Welcome to IS3261

Mobile Apps Development for Enterprise

This course is about

- Understanding the basics of Android
- Learning how to write Android Apps
- Acquiring practical skills in Android Apps development

You are expected to

- be resourceful or willing to learn to be resourceful
  - Learn to look for information yourself ! You need constant updating!
- have lots of hands-on practice – get your hands very dirty
- code in Java (a lot), xml, JavaScript, css3, html5

Course Structure

• Lecture	
• Tutorial	15%
• Mid Term	10%
• Course Project	35%
• Final Exam	40%
(4th Dec 2017, 12noon)	

Teaching Team

Lecturer:	A/Prof Ng Teck Khim office: COM2 03-28 email: ngtk@comp.nus.edu.sg

Teaching Assistants:

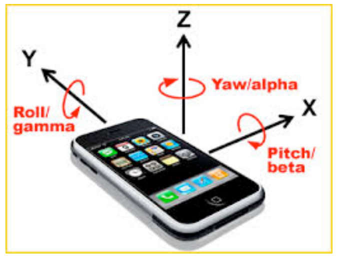
Guo Mingxuan	Quek Yang Sheng	Alex Fong Jie Wen
a0130749@u.nus.edu	ysquek@u.nus.edu	alex.fong.jie.wen@u.nus.edu

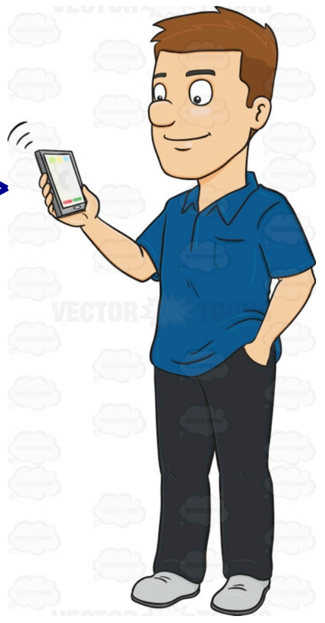
Potential of Mobile

Computation

Communication

Sensors





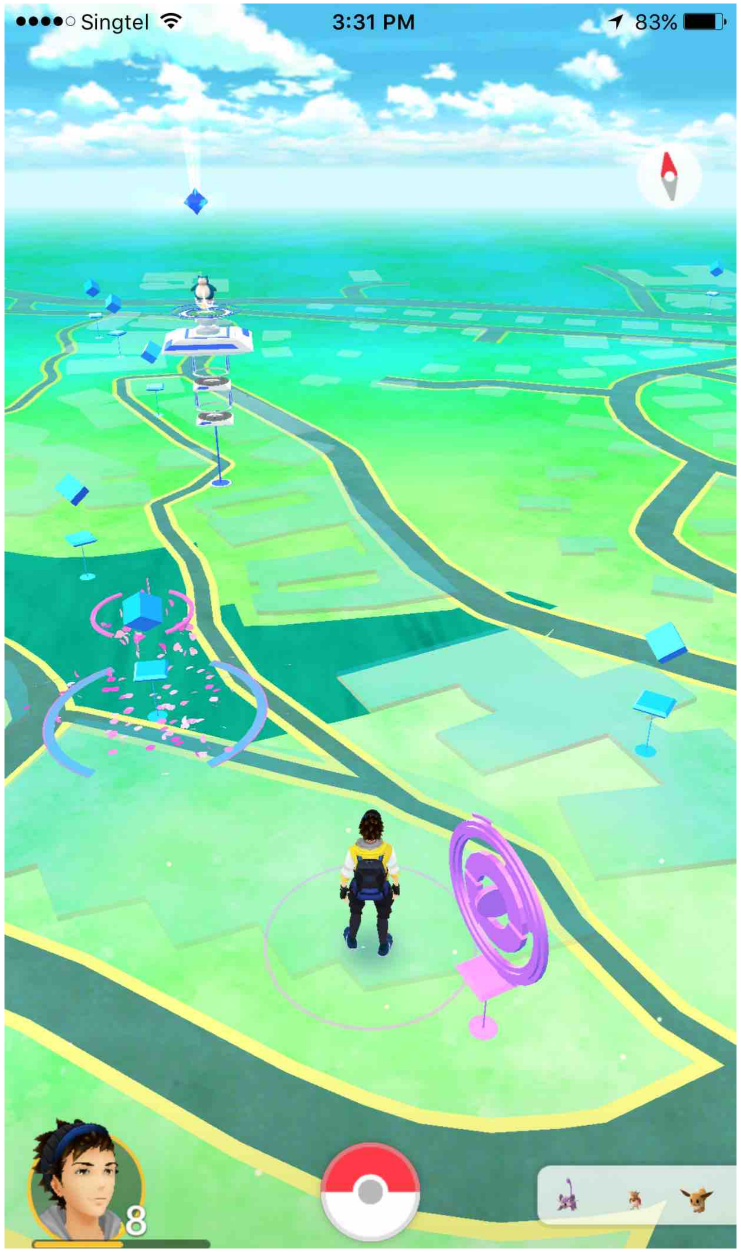
Smart Phone is a Computer Packed with Sensors and Communication Functionalities!

