31 October 2014

Mobile Technology

Breanne Nowicki
Brett Koenig
Chris Meek
David Meyer

Watson Capstone - Lifelong Learning

Agenda

- History of mobile development
- Event-based programming
- iOS overview
- Android overview
- Hybrid App overview
- Questions

History of Mobile Dev

- First mobile phone 1983 by Motorola, $4,000
- Prior to applications, Wireless Application Protocol using WML
- Evolved from readers to catalogues
- Content/Payment delivery over SMS

History of Mobile Dev

- PDAs began running compact operating systems
- Handheld gaming devices and music players came about

History of Mobile Dev

- Mobile apps go back to calculators, ringtone editors in 1990’s
- Mobile development as we think of it is when it opened to third-party development
- Mid 1990’s third-party C/C++ development on Palm

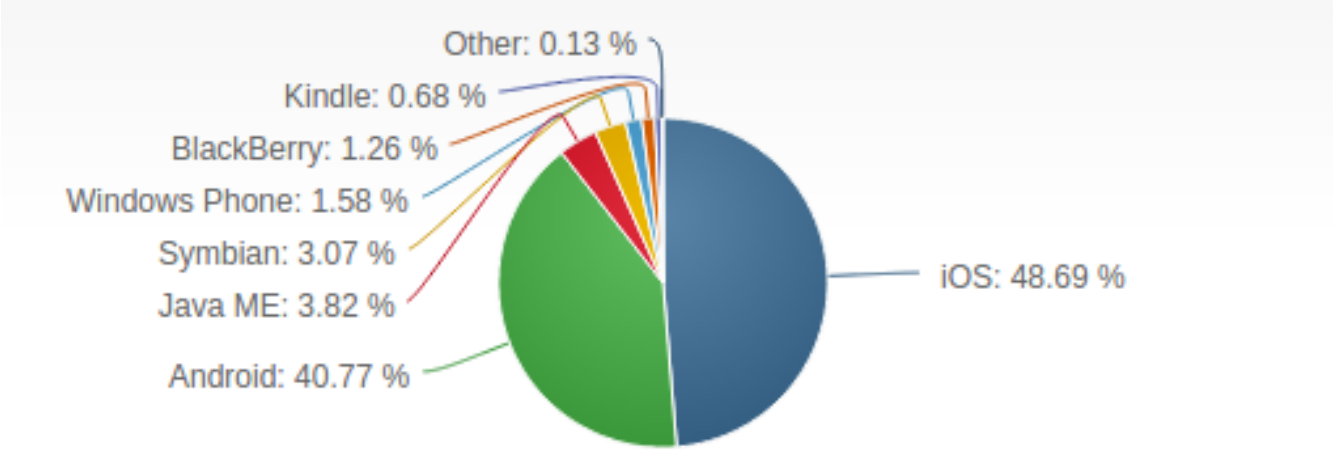
History of Mobile Dev

- January 9, 2007
  - October 22, 2008
- Today - 1.2+ million apps on App Store
- Only web applications at App Store launch
- 2008 SDK was released and native applications were born

Event-driven programming

“In computer programming, **event-driven programming** is a programming paradigm in which the flow of the program is determined by events such as user actions (mouse clicks, key presses), sensor outputs, or messages from other programs/threads. Event-driven programming is the dominant paradigm used in graphical user interfaces and other applications (e.g. JavaScript web applications) that are centered on performing certain actions in response to user input.” - Wikipedia

Market share



Source: netmarketshare.com

iOS - Background

- Released in June 2007 by Apple for the iPhone
- Developers use Objective-C and Swift
- OS for iPhone, iPad, and Apple Watch

