ECWM506

Week 11

Android and iOS development

Considerations

Why both?

- Quite simply to reach a bigger market
- iOS and Android currently dominate the smartphone market
- Millions of users
- Both have reasonably good publishing models and app stores

Why not both?

- More expensive to develop for two platforms
- Significant differences in UI so impacts design
- Different code base
- May only have expertise in 1 area
- If a small developer then danger of only producing an average app for both platforms instead of an amazing app for one

Development

- Design
- Design should be generic for analysis and most of the design

No need for iOS/Androids team here

Standard design languages should be employed such as UML etc.

Problem definition

- Generic with maybe reference to smartphones in general and smartphone capability etc. Should just be describing the problem, functionality and environment.

‘A system shall be created that allows users to…’

Generic Design - Requirements

- Requirements
- Should be identical in both cases
- These should be entirely platform agnostic
  - ‘The software shall allow the user to send a message to another user’
- Should NOT mention platforms or platform technology

Analysis

- Generic – no need to discuss platforms
- Use case analysis, scenarios
- Class diagrams (Domain model) – generic and no reference to app or platform or servers etc.
  - Should be domain entitles only
- Collaboration diagrams
  - Sequence – generic MVC

Data Design

- Be generic but…
  - Think in objects and tables
  - Generic data dictionary
- Table are no use for iOS as you will use Core Data which an Object Relational Mapping framework, this is a rich and sophisticated framework
- Standard SQL tables are of more use to Android

Detailed Design

- This is the refinement of the initial analysis
  - Can still be generic – we are not coding yet
  - MVC is very strongly implemented in IOS
  - Android is more MVP and even that is quite vaguely implemented
  - This means you might now be producing 2 designs

    MVC for iOS (very clear)

MVP (Activities, creating a Presenter, models) – you will need to now look at Android developer patterns

Data Options

- Preferences (not sharable on iOS)
  - Store private primitive data in key-value pairs (e.g. NSUSerDefaults for iOS)
- Internal Storage
  - Store private data on the <u>device</u> memory.
- External Storage
  - Store data on the shared external storage (iCloud, google docs etc)
- SQLite Databases (core data ORM on iOS)
  - Store structured data in a private database.
- Network Connection
  - Store data on the web with your own network server.