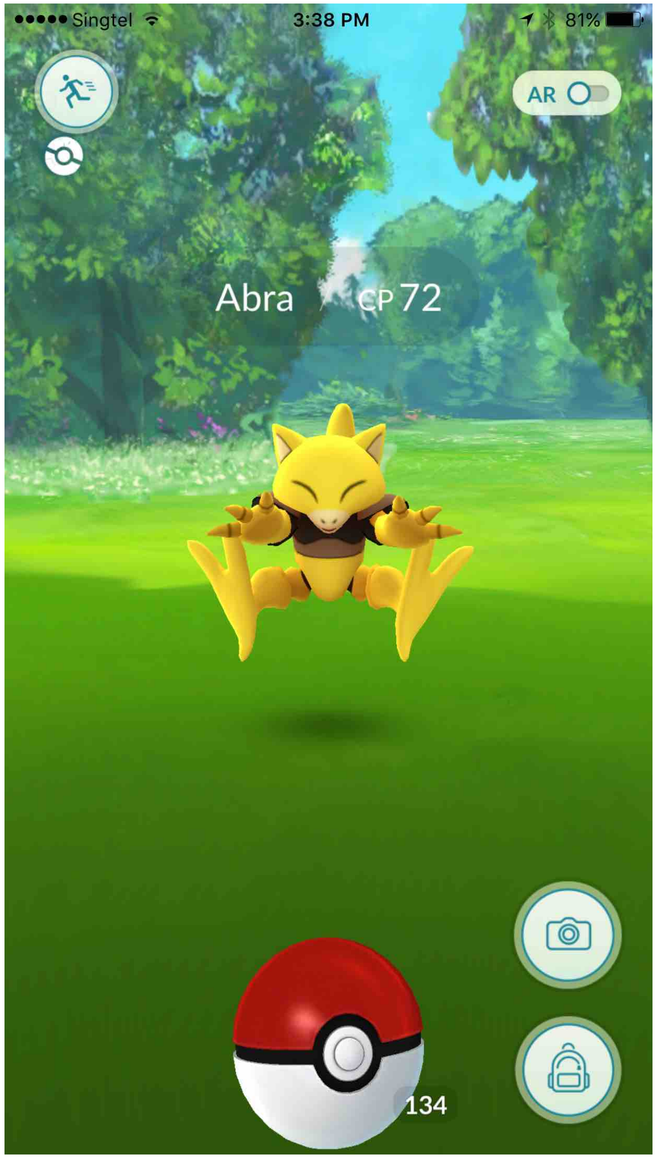
But …

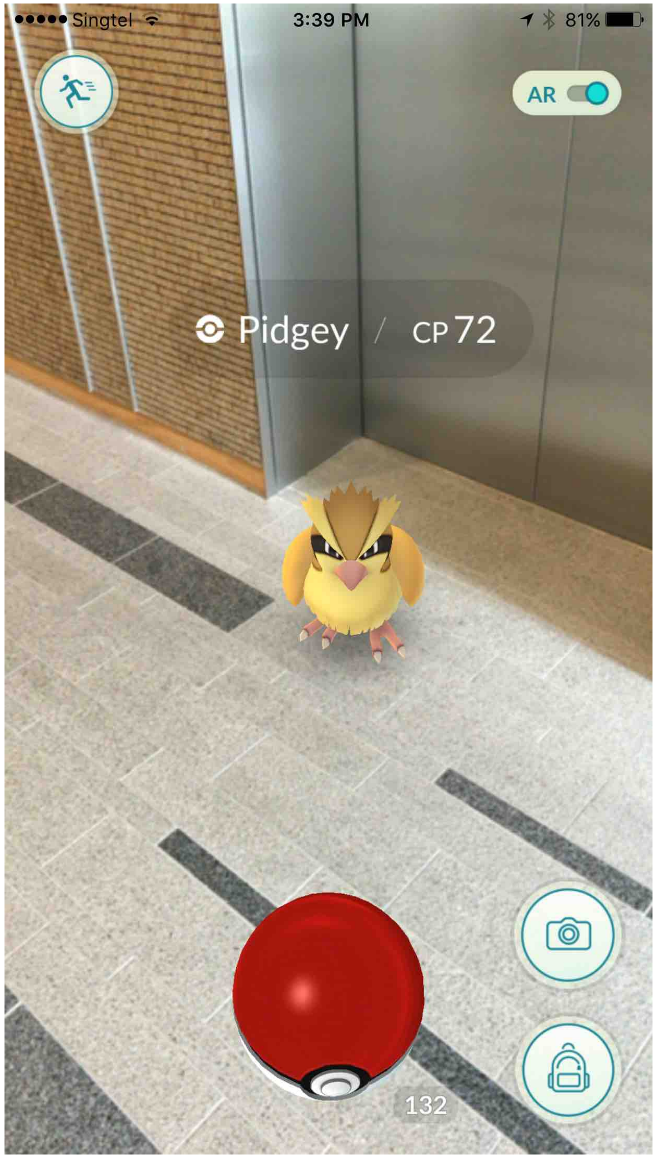
Smart phone == computers 20 years ago

- Processing power is low
- RAM is limited
- Data connections:
    - Intermittent, low bandwidth, high latency

POKéMON GO™


Singtel
3:31 PM
83%
8


●●●●● Singtel
3:38 PM
81%
AR
Abra / CP 72
134


●●●●● Singtel
3:39 PM
81%
AR
Pidgey / CP 72
132

What it means to Enterprise Environment

Some of the Challenges in Enterprise Environment

- multiple platforms (iOS, Android, etc)
- bulk device management and security
- server side database retrieval, caching
- big and complicated project -> team of developers, versioning, etc.
- timely response critical (mobile users are less forgiving)
- good user-interface
- Payment

Responsive, Effective, Transactions, Secure

More about this course …

- Focus on basic skill sets on Android Native and Hybrid

- Includes the following
  - skill set to create a basic app (both native and Hybrid)
  - communicating with server through internet
  - parsing data obtained through http request
  - QR Codes
  - maps
  - using sensors eg. location, camera
  - user interface (including graphics)

What is a Mobile App ?

Applications can reside in Server or in Client

Resident in Server: Web applications optimized for mobile devices (we don't call these mobile apps)

Resident in Client: Native Apps / Hybrid Apps $\Big\}$ **Mobile Apps**

Resident in Server

websites optimized for mobile devices

- Method 1

  Detect browser capabilities

  - WURFL (framework used by Facebook) for mobile device detection.
  - ASP.NET can be configured to use WURFL
  - can group devices based on capabilities and serve accordingly
- Method 2

  use Responsive Web Design

Resident in Client

native mobile apps (Swift in iOS, Java in Android)

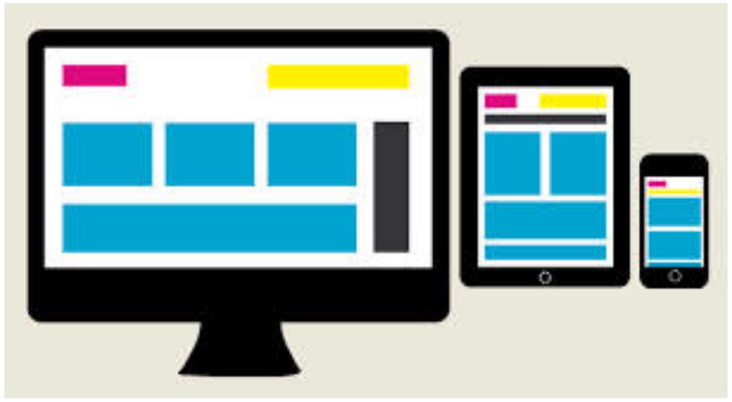
or

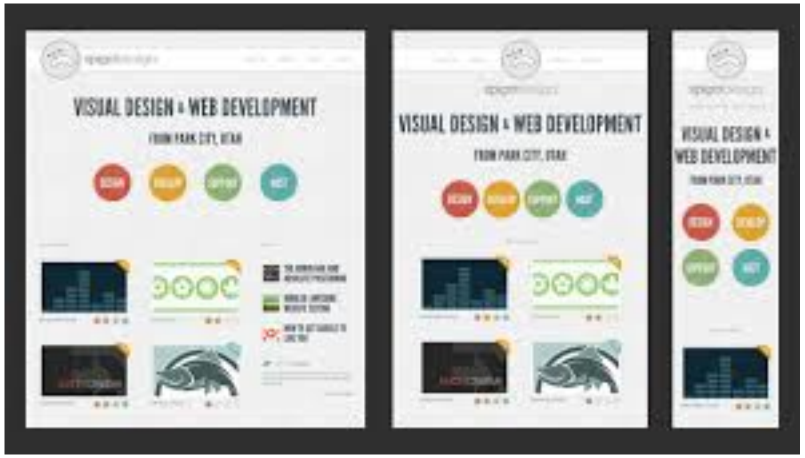
Hybrid mobile apps (HTML5+JavaScript + CSS3)

Web Applications

(Hybrid apps development can benefit from web application techniques)

Responsive Web Design



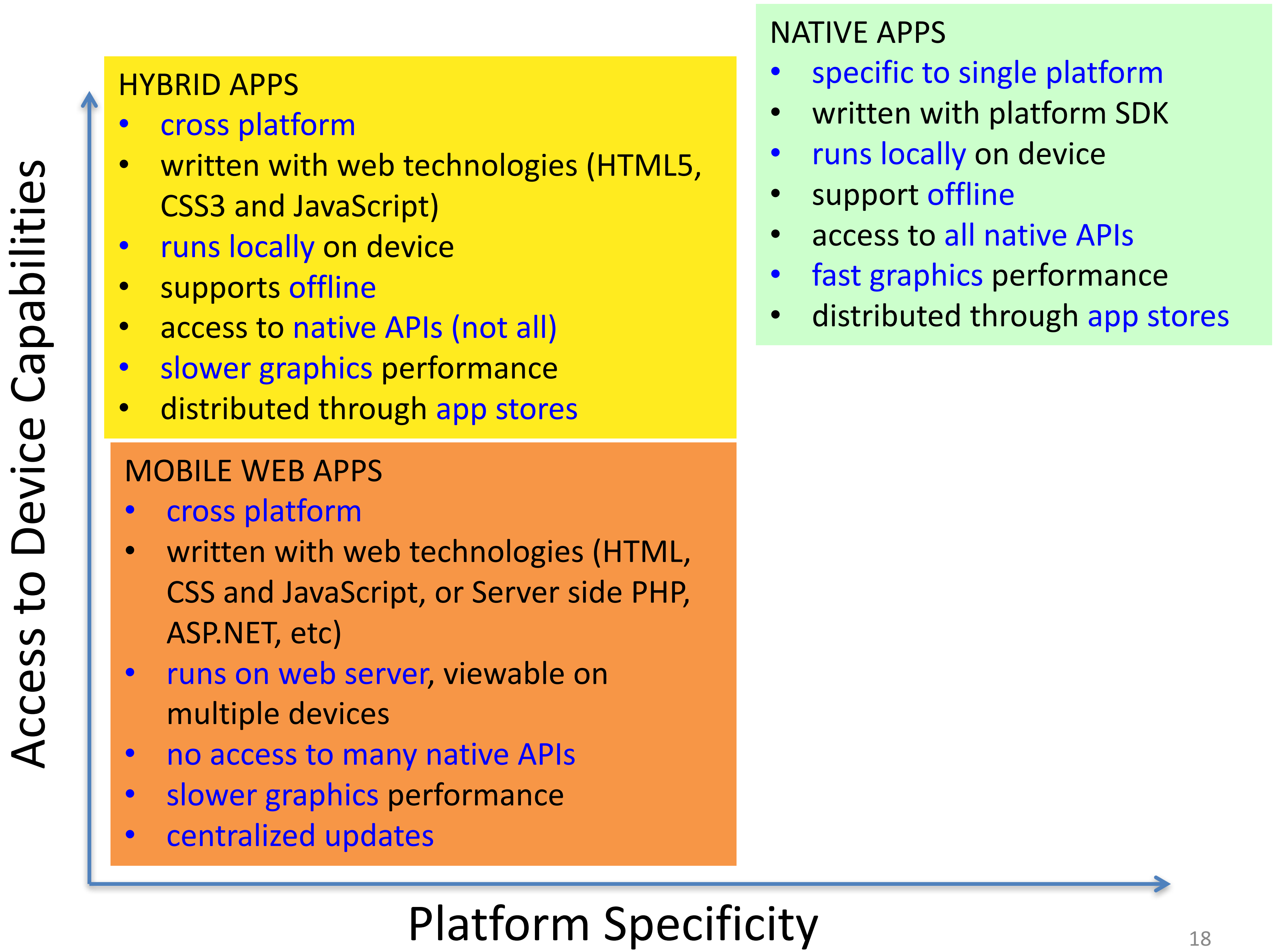


http://designmodo.com/responsive-design-examples/

http://colly.com
http://www.anderssonwise.com
http://stephencaver.com
http://foodsense.is

etc.

