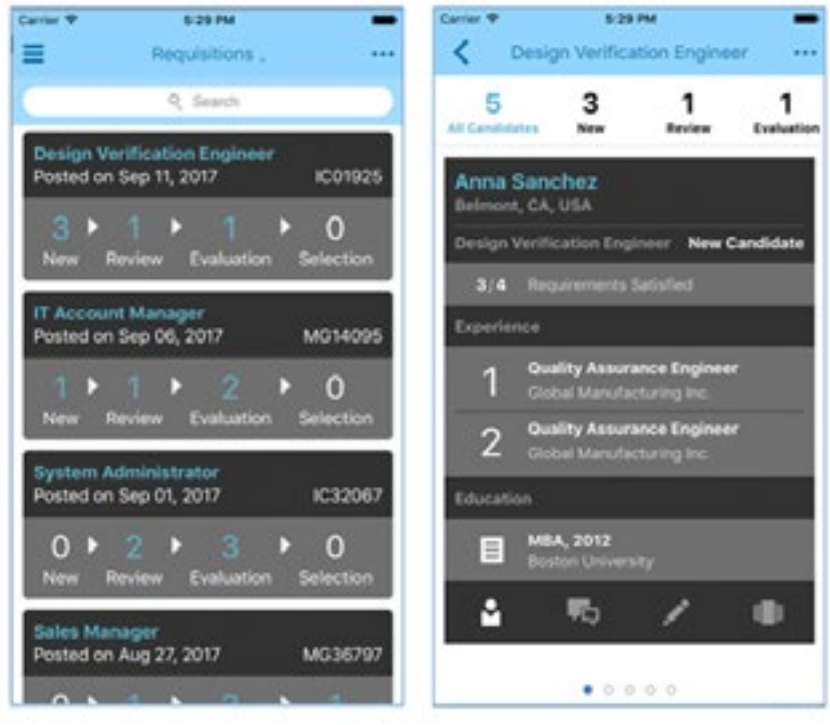

Figure 2-17. (Left) Oracle HCM Cloud Hiring: Card view displaying requisitions and (right) requisition candidates

The user taps a requisition card to display the list of candidates who have applied for that specific job requisition. The user swipes left or right to display each candidate card. The user can narrow the list of candidates that are displayed by tapping the filters at the top of the page. To see details about a candidate, the user taps the candidate name.

The user can change the context of the Requisitions home page from hiring requisitions to hiring candidates, by tapping the title in the global header.

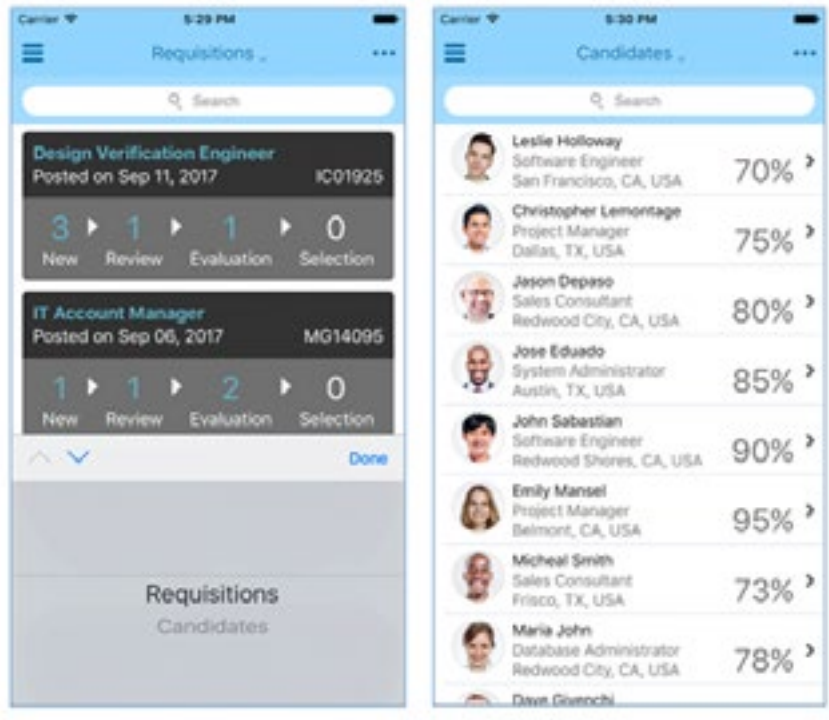

Figure 2-18. (Left) Oracle HCM Cloud Hiring: Context switcher and (right) the candidates displayed in a list view