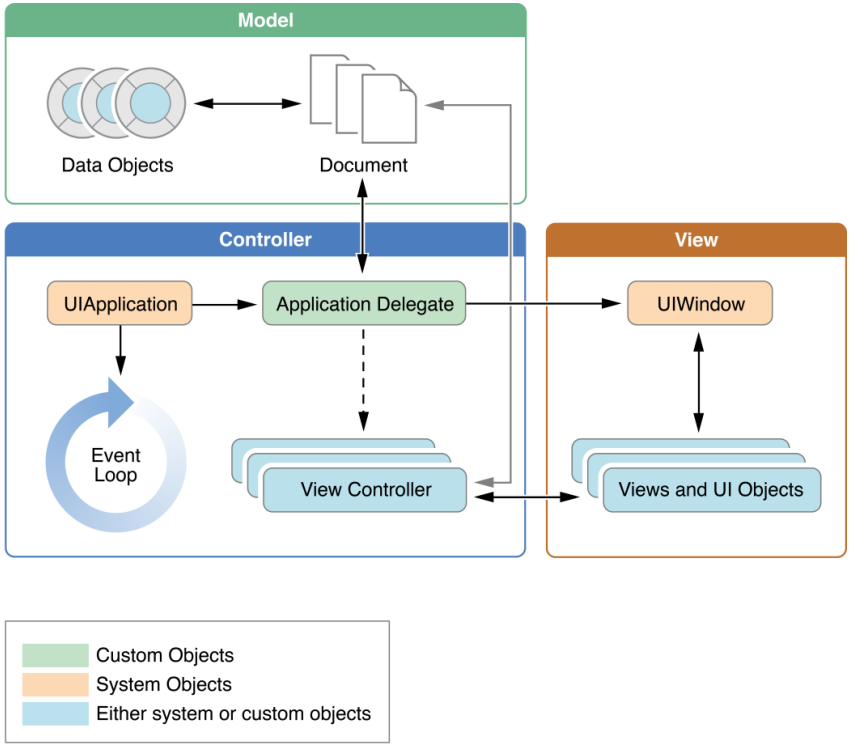
iOS - Development

- Objective-C and Xcode
- Swift programming language introduced with iOS 8
  - support for “playgrounds”

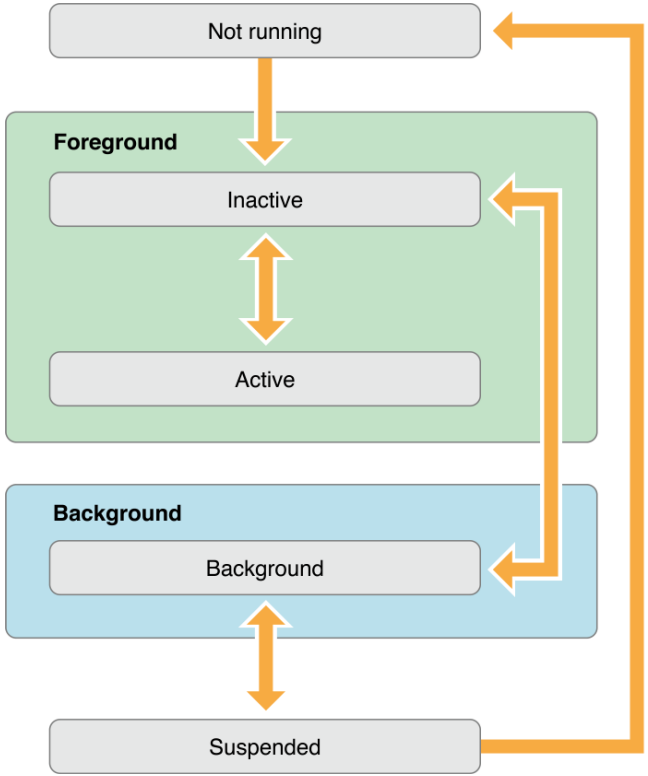
iOS - Components

- Controllers
- Views
  - layout specified via Interface Builder, connected to code with @IBOutlet tags
- Delegates
  - handle multiple “events”, unlike listeners