Data Options

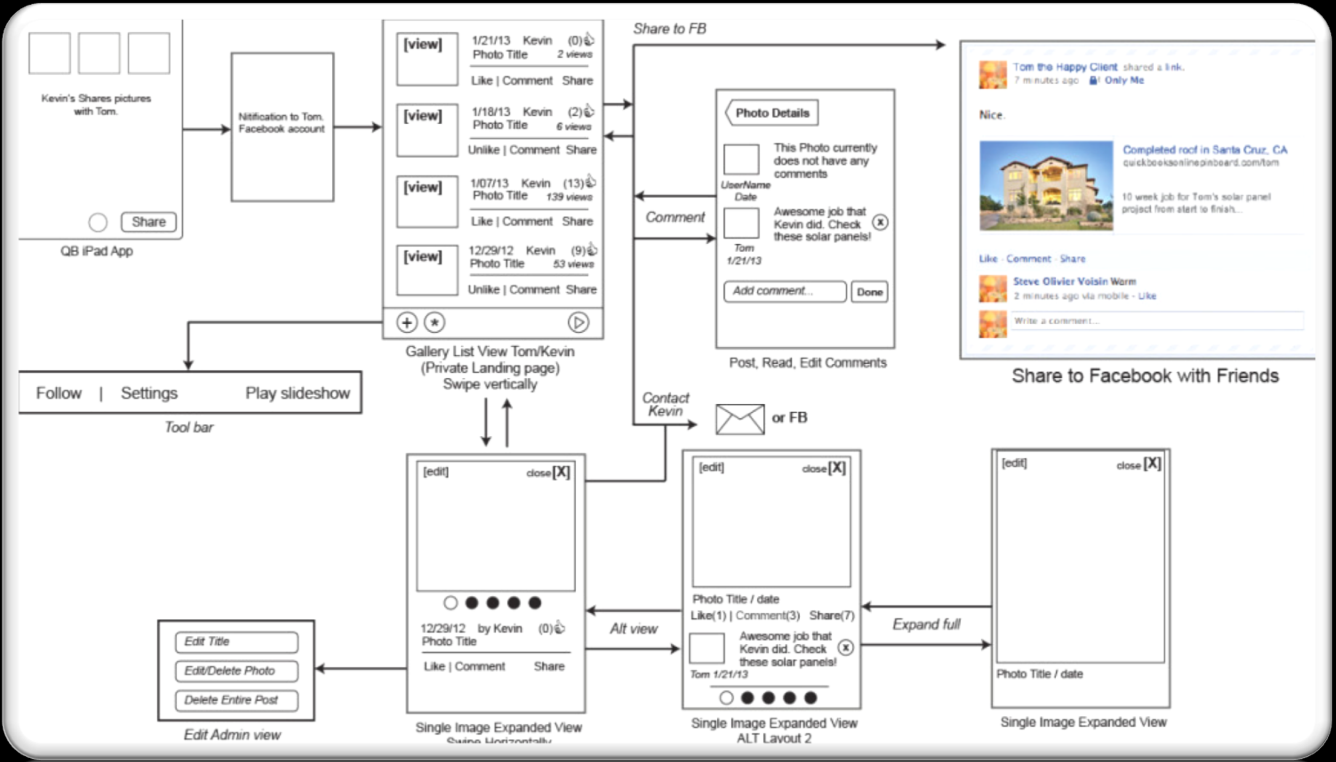
- Difference
  - You CANNOT share your apps data with another directly within iOS as each app is ‘sandboxed’ for security reasons
  - You CAN on Android

Testing

- Tests can be on the whole generically specified
- You are testing the requirements and functionality and these are independent of platform
- UI is at least partly platform specific so will need to think about each platform once the generic UI tests have been created

Storyboarding

- Can be generic initially


Kevin's Shares pictures with Tom.
Share
QB iPad App
Nitification to Tom. Facebook account
[view] 1/21/13 Kevin (0) Photo Title 2 views Like | Comment Share
[view] 1/18/13 Kevin (2) Photo Title 6 views Unlike | Comment Share
[view] 1/07/13 Kevin (13) Photo Title 139 views Like | Comment Share
[view] 12/29/12 Kevin (9) Photo Title 53 views Unlike | Comment Share
Gallery List View Tom/Kevin (Private Landing page) Swipe vertically
Follow | Settings Play slideshow
Tool bar
Share to FB
Comment
Photo Details
This Photo currently does not have any comments
UserName Date
Awesome job that Kevin did. Check these solar panels!
Tom 1/21/13
Add comment... Done
Post, Read, Edit Comments
Tom the Happy Client shared a link.
7 minutes ago · Only Me
Nice.
Completed roof in Santa Cruz, CA
10 week job for Tom's solar panel project from start to finish.
Like · Comment · Share
Steve Olivier Voisin Warm
2 minutes ago via mobile · Like
Write a comment...
Share to Facebook with Friends
Contact Kevin
or FB
[edit] close [X]
12/29/12 by Kevin (0) Photo Title
Like | Comment Share
Single Image Expanded View Swipe Horizontally
Alt view
Photo Title / date
Like(1) | Comment(3) Share(7)
Awesome job that Kevin did. Check these solar panels!
Tom 1/21/13
Single Image Expanded View ALT Layout 2
Expand full
Photo Title / date
Single Image Expanded View
Edit Title
Edit/Delete Photo
Delete Entire Post
Edit Admin view

Specific UI design

- The current iOS and Android UI are different
- You will need a specific design for each platform that will be based on your generic storyboarding
- You will also need to have an understanding of the various UI components
- E.g. UITableViewController in IOS, ListView in Android etc.