In a list view, the user can see details about a candidate by tapping the drill icon in a candidate row. On the detail page, the user can swipe to see all submissions made by the candidate. The user can also view remote attachments, such as the candidate's resume or cover letter, by tapping the relevant link in the Attachments section. Attachments are displayed as full-page documents.

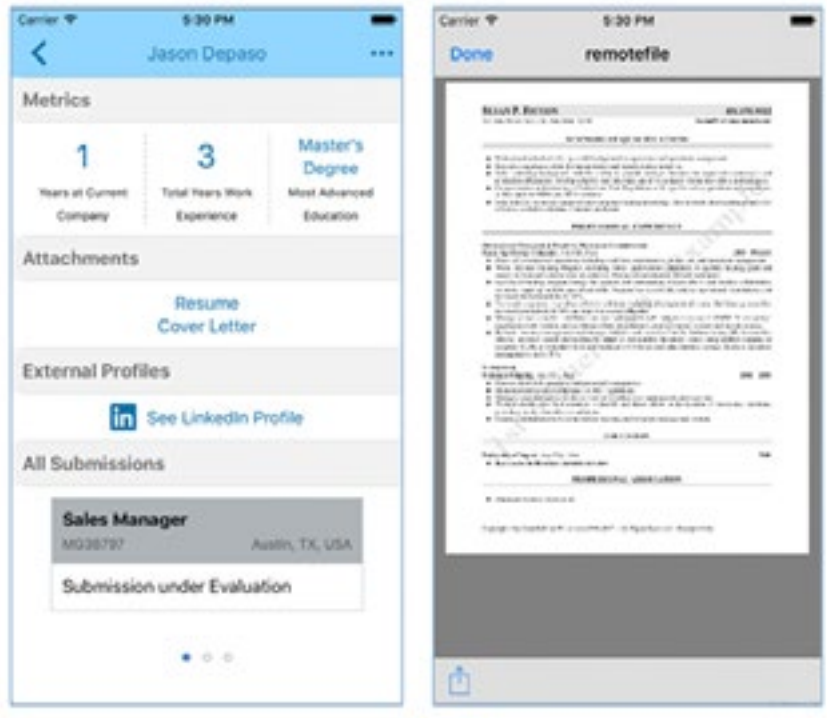

Figure 2-19. (Left) Candidate detail and (right) remote attachment

Be sure to consider performance and usability when presenting drill-down options to the user, especially for use cases in which the user will need to download remote attachments.

2. Building a Mobile Application

You can use the code samples in the RDK demo application to accelerate the build of your mobile application or prototype, by doing one of the following:

- Using the sample application as a reference to see how features are implemented.
- Extracting reusable parts or feature samples and copying these over to your new application.
- Making a copy of the sample application and modifying the feature code.

Build Prerequisites

At this point in the development cycle, you should have garnered stakeholder acceptance of the user interface design, and you should have the data model in place.

Prepare the Foundation

Much like constructing a building, developing an Oracle MAF application begins with putting together the skeleton or foundation of your mobile application, including the Oracle JDeveloper application, projects, and configurations.

Application JWS and Project JPRs

When you create a new Oracle MAF application, it will include two projects:

- Application Controller project: Contains common logic across the application such as, shared code, lifecycle code and so on. There is only one such project in each Oracle MAF application.
- View Controller project: Holds application features. You can either put all of your features into a single view controller project or each feature in a separate view controller project. The latter promotes feature-level reuse, and it is easier to manage large applications and large development teams.

To see the JWS and JPRs that are used in the RDK sample application, see Figure 1.

Configure the Application

You can configure some of the application-level appearance and behavior in the Oracle MAF application XML configuration file. You can configure the name of the application as it appears on the deployed mobile device, include a navigation bar, use the built-in springboard as shuttle, use a custom one, or use none at all.