Mobile Apps:
Native Apps & Hybrid Apps



NATIVE APPS

- specific to single platform
- written with platform SDK
- runs locally on device
- support offline
- access to all native APIs
- fast graphics performance
- distributed through app stores

HYBRID APPS

- cross platform
- written with web technologies (HTML5, CSS3 and JavaScript)
- runs locally on device
- supports offline
- access to native APIs (not all)
- slower graphics performance
- distributed through app stores

MOBILE WEB APPS

- cross platform
- written with web technologies (HTML, CSS and JavaScript, or Server side PHP, ASP.NET, etc)
- runs on web server, viewable on multiple devices
- no access to many native APIs
- slower graphics performance
- centralized updates

2 Strategies of Residence Apps in Client:

- HTML5 + CSS + JavaScript
  - good if you need to target many different platforms
  - not as efficient but suffices in many cases
  - Ionic
    - an HTML5+CSS+Javascript mobile app development framework for building hybrid mobile apps

- Native codes
  - Swift for iOS
  - Java for Android

What is a Hybrid App ?

- runs on the mobile device
- written with web technologies (HTML5, CSS and JavaScript)
- runs inside a native container
- leverage the device's browser rendering engine (but not the browser) to render the HTML and process the JavaScript locally
- use a web-to-native abstraction layer to access device capabilities that are not accessible in mobile web applications, such as accelerometer, camera, and local storage

WebKit is browser rendering engine for iOS, Android, and Blackberry

- the abstraction layer exposes the device's native APIs to the hybrid app as a JavaScript API
- Apache Cordova (formerly known as PhoneGap) is an example of JavaScript abstraction layer

Note: Can access camera from HTML5 now, so may not be necessary to use Cordova. See

http://www.w3.org/TR/html-media-capture/

Both iPhone and Android phones use WebKit as their computational engine. This makes it possible to develop HTML5 apps and run on both iPhone and Android.

Why do we want to learn Hybrid ?

Write once – run all (almost)

- very attractive if need to serve users using different mobile platforms

Mobile Native Apps Basic Requirements:

- iOS
  - mac computer, Xcode
  - Need to be registered as Apple Developer

- Android
  - Windows PC (Windows 7 or later), Mac, Linux
  - IDE is Android Studio (Eclipse is no longer supported)
  - Don’t need to be a registered developer

- Windows
  - Windows PC, Visual Studio for Windows Phone

Fast Changing World
Do not be left behind !

- APIs deprecated fast
- new APIs
- surprises…

You need to capitalize on …

- internet search engines
  - you must develop the skill set (and attitude) to, eg. **Google**
- Online learning materials and discussion forums etc
- Learning from your fellow apps developers

References

“Android Application Development Cookbook: 93 Recipes for Building Winning Apps” by Wei-Meng Lee

“iPhone/iPad Apps Development” by Kazuhiro Furuhata
(We will refer to examples in this book. English version does not exist)

+ many valuable online resources …

Some soft skills that are also important :

DESIGN

Human Computer Interface
iOS and Android

The materials in this set of slides are taken from the following:

Ref:
https://developer.apple.com/library/ios/documentation/userexperience/conceptu
al/mobilehig/
Ref: http://developer.android.com/design/get-started/principles.html

Design Principles

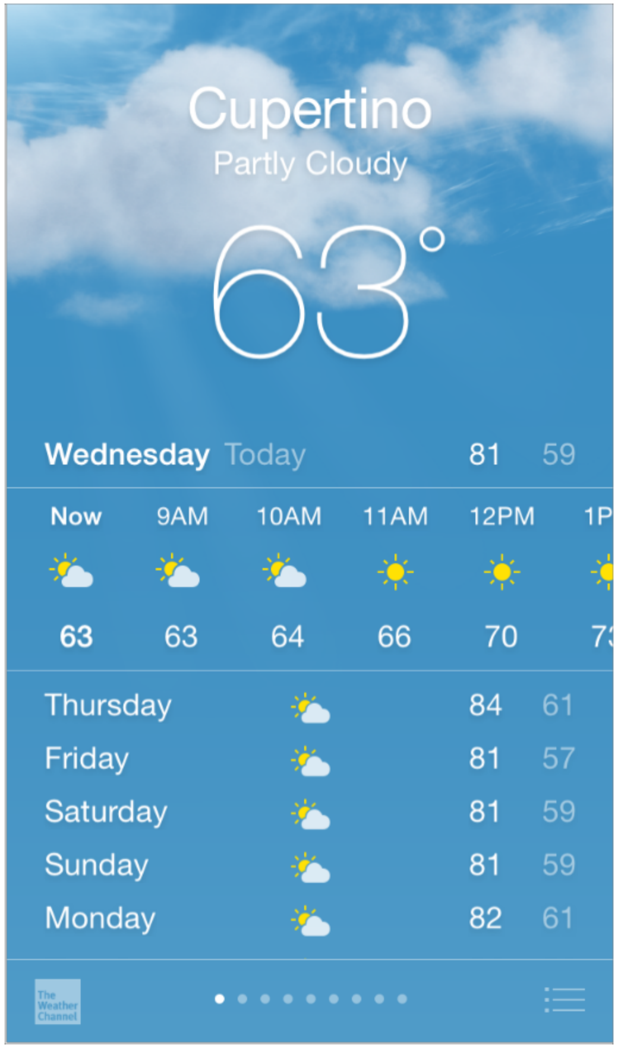
- Content > UI
  - The UI helps users understand and interact with the content, but never competes with it

- Clarity
  - Text is legible at every font size
  - Icons are precise and easy to understand
  - Decorations are subtle and appropriate

Design Methodology

- First, strip away the UI to expose the app’s core functionality and confirm its relevance
- Next, include the important details and useful decorations back. Discard the rest.
- At all time, use content and functionality to motivate every design decision