Services

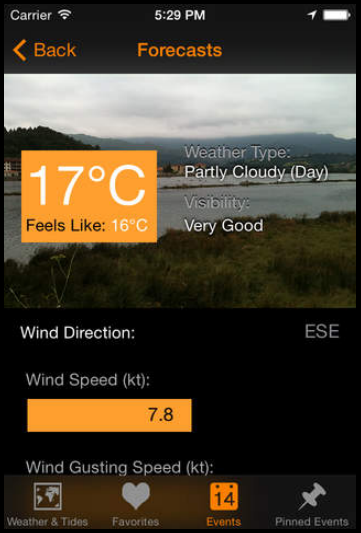
- Totally generic
  - Use Web Service Wrappers
    - Databases
    - Information services
  - In general use JSON as a data exchange format
  - Create simple platform agnostic API
  - If you use external services (written by third parties) then you might need to be flexible

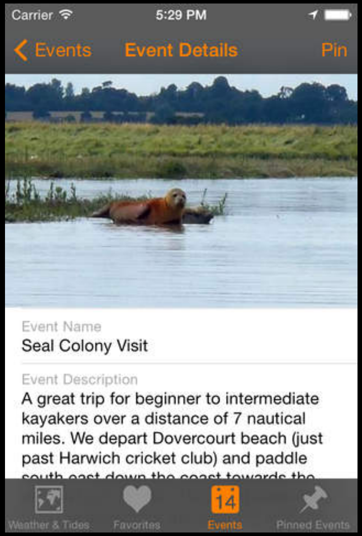
Example - iKayaker Services

App

MapKit

Notifications





Event Name
Seal Colony Visit

Event Description
A great trip for beginner to intermediate kayakers over a distance of 7 nautical miles. We depart Dovercourt beach (just past Harwich cricket club) and paddle south east down the coast towards the

iKayaker Back-end web services (JSON)

Push Message Service

Weather service (web scrape)

Weather service (web scrape)

Facebook API

External Services

Met Office Data

Tide Data

FaceBook

Apple Push Server

Could use Cross Compilers

- Cross compilers exist such (as Flash, Unity, Xamarin etc.) See: https://www.udemy.com/blog/cross-platform-mobile-development/ for more
- Good because less code but despite all claims these are not as good as coding individually for each platform (except possibly Unity as specific to games)
- Need to assess on case by case basis

Working with the UI

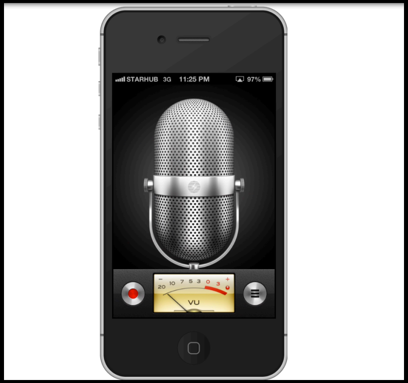
- You may have an existing iOS or Android app and now want to port it to the other platform
- Choices
  - Straight port – this is where is essentially copy your current design as much as possible
  - Custom port – redesign the UI to best suit the target platform and client UI expectations (this is the best choice and probably not more work in the long run)

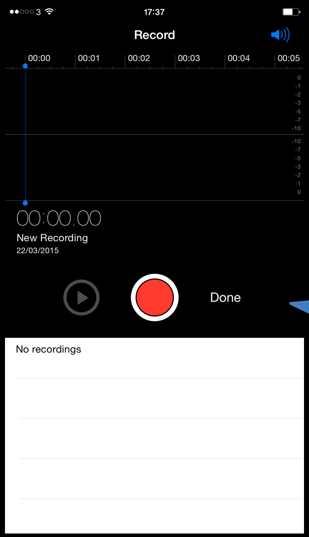
UI Differences

- With iOS, we are accustomed to apps which place high emphasis on both functionality and aesthetics
- The skeuomorphistic views (realism) have gone since iOS7 but the design philosophy is the same