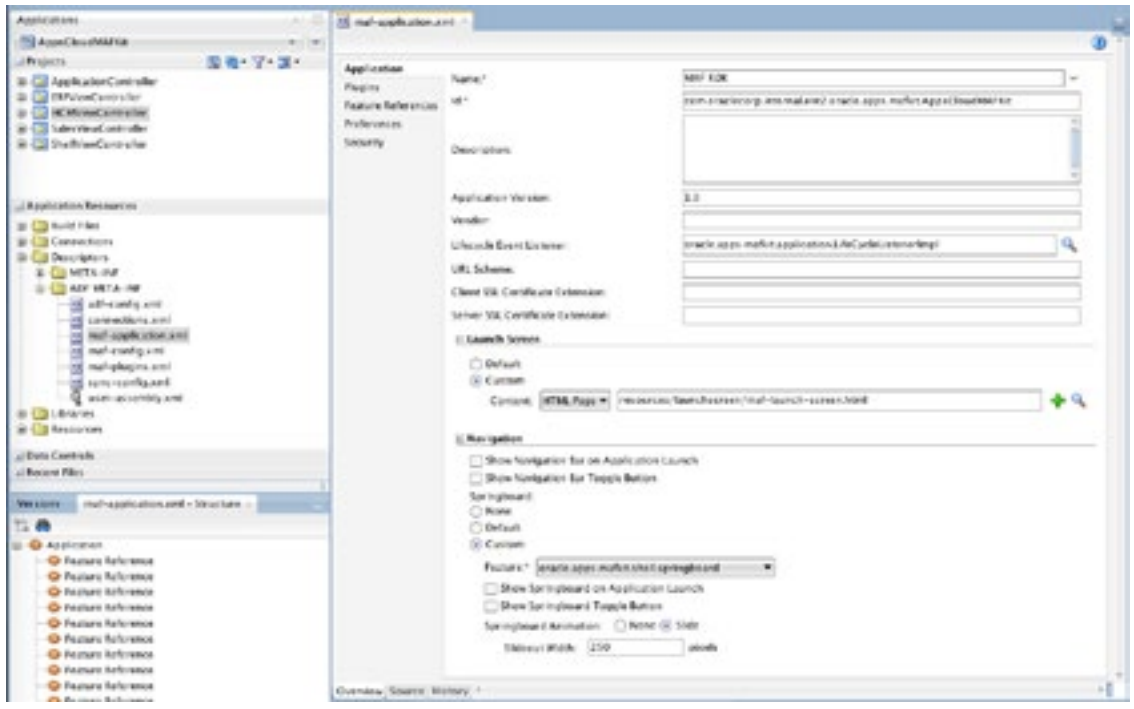

Figure 3-1. Oracle MAF application configuration

Under Plugins, you can configure access to device features, such as the camera, GPS, and so on, as well as additional plugins. These get written to the maf-plugins.xml file, which is accessible under Application Resources, along with the maf-application.xml file.

Under Feature References, you identify features that make up the application, and determine whether you want these to appear on the springboard (shuttle) or navigation bar. You can also set up security features, such as authentication and authorization, the login page you want to use, and the URLs that you want the application to access.

Deployment Profiles and FARs

You set up Oracle MAF Feature Archive (FAR) type deployment profiles for each of the projects, including the Application Controller and View Controller projects. It is good practice to collate the libraries into a single library folder that is peer to the application JWS file.

The application JWS itself will have default Android, iOS, and Windows deployment profiles, but you will need to manually add in the FAR dependencies.

You can configure platform options. For example, for the iOS platform, you can set the minimum iOS version, simulated device and family, and push notification environment.