Some examples …



Cupertino

Partly Cloudy

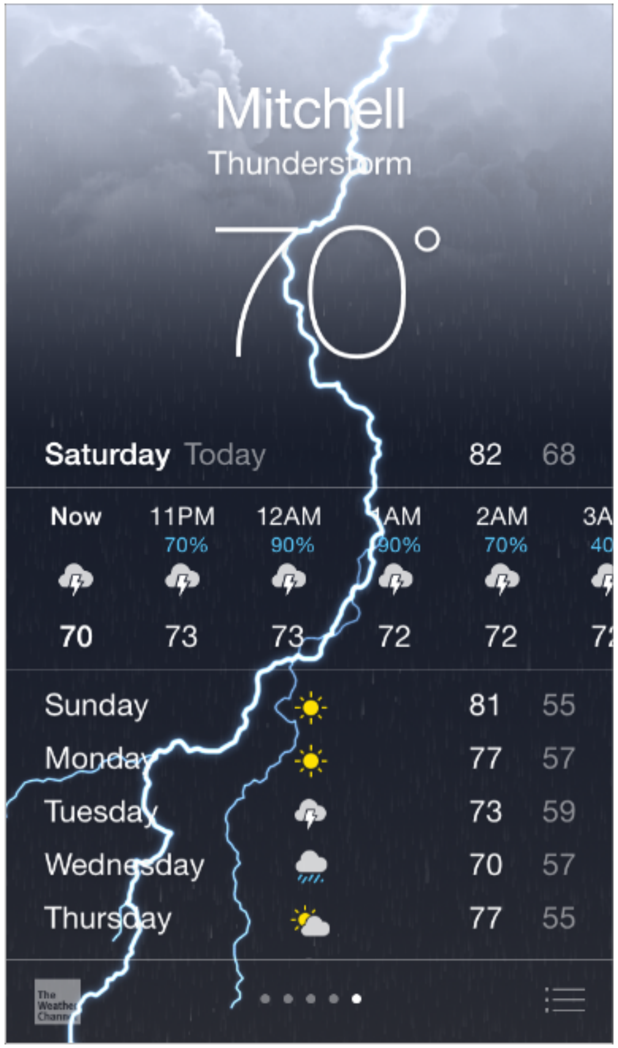
63°

Wednesday Today 81 59

Now	9AM	10AM	11AM	12PM	1P
63	63	64	66	70	7

Thursday	84	61
Friday	81	57
Saturday	81	59
Sunday	81	59
Monday	82	61

Blue sky with cloud



Mitchell

Thunderstorm

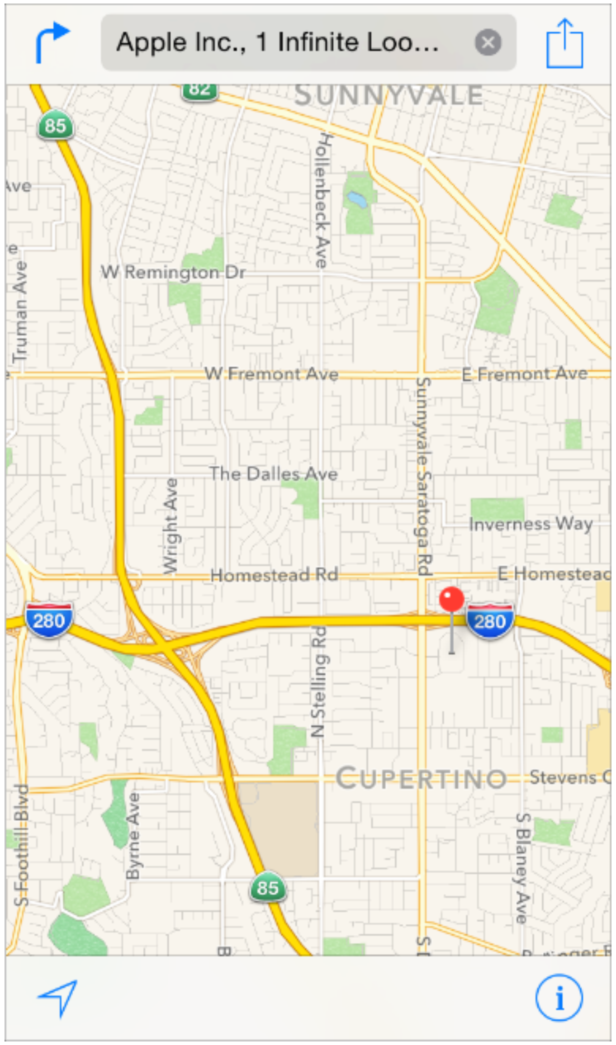
70°

Saturday Today 82 68

Now	11PM	12AM	1AM	2AM	3A
	70%	90%	90%	70%	40
70	73	73	72	72	72

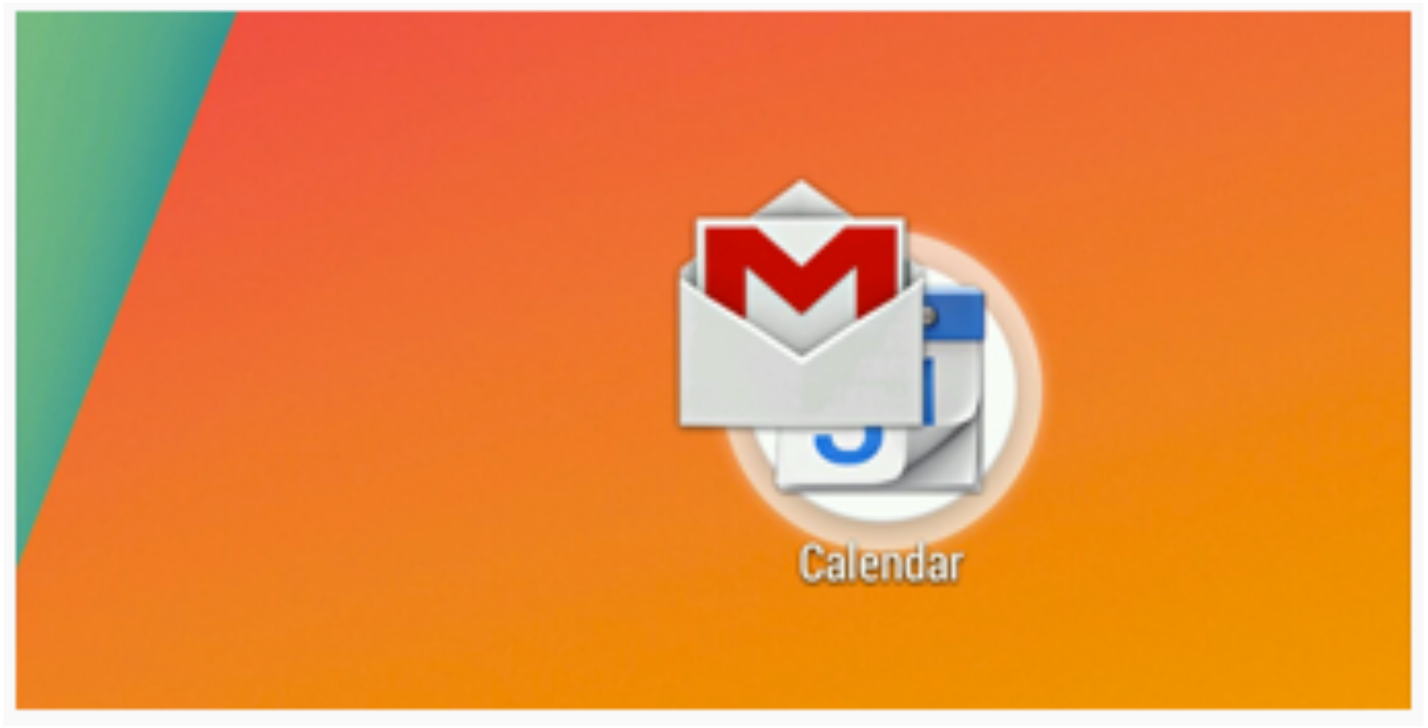
Sunday	81	55
Monday	77	57
Tuesday	73	59
Wednesday	70	57
Thursday	77	55