UI Style

- Apple places emphasis on its 'tapworthy' areas, by drawing the eye to the actionable buttons with design

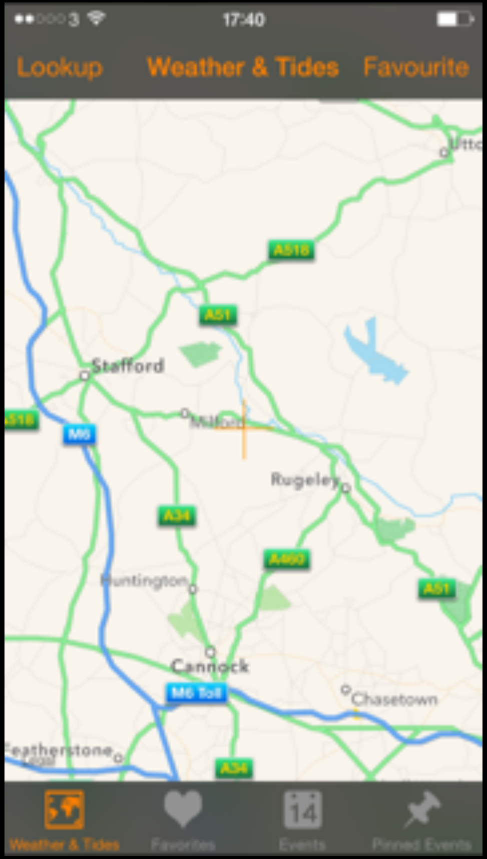




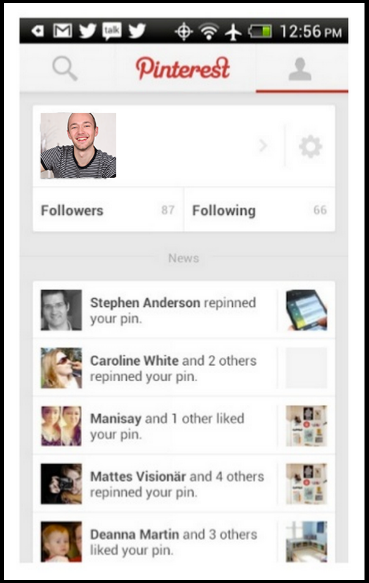
Apple's new style has become much more minimalist

Tab navigation

- A popular pattern is iOS is the tab bar



- Android is more tap-anywhere ideologies & minimal design styling.
- With fragmentation (with 1000’s devices & counting), fluid and responsive design is essential



Followers 87 Following 66

News

Stephen Anderson repinned your pin.

Caroline White and 2 others repinned your pin.

Manisay and 1 other liked your pin.

Mattes Visionär and 4 others repinned your pin.

Deanna Martin and 3 others liked your pin.

Navigation

- Apple also ensures they give you a clear indication on every page you are on
- They do so with push view controllers
- A back button indicating where you came from is always present. A page label with a short snippet that represents the screen you are currently on will be at the top
- Altogether, this arrangement gives you a clear indication of where you are at all times, and no matter how deep you are into the app
- Apple typically recommends 3 levels in