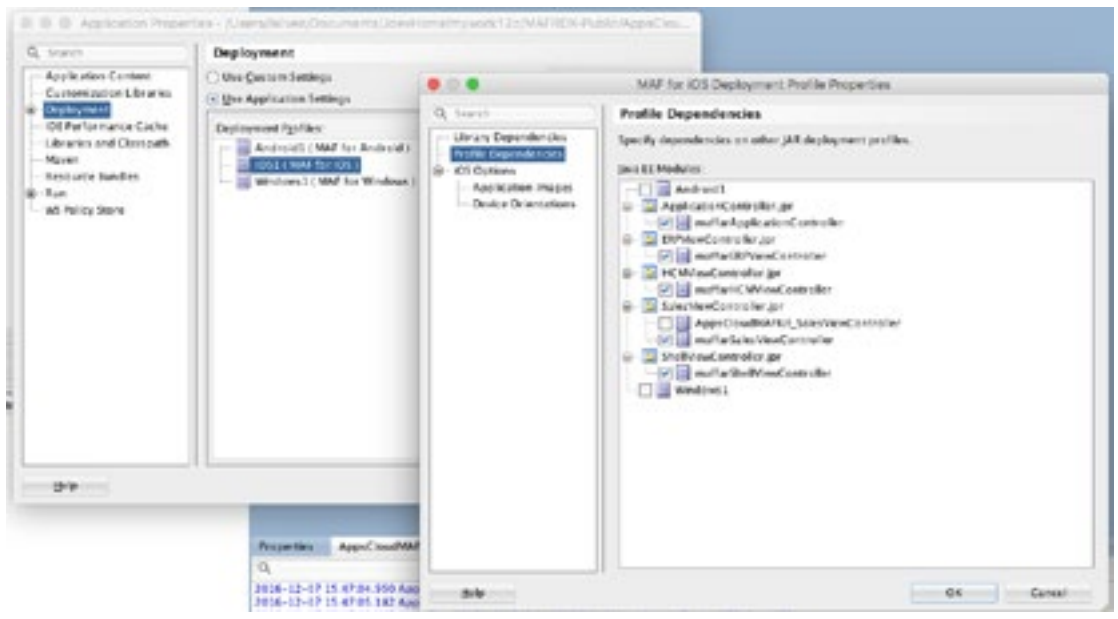

Figure 3-2. FAR dependencies

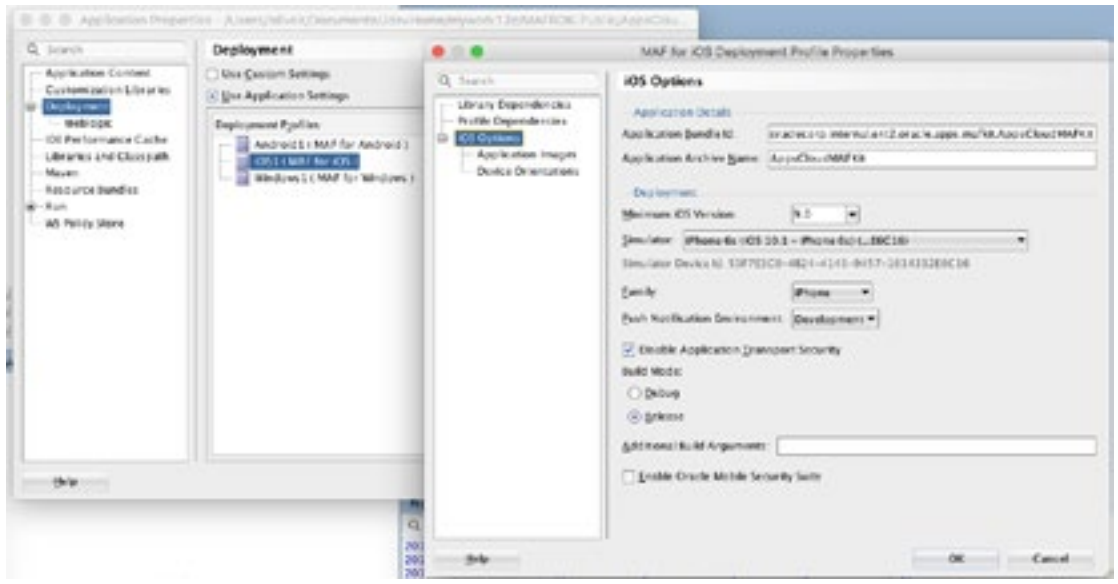

Figure 3-3. Example: Platform deployment options for iOS

During the development phase, you may find it useful to disable application transport security (ATS), to allow non-HTTS requests. However, it is not recommended to use non-HTTPS requests for sensitive data. A better approach would be to fix the non-secure resources required by the application.