iOS - Architecture

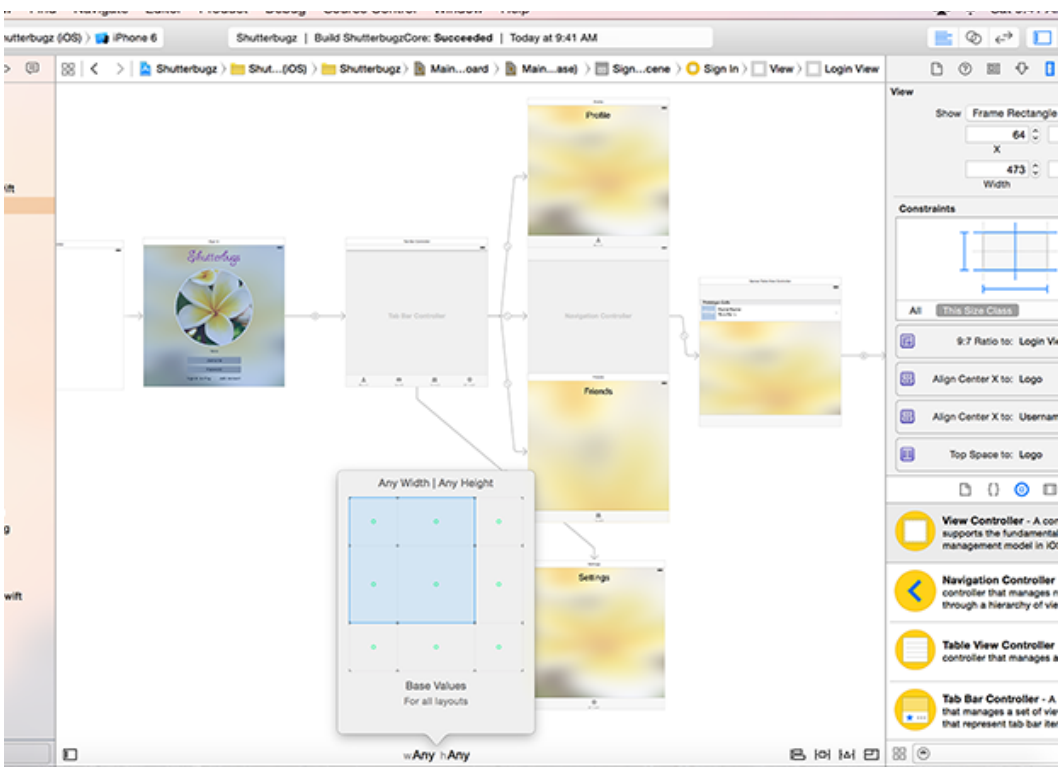


iOS - Architecture (cont'd.)

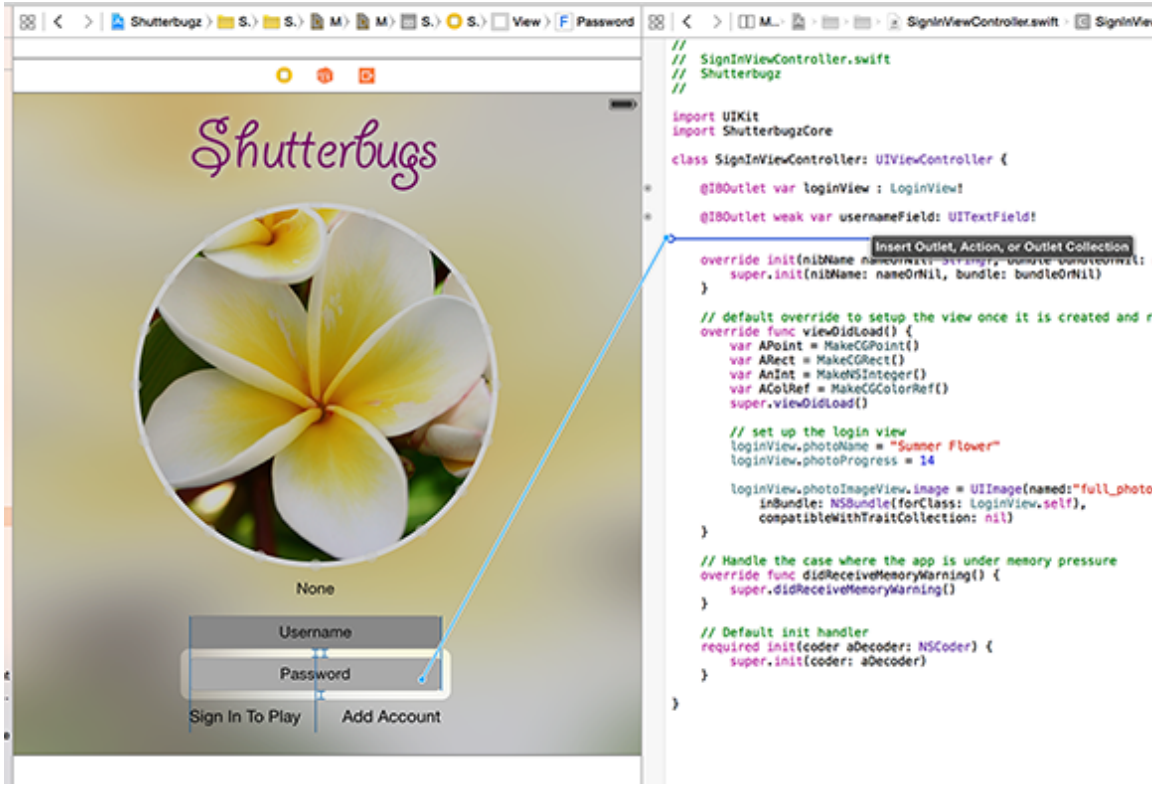



```
application:willFinishLaunchingWithOptions

application:didFinishLaunchingWithOptions

applicationDidBecomeActive

applicationWillResignActive

applicationDidEnterBackground

applicationWillEnterForeground

applicationWillTerminate
```