Thunderstorm

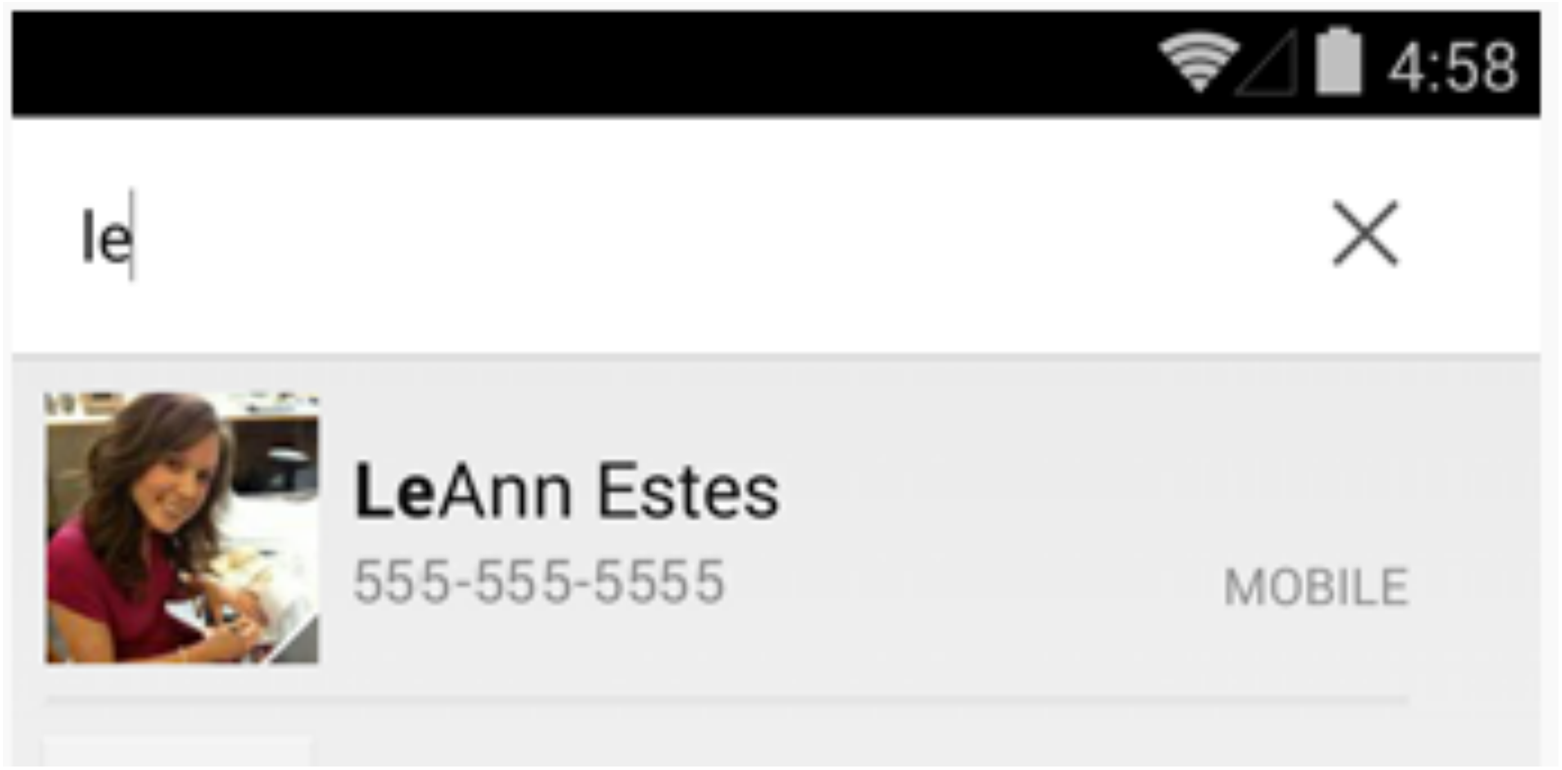


Positioning without overwhelming with shadows

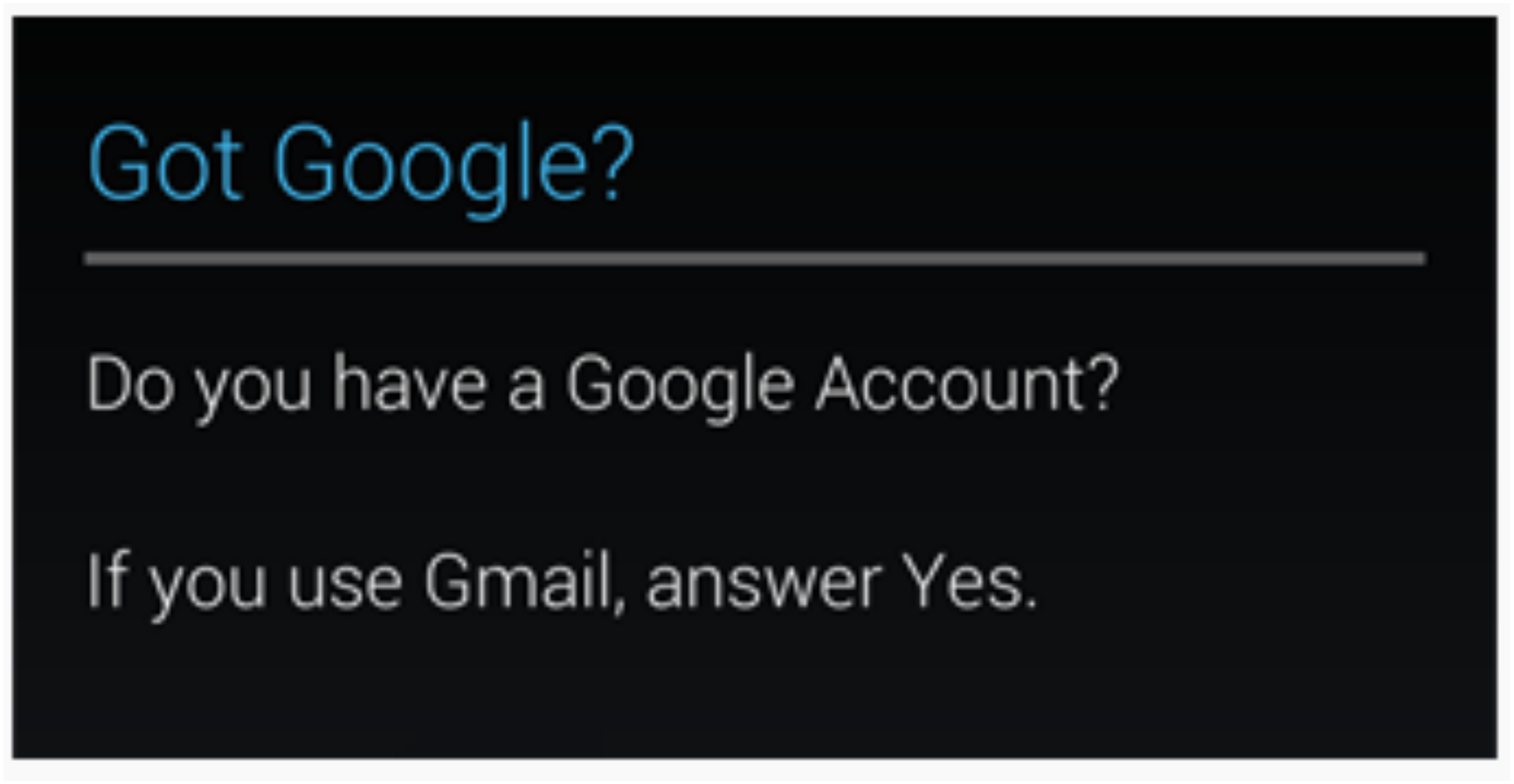
Real objects are more fun than buttons and menus:



Learn peoples' preferences over time. Place previous choices within easy reach.



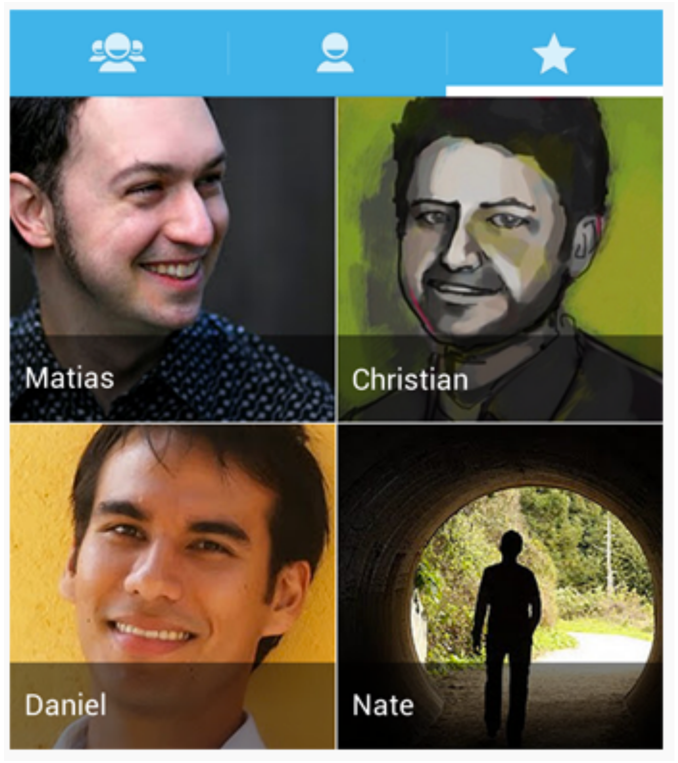
Keep it brief: Use short phrases with simple words.


Got Google?

Do you have a Google Account?