Style Sheets

Style sheets and hierarchies can be identified and registered in maf-skins.xml. The styles within these style sheets can be used in all View Controller projects.

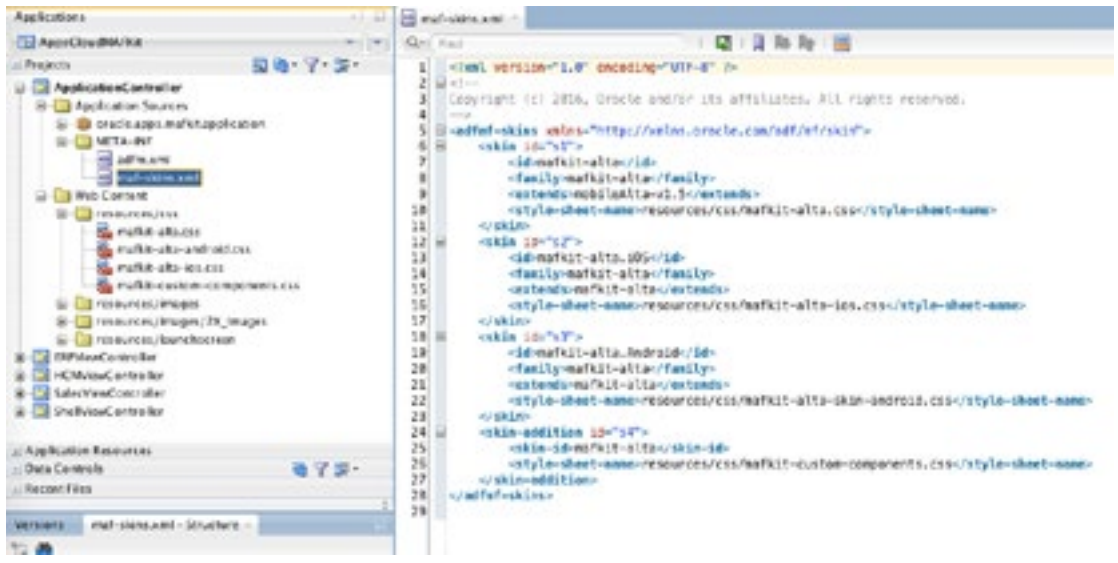

Figure 3-4. Style sheet hierarchies and registration

This completes the minimum scaffolding for the mobile application. We are now ready to code the common parts and application features.

Develop the Common Parts

A typical mobile application will have a home page that the user logs into and some common features, such as notifications, a custom springboard (shuttle) and so on. Enterprise applications typically have a common shell into which the selected feature gets loaded. With Oracle MAF, each of these will be developed as a separate feature and can be organized into a common view controller project. In the RDK sample application, this is the Shell View Controller project.

Configure the Project

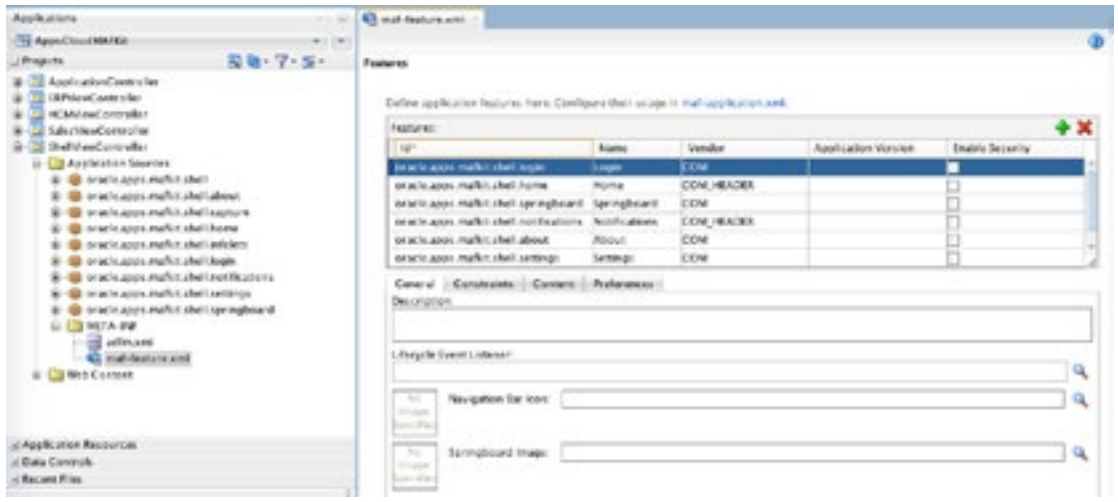

Figure 3-5. Project-specific Oracle MAF feature configuration file

Specify the features within a view controller project in the maf-feature.xml file specific to that project. You can enable security at a feature level to control whether the user needs to be authenticated in order to access the feature.