iOS - Interface Builder



iOS - Interface Builder (Assistant)



iOS - Swift

- inferred types

```
var myString = "Hello World"
```

- closures

```
reversed = sorted(names, { (s1: String, s2: String) -> Bool in
    return s1 > s2
})
```

- let keyword for constants

```
let numbers = [16, 58, 510]
```

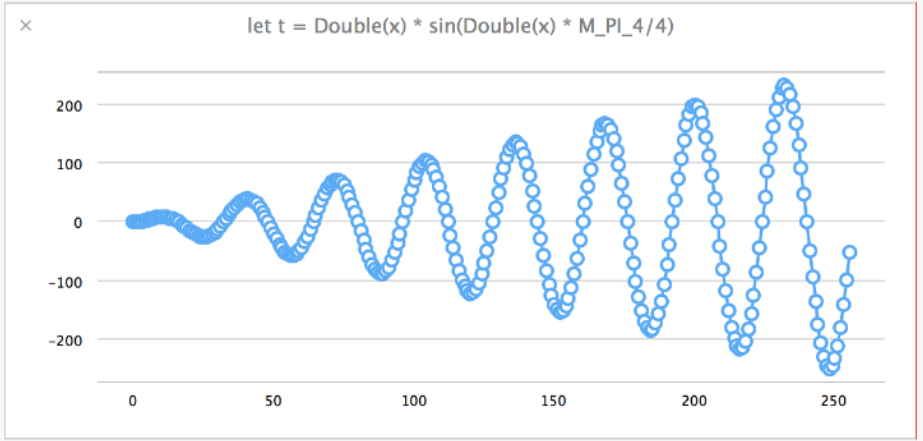
- tuples, multiple return values

```
func someFxn(array: [Int]) -> (num: Int, otherNum: Int) {
    var myNum = array[0]
    var myOtherNum = array[1]
    return (myNum, myOtherNum)
}
```

iOS - Playgrounds

```
for x in 0...255 {
    let t = Double(x) * sin(Double(x) * M_PI_4/4)
}
```

(256 times)



iOS - Pros/Cons

- quick feedback when developing UI
- long-standing ecosystem, documentation, and community
- decreased need to support many different types of devices
- large market share

- “walled garden”
- setting up development devices can be more difficult than Android
- joining the Developer Program costs money
- the approval process to publish in the App Store can be tricky

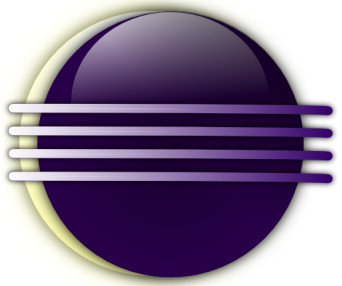
Android - Background

- Mobile OS introduced commercially in September 2008 with most recent version being 5.0 (Lollipop)
- Currently developed jointly by Google and Open Handset Alliance as open source project
- Unix based and written in C, C++, Java with developers using Java and Android APIs to write apps
- Steadily gained market share to become most popular mobile OS. Sales have risen to account for as much as 80% all phone sales last year.
- Runs on variety of devices including phones, tablets, wearable devices, and automobile platforms

http://en.wikipedia.org/wiki/Mobile_operating_system#Market_share
http://en.wikipedia.org/wiki/Android_(operating_system)