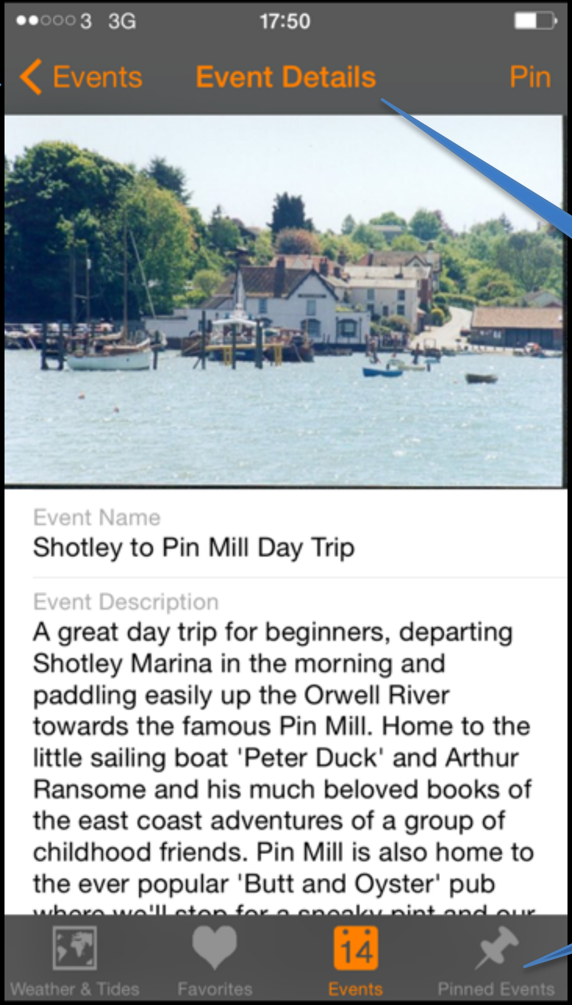
iOS Navigation Example

Button to navigate back

buttons

Page you are currently on

Tabs to take you different aspect of the app



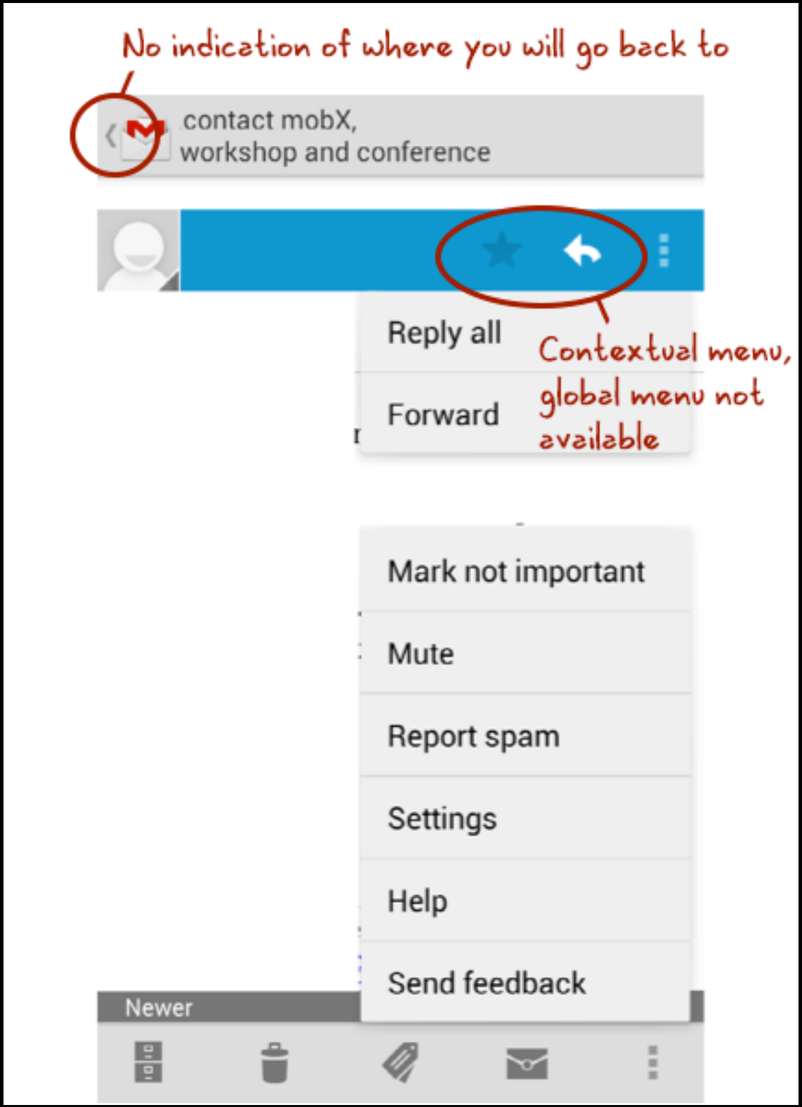
Event Name
Shotley to Pin Mill Day Trip

Event Description
A great day trip for beginners, departing Shotley Marina in the morning and paddling easily up the Orwell River towards the famous Pin Mill. Home to the little sailing boat 'Peter Duck' and Arthur Ransome and his much beloved books of the east coast adventures of a group of childhood friends. Pin Mill is also home to the ever popular 'Butt and Oyster' pub

Android example

Think Global, Act Local.