You can specify constraints to control whether a feature is available. That way, you can ensure that you only offer features that are supported by the device. For example, if the device does not have a camera, you can disable the action wheel feature.

You now identify the pages and flows that make up a feature. Content can be made up of AMX pages, remote URLs, and/or local HTML pages. To exploit the power of Oracle MAF, you should use AMX pages and Oracle MAF task flows.

Build Files for Common Parts

Create the task flows and pages that make up each feature. Depending on the size of the application and consequently the number of features, flows, and pages, you will need a suitable packaging and file naming convention.

Make sure you have the page and task flow organization and architecture defined in consultation with domain experts, assisted by the visual nature of task flows in the Oracle JDeveloper diagram tool.

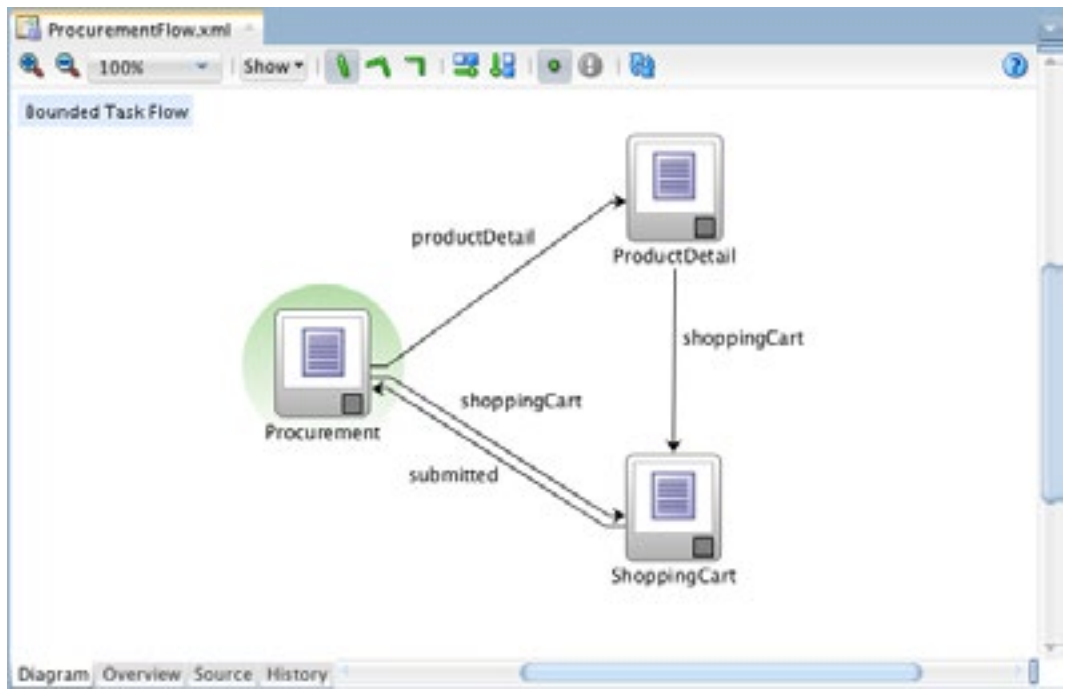

Figure 3-6. Oracle MAF task flow diagram

Build the content of each AMX page using declarative AMX components and data controls. Business logic may reside on device and implemented in Oracle Java or business web services hosted on a remote server. Data controls on the model layer provide a consistent interface when displaying data on a page, regardless of how the business service and data are implemented.