Android - Dev Environment

- Android Studio (Beta)
- Eclipse with Android Developer Tools (ADT) plugin
- Each have built in emulator to test apps and capability to update Android SDK to stay current
- Genymotion is an alternative emulator that is touted as faster, more customizable, and compatible with both IDEs



Android - Design Features

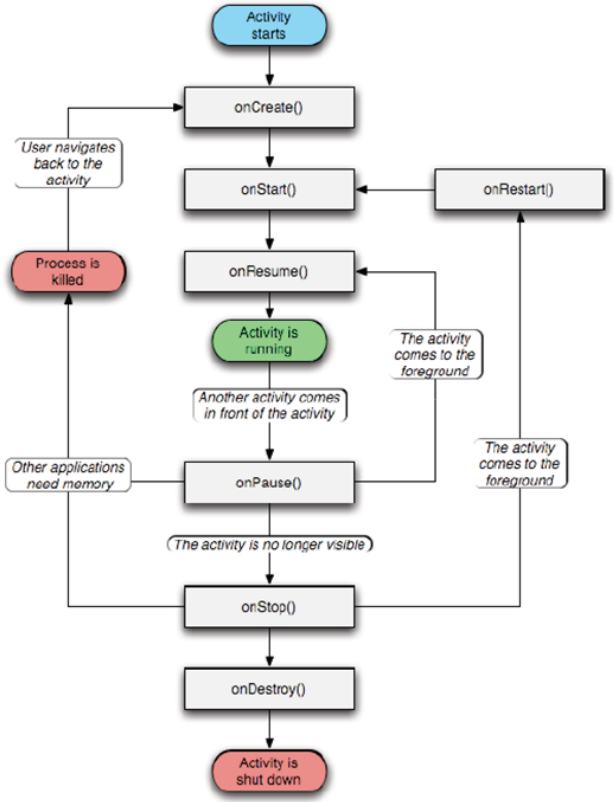
- Activity – easy to think of this as a “screen” in an application
- Layout – defined in an xml file; different flavors (absolute, relative, linear)
- Views – visible component of an activity; location defined in layout file; often have onClick handlers
- Service – a background process that may not have a view associated
- Widget – view only interface to service

Android - Code Sample

```
14  public class AppSettingsActivity extends ActionBarActivity {
15
16      Intent settingsIntent;
17      Button logoutButton, deleteButton;
18
19      View.OnClickListener delete = (new View.OnClickListener() {
20          public void onClick(View v){
21              // todo
22
23          }
24      });
```

```
34      @Override
35      protected void onCreate(Bundle savedInstanceState) {
36          super.onCreate(savedInstanceState);
37          setContentView(R.layout.activity_app_settings);
38          //define intents
39          settingsIntent = new Intent(this, SettingsActivity.class);
40          deleteButton = (Button) findViewById(R.id.delete_account_button);
41          deleteButton.setOnClickListener(delete);
42          logoutButton = (Button) findViewById(R.id.logout_button);
43          logoutButton.setOnClickListener(logout);
44      }
```

Android - Activity Lifecycle



Android - Publishing

- Google Play Store formerly known as Android Marketplace
- Platform for distribution of digital Music, Literature, Games and Applications for mobile use
- Initially released in October 2008, it now has over 1.3 million apps
- Upfront notification of permissions an app requires
- Google monitors the apps listed on the marketplace for security and policy violations
- Amazon Appstore is also growing distribution platform

Android

Advantages

- Free to start developing
- Large and growing user base
- Less restrictive policies to distribute app
- Multiple distribution platforms