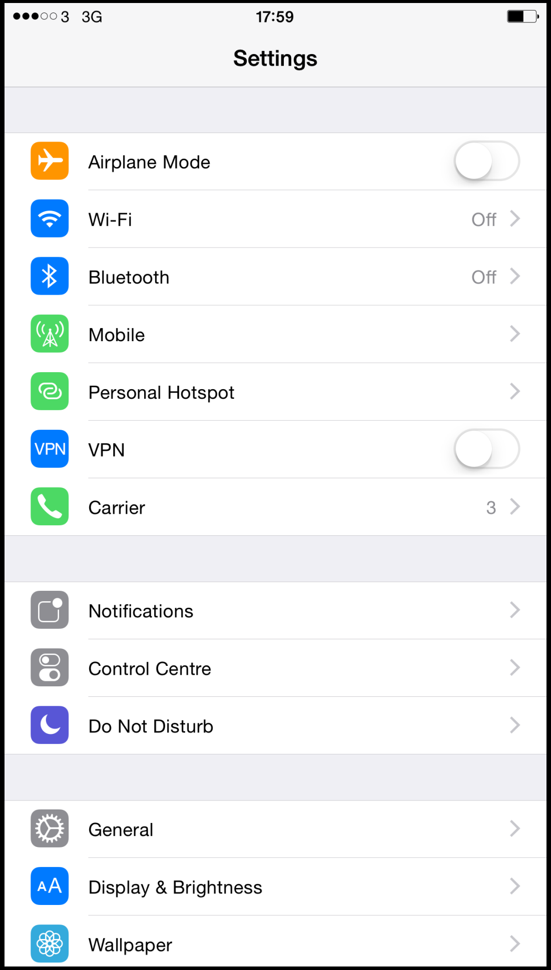
This approach entails removing indicators of your location on the view you are currently in. There will neither be indicators informing you of which page you came from, nor would you have access to the main menu from the content pages. Instead, on every page, it is replaced with contextual menus & buttons with which the user is expected to take with the info they have on the current view. Users will have to get used to the fact that to get to the main menu, they will need to track back all the way to the main page.


No indication of where you will go back to
contact mobX, workshop and conference
Reply all
Forward
Contextual menu, global menu not available
Mark not important
Mute
Report spam
Settings
Help
Send feedback
Newer

- User Experience

- In iOS, arrows etc. act as indicators to inform us which areas are tappable, and have hidden content – it is very clear what is tappable