Expression Language (EL) can be used to bind components to business functions. AMX pages can be optionally backed by Oracle Java classes, which can hold page-specific data or logic. Oracle MAF task flows contain a collection of AMX pages with the control flow between these is governed by navigation rules and make up a modular unit of the application. Task flows can be features themselves or can be called by other task flows.

Develop Each Feature

Depending on the size of your application, the number of features and potential feature reusability, you will organize your features

or groups of features into separate view controller projects. Each view controller project can be deployed as a FAR, which essentially packages a feature and its component parts into a library deployed to the file system. This library can be imported into another application by consuming the library and making a reference in the consuming application's maf-application.xml file.

Create a new view controller project (See Section 5.3). Configure the project (see section 5.3.1) and build page content (see section 5.3.2). From an Oracle MAF perspective, discrete application features are the same as common application features.

Leverage Device Capabilities

This section briefly covers device-specific services, such as SMS, email, contacts, camera, and GPS, which are used extensively and relied upon by users. These services can be easily integrated into your mobile application to improve the user experience.

Device Properties

You can test device properties using EL to get details about the device, hardware, and where your Oracle MAF application is running, so that you can hide functionality not available on a particular device. Some common properties are listed below, but this is not an exhaustive list. Refer to the Oracle MAF documentation for a detailed list of such properties.

- Operating System: device.os
- Device Model: device.model
- Plugin Version: device.phonegap
- Camera: hardware.hasCamera
- GPS: hardware.hasGeolocation
- Compass: hardware.hasCompass
- Size: hardware.screen.width, hardware.screen.height
- Status: hardware.networkStatus

Specify feature constraints by using EL, such as #{deviceScope.device.hasCamera}. (See section 5.3.1.)

User Properties

Assuming that you have implemented security and the user has successfully logged into the application, you can find user-specific information to improve the usability of your application. Some user properties follow, but this is not an exhaustive list. Refer to the Oracle MAF documentation for a detailed list of such properties.

- Privileges: user.privileges
- Roles: user.roles

On-device Capabilities

You can easily integrate on-device capabilities into your application by using built-in operations available as data controls under the device features groups.

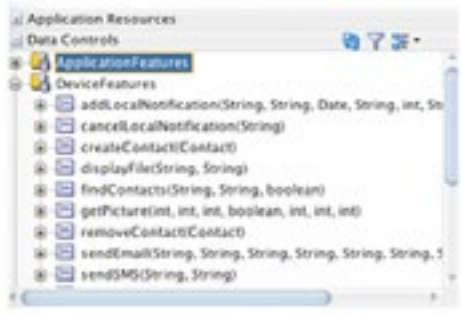

Figure 3-7. Device features as data controls

Simply drag-and-drop the methods corresponding to the device features you want, and add values for associated parameters.

Implementation Techniques

You can integrate device services into an Oracle MAF application in one of three ways, depending on the level of control you need over the operations. Data controls provide the easiest access using drag-and-drop. Use Oracle Java APIs for finer-grained access, for example doing some processing for adding business logic. Use JavaScript APIs if you want to access these services from HTML.

Accessing Web Services

Business logic can typically be built either on-device or off-device and accessed remotely, while hiding the implementation and data

storage from the developer. The most common web service implementation technologies are SOAP and REST. Both use HTTP as a transport mechanism, but REST services are popular with mobile applications.

SOAP provides well-defined interoperability, has a formal contract defining its interface, is transport agnostic, and has additional standards for security and reliability. However, SOAP adds to the message size and complexity, which requires greater tooling on the client.

REST has a lighter and less complex implementation and consequently uses less bandwidth. It has a uniform interface for a given URI based on HTTP methods, and uses out-of-the-box HTTP security and caching features. However, the interface or contract is less clearly defined.

Starting with the Oracle MAF 2.2.0 release, support for SOAP services has been deprecated. The Oracle MAF team recommends using JSON based REST services as they perform much better and are more suited to mobile application consumption. Data controls provide a simple mechanism for integrating REST web service functionality. However, for greater control on the invocation and returned data, you can programmatically invoke REST services from Oracle Java.

Integrating with Oracle Mobile Cloud Service

Oracle Mobile Cloud Service (MCS) provides a Mobile Backend (MBE) that simplifies and secures the process of connecting your mobile applications with your enterprise systems and other cloud services. Oracle MCS also provides insight into the success of your mobile strategy through usage and performance metrics.