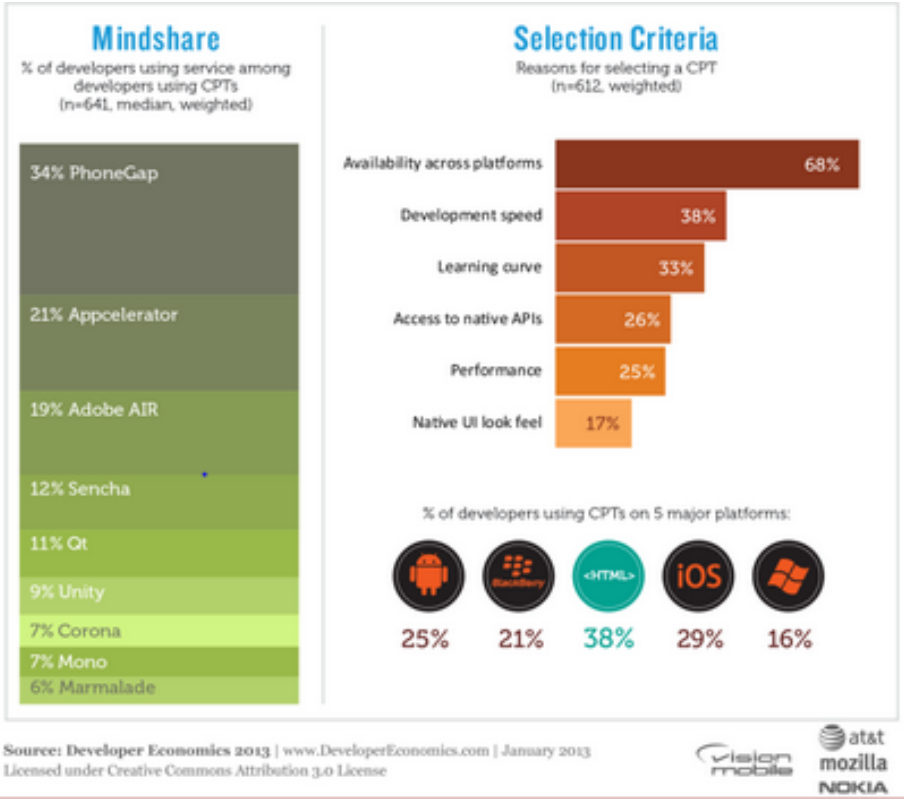
Disadvantages

- Registration fee to distribute app to Play Store
- Large variety of devices and OS versions to support
- Multiple distribution platforms

Hybrid Apps - Background

- Cross-platform - part native app, part web app
- Based on HTML5 web services
- Wrapped in container specific to target OS
- Can access phone’s hardware and native APIs