Tappable areas in iOS



●●●○○ 3 3G 17:59

Settings

Airplane Mode

Wi-Fi Off

Bluetooth Off

Mobile

Personal Hotspot

VPN

Carrier 3

Notifications

Control Centre

Do Not Disturb

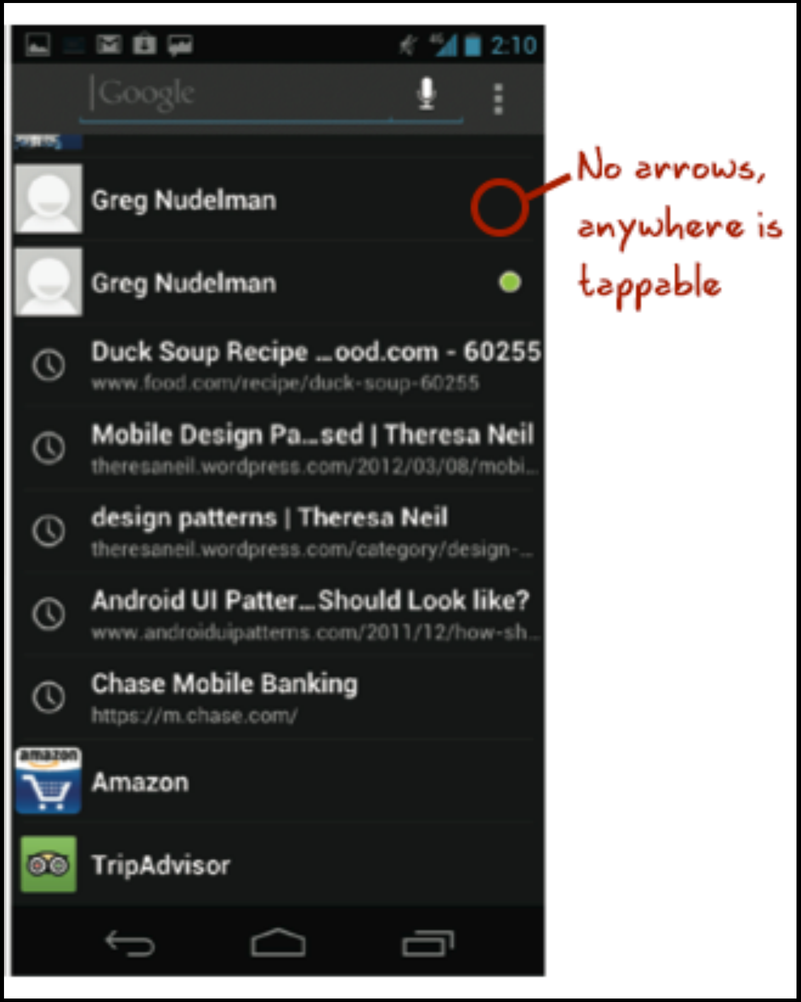
General

Display & Brightness

Wallpaper

Android

With Android, we have to get used to the fact that you-can-tap-anywhere. This is especially apparent on their table cells. Arrow indicators are removed, as the design principle behind it is that almost anything on the screen is tappable. Therefore, careful considerations have to go into the placement of buttons and content, as the UI does not have a tell tale sign what are the call to action. Instead, users will allow their eyes and fingers to wander the page until something catches their eye, and they will tap it. Therefore the flow and hierarchy of information should make sense right from the get go, as their eyes scan the content.


2:10
Google
Greg Nudelman
Greg Nudelman
Duck Soup Recipe ...ood.com - 60255
www.food.com/recipe/duck-soup-60255
Mobile Design Pa...sed | Theresa Neil
theresaneil.wordpress.com/2012/03/08/mobi...
design patterns | Theresa Neil
theresaneil.wordpress.com/category/design-...
Android UI Patter... Should Look like?
www.androiduipatterns.com/2011/12/how-sh...
Chase Mobile Banking
https://m.chase.com/
Amazon
TripAdvisor
No arrows, anywhere is tappable

UI Summary

- This is small sample if UI differences and there are many more such as ‘home screen’, ‘action bars and sheets’, ‘search bar placement’, ‘sections’, ‘pickers’ ‘ button and words for buttons’, ‘icons’ ‘drawer menus (not standard on IOS though popular), ‘segmented controls’, ’split action bars’ etc.

Summary

- We have gone through some default UI elements that makes Android design different from iOS
- Therefore, if designing for an Android or iOS interface, we should carefully go through the app information architecture, and think through what are the UI elements we can use that will be most relevant for your content
- instead of porting the exact design from Apple to Android or vice-versa

Design summary

- Clearly after some generic design we have to become specific
- Implementation is entirely specific (ignoring cross-compilers)
- The languages and tools and different
  - IOS – Objective C, C, Swift
  - Android – Java
- Tools
  - iOS – XCode (mature and sophisticated tool)
  - Android - either Eclipse or similar or Android Studio (less mature tool but improving)

End of lecture