For more information about Oracle MCS, see https://cloud.oracle.com/mobile.

Integrating Oracle MAF with Oracle MCS can easily be achieved by using built-in APIs. For more information about using Oracle MCS with Oracle MAF, see the Oracle MAF developer guide at http://docs.oracle.com/middleware/maf233/mobile/develop-maf/toc.htm

3. Deploying a Mobile Application

To test your mobile application, deploy it to your mobile device or virtual device (simulator or emulator). When you create a new Oracle MAF application, Oracle MAF automatically generates deployment profiles for each target platform.

For more information about platform-specific deployment details, see the Oracle MAF developer guide: http://www.oracle.com/maf.

Android Platform

While working with the Android platform, you can deploy your application in debug or release mode to an Android device, emulator, or APK file. The last option is required when you want to publish your application to users through Google Play or a similar marketplace.

iOS Platform

While working with iOS, you can only deploy your Oracle MAF application using an Apple computer. You can deploy the application directly to an iOS device simulator or to an IPA file in iTunes, which can then be synchronized to an Apple device. You need to be a registered Apple Developer in order to distribute your Oracle MAF application through the Apple App Store.

Windows Platform

While working with the Universal Windows Platform (UWP), you can deploy your application in debug or release mode to the local machine on which you develop the application, or to a UWP file, which can be installed on a supported UWP device, and published in the Windows Store.

Frequently Asked Questions

This section answers some of the most common questions about the Oracle Mobile Cloud Service Applications User Experience Rapid Development Kit. For more information, see our Oracle Usable Apps page (http://tinyurl.com/paas4saas).

Can I use the RDK to design and develop applications for different mobile platforms?

Oracle MAF is a write once, deploy multiple times development environment. Aside from adjusting style classes for different mobile platforms, a single code line can be used to generate executable files to install on different mobile platforms, including Apple iOS, Google Android, and Microsoft Windows Mobile.

Can I use Eclipse for Oracle MAF development?

Yes. You can use Oracle Enterprise Pack for Eclipse (OEPE) to develop Oracle MAF applications. You can see a brief demo of using OEPE for Oracle MAF application development at http://download.oracle.com/otn_hosted_doc/maf/oepe-maf-demo/index.html

Does the RDK work with Xcode 8.3.3 and higher?

Yes. The RDK was developed using Xcode 8.3.3 and tested on an early drop of Xcode 9. No code changes are required.

Can the RDK be deployed to iOS10 and iOS11?

Yes. The RDK was deployed to iOS10 during development and subsequently deployed and tested on iOS11. No code changes are required.

Which versions of Oracle MAF does RDK work with?

The RDK was developed using Oracle MAF 2.4.1 and should work with any later version of Oracle MAF, too.

Does the RDK use Oracle Alta UI?

Yes. The sample applications are based on Oracle Alta Mobile (http://www.oracle.com/webfolder/ux/mobile/index.html) and contain specific class additions.

Does the RDK use Oracle MCS?

No. The sample application does not connect to any external services including Oracle MCS. However, you can use the MCS Mobile Backend in conjunction with an Oracle MAF application. For more information, see the Oracle MCS documentation.

Can I use the RDK code in production?

While you can use the RDK code in production, you should note that the sample application included in the RDK is merely demo open source code. It is not officially supported, so please do not contact Oracle Support with questions about the RDK or production-related questions.

Also note that the RDK sample application does not connect to any REST data services, does use the local SQL-Lite database, and does not have production-level authentication and authorization.

Does the RDK invoke Oracle Applications Cloud services?

No. The RDK sample application is built to run standalone without dependence on external services and a network connection. However, you can find code examples at http://www.oracle.com/maf for invoking Cloud REST services.

Where can I find additional resources on Oracle MAF technology?

To learn more about Oracle MAF, visit http://www.oracle.com/maf. Here you will find a wealth of resources, including tutorials, code samples, downloads, and links to forums and documentation.

How much Oracle MAF do I need to know?

The RDK is not intended to be a learning tool for developers who are new to Oracle MAF. You should be an Oracle MAF developer with intermediate to advanced skills, and you should be able to build mobile and cloud applications.