Hybrid Apps - Overview

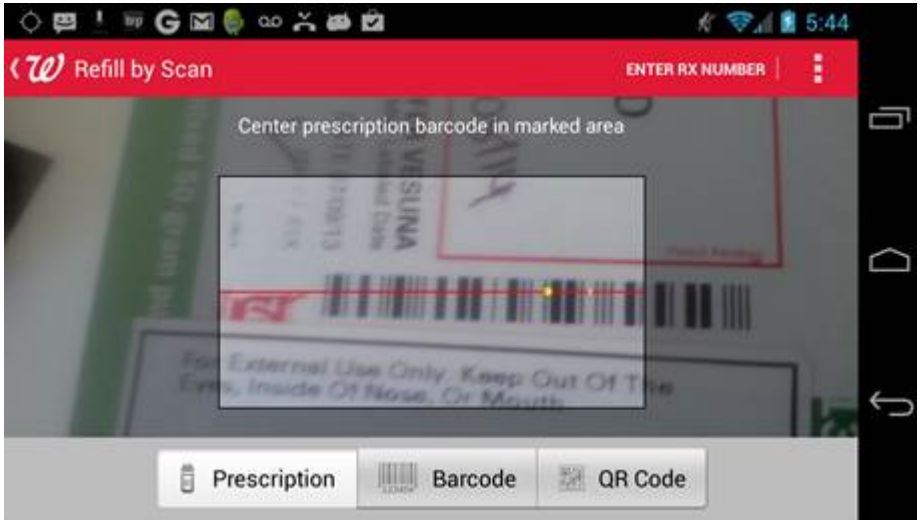


Hybrid Apps - Development

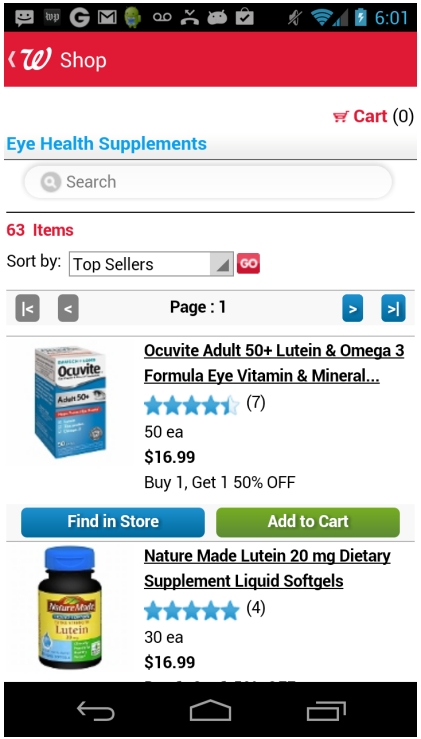
1. Develop App
   - Use tools to develop app with HTML, CSS, Javascript
     - Sencha Touch - mobile framework
2. Add device specific features and wrap
   - Use tools to wrap HTML code or existing website into native coded container
     - Adobe PhoneGap
     - Apache Cordova (open source)

Hybrid Apps - Example

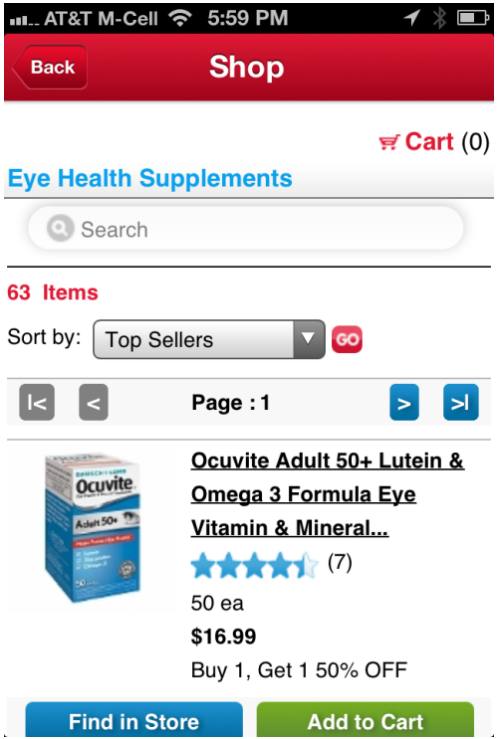
- Walgreens
  - ‘Scan to refill’ feature uses native camera on Android and iOS
  - Shopping in mobile store feature is the website wrapped to



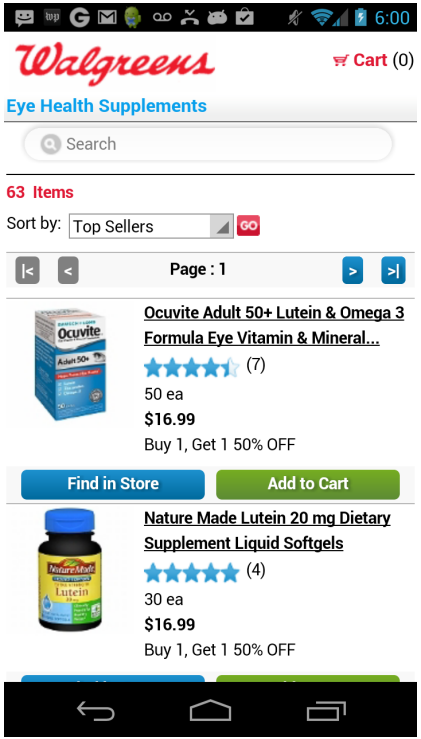
Hybrid Apps - Example



Android app



iPhone app



Android web browser

Hybrid Apps

Pros

- Many developers in HTML, JS, CSS
- Reduce dev costs
- Plugins
- Greater reach = easy marketing
- Maintenance

Cons

- UI differences
- Platform integration
- Framework may not allow for some features
- Code could run slower