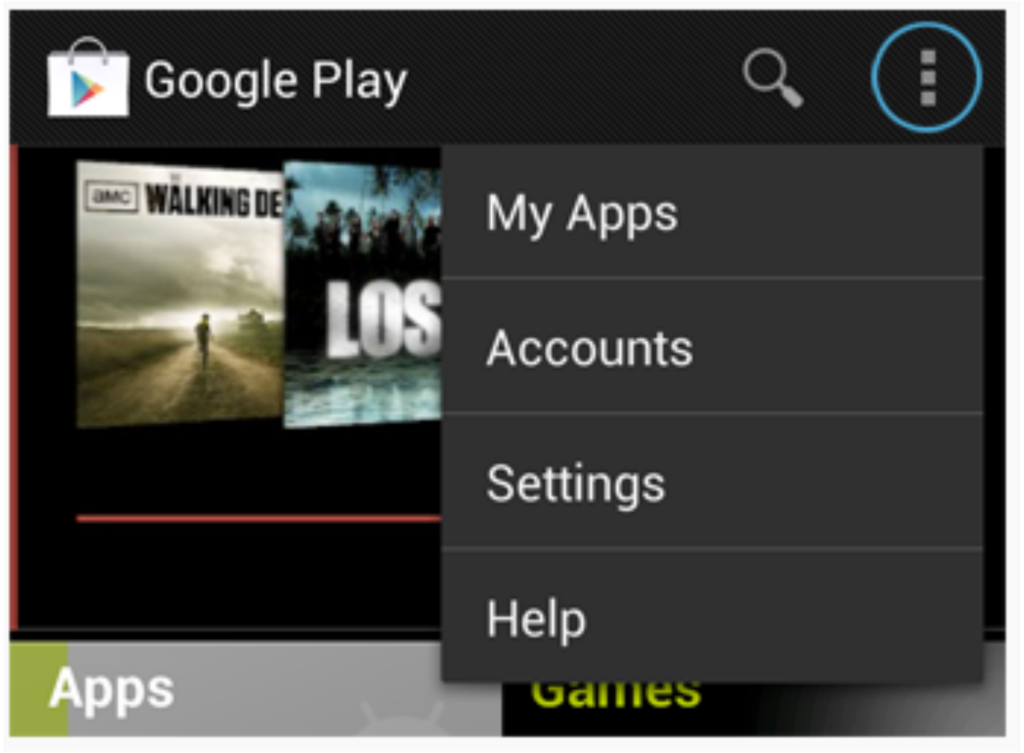
If you use Gmail, answer Yes.

Use pictures to explain ideas



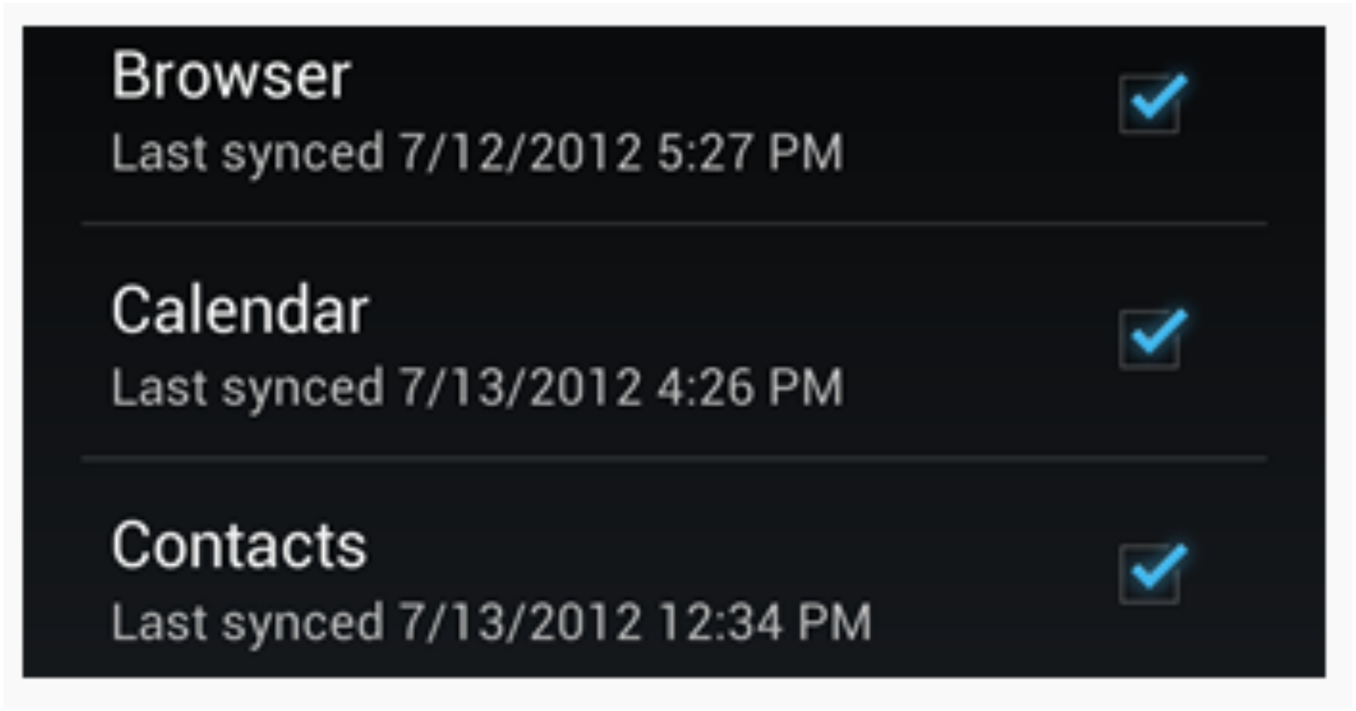
Only show what I need when I need it:

Break tasks and information into small, digestible chunks. Hide options that aren't essential at the moment, and teach people as they go.



Never Lose My Stuff:

Remember settings, personal touches, and creations across phones, tablets, and computers. It makes upgrading easier.

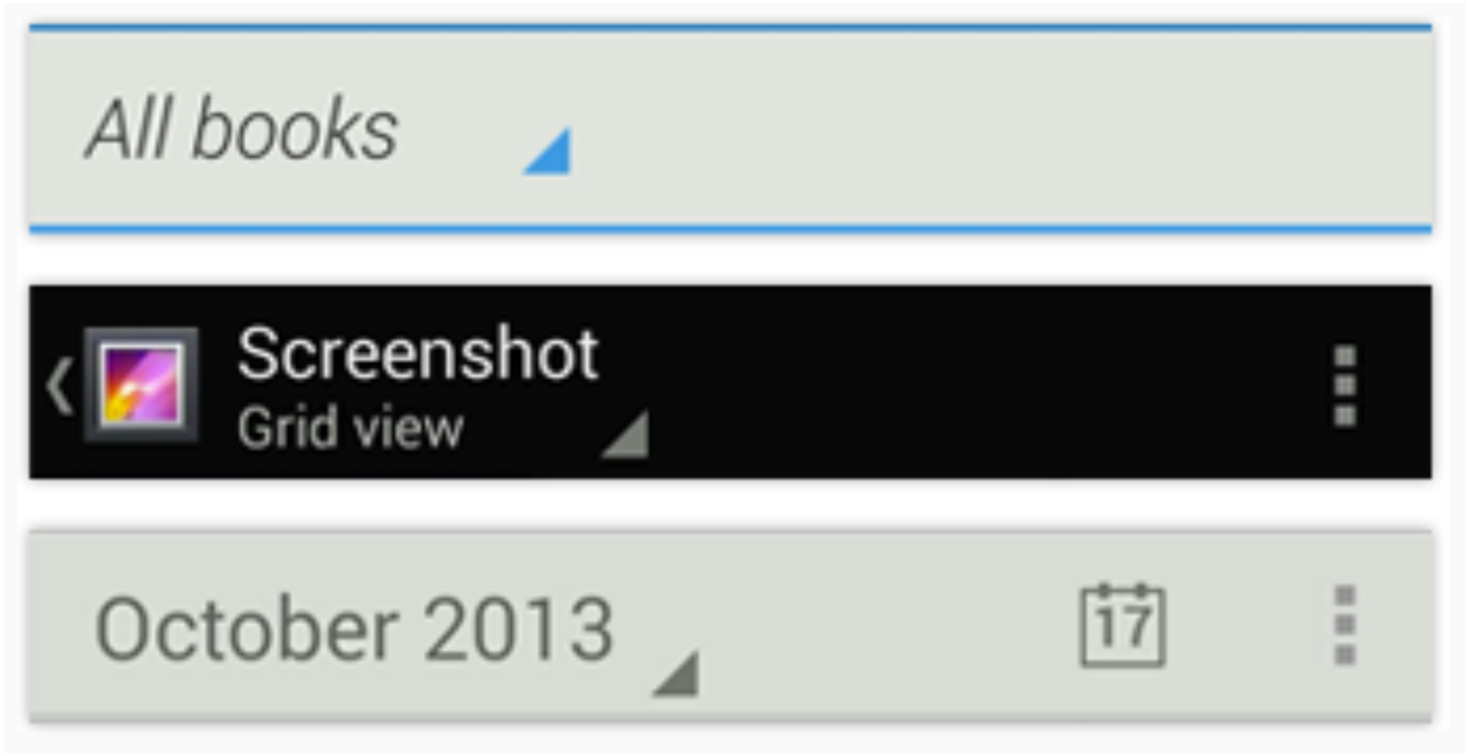


Browser ✓
Last synced 7/12/2012 5:27 PM

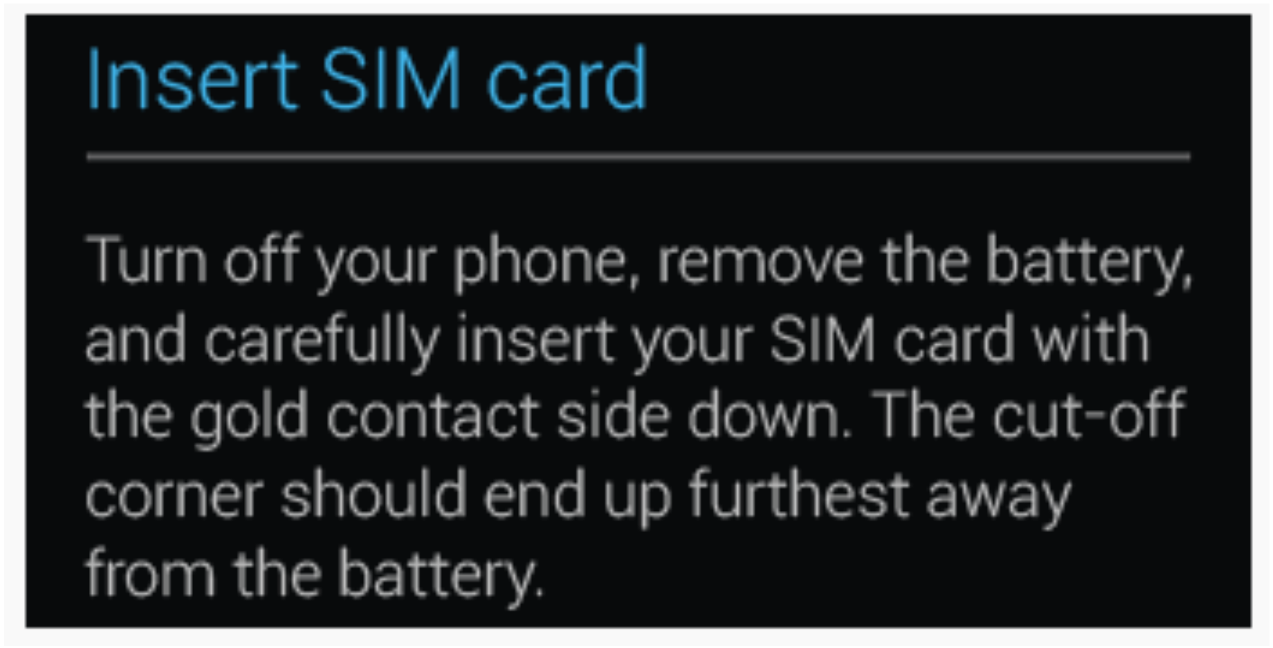
Calendar ✓
Last synced 7/13/2012 4:26 PM

Contacts ✓
Last synced 7/13/2012 12:34 PM

If it looks the same, it should act the same



Give clear instructions but don't make people feel stupid.

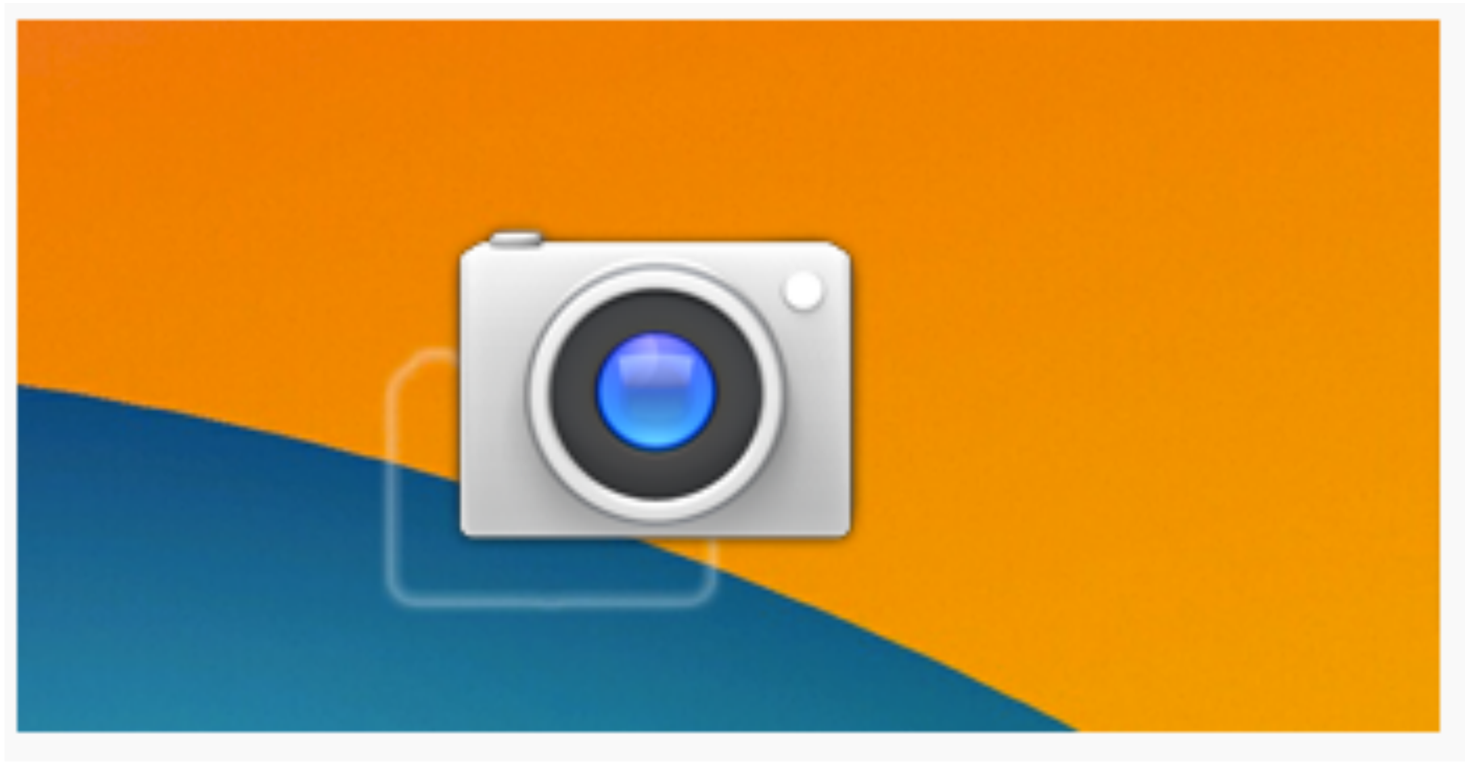


Insert SIM card

Turn off your phone, remove the battery, and carefully insert your SIM card with the gold contact side down. The cut-off corner should end up furthest away from the battery.