Questions

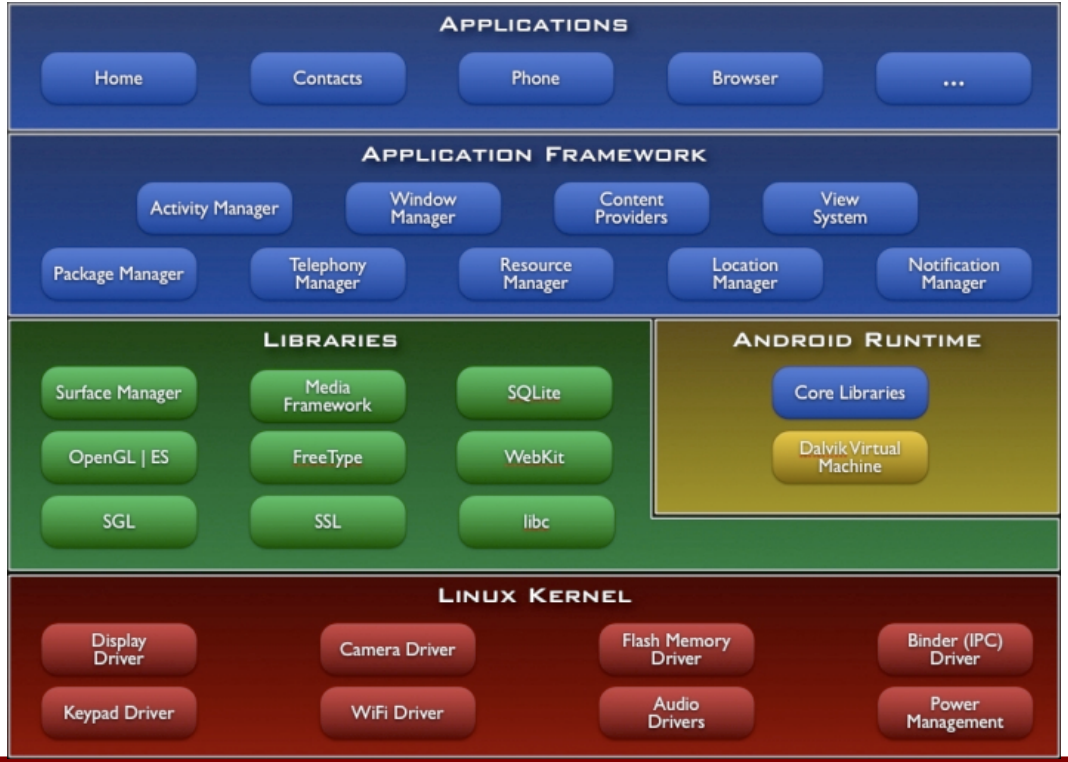
Resources - Android

- http://www.android.com/
- http://developer.android.com/develop/index.html
- http://www.stackoverflow.com
- http://www.genymotion.com/
- http://www.vogella.com/tutorials/android.html

Android - Design Features

- Action Bar – introduced in 3.0; provides useful user options
- Options menu – resides in Action Bar and present by default; options defined in menu layout xml files
- Fragments – reusable portions of UI; useful for supporting both small and large sized screens
- Persistence – Data can be stored on internal phone storage or external SD card using SQLite
- Content Providers – avenue for sharing data between other apps on device (like contacts or media)

Android - Framework



Blue background:
Java
Other colors:
C/C++

History of Mobile Dev - Resources

- http://en.wikipedia.org/wiki/Google_Play_Store
- http://en.wikipedia.org/wiki/Android_(operating_system)
- http://www.cultofmac.com/125180/steve-jobs-was-originally-dead-set-against-third-party-apps-for-the-iphone/
- http://www.informit.com/articles/article.aspx?p=1388959
- http://mobileappin.wordpress.com/2014/05/15/a-history-of-mobile-application-development/
- http://manifesto.co.uk/history-mobile-application-development/
- http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/
- http://en.wikipedia.org/wiki/App_store
- http://www.sundoginteractive.com/sunblog/posts/how-did-the-iphone-start-the-smartphone-explosion
- http://www.uky.edu/~jclark/mas490apps/History%20of%20Mobile%20Apps.pdf