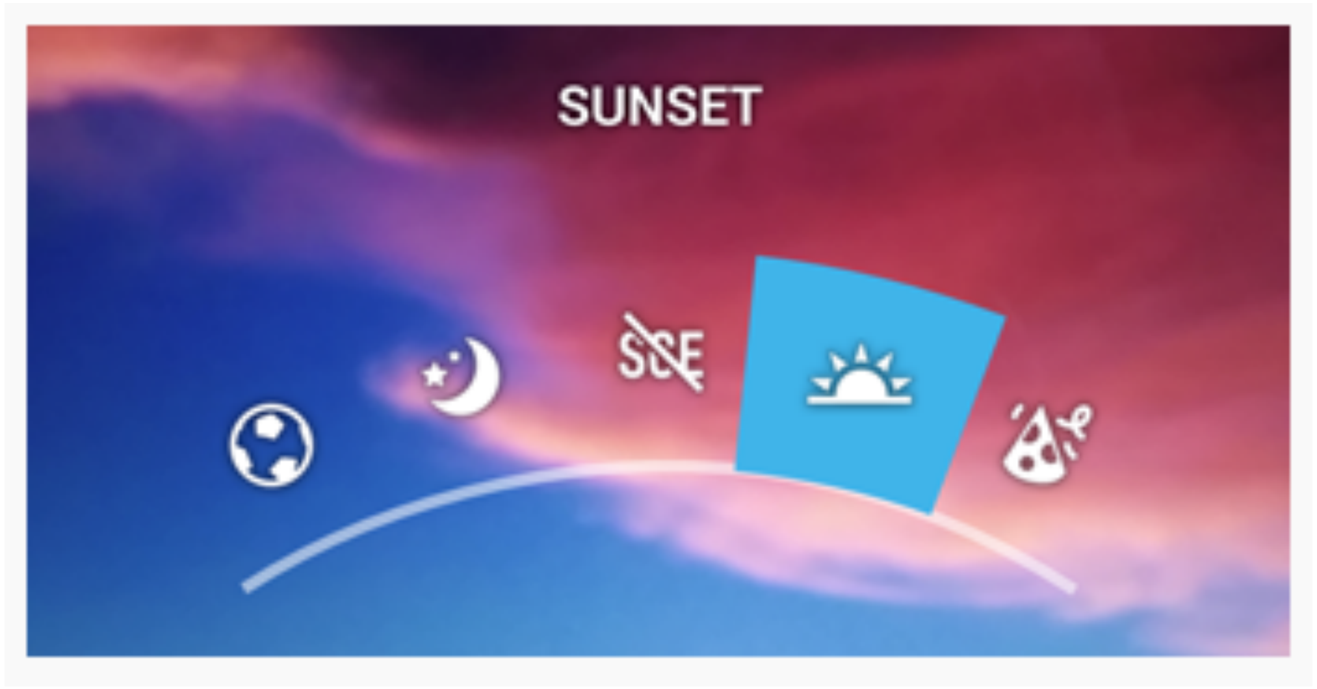
Sprinkle encouragement:

Break complex tasks into smaller steps that can be easily accomplished. Give feedback on actions, even if it's just a subtle glow.



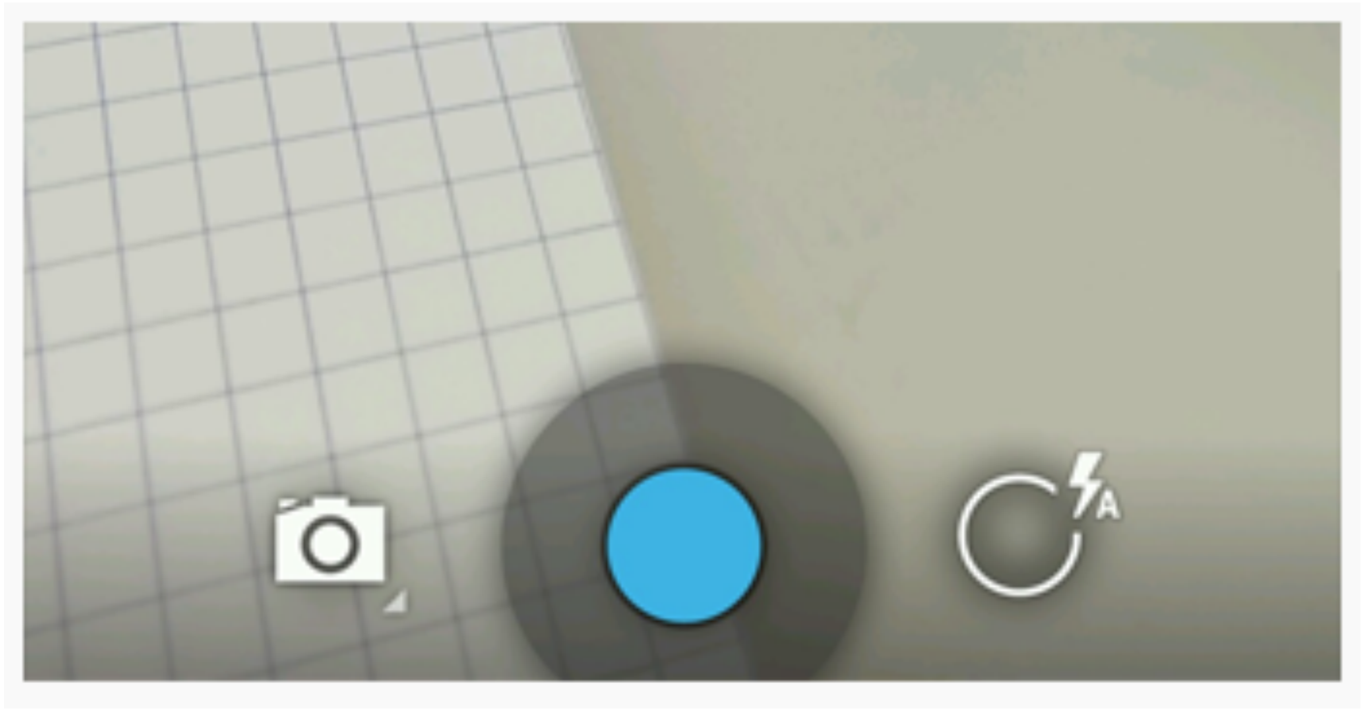
Do the heavy lifting for me:

Make novices feel like experts by enabling them to do things they never thought they could. For example, shortcuts that combine multiple photo effects can make amateur photographs look amazing in only a few steps.

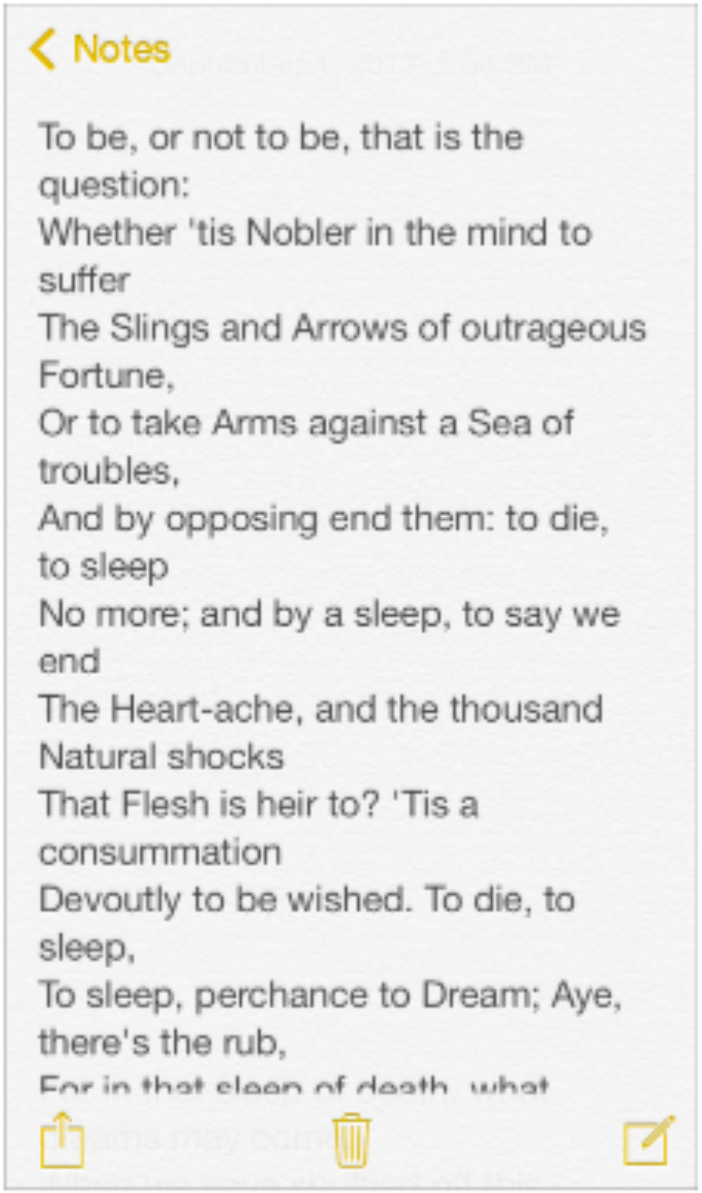


Make important things fast:

Not all actions are equal. Decide what's most important in your app and make it easy to find and fast to use, like the shutter button in a camera, or the pause button in a music player.



Let Colour Simplify the UI



To be, or not to be, that is the question:
Whether 'tis Nobler in the mind to suffer
The Slings and Arrows of outrageous Fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
The Heart-ache, and the thousand Natural shocks
That Flesh is heir to? 'Tis a consummation
Devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream; Aye, there's the rub,
For in that sleep of death, what

A key color—such as yellow in Notes—highlights important state and subtly indicates interactivity. It also gives an app a consistent visual theme. Use a family of pure, clean system colors that look good at every tint and on both dark and light backgrounds.