Comp 388/422 - Software Development for Wireless and Mobile Devices

Fall Semester 2015 - Week 5

Dr Nick Hayward

Contents

Cordova Design - architecture

intro

- quickly recap the architecture and design behind a Cordova Native application
- Cordova effectively consists of the following components
  - *source code to allow us to build a native application container*
  - *specific to the mobile platforms we choose to add to our project, eg: Android*
  - *a collection of various APIs, implemented by Cordova as plugins*
  - *web application running within the container*
  - *access to native device functionality, APIs, and applications*
  - *provides a useful set of tools that help us manage our projects*
  - *creating a project, project files...*
  - *manage required plugins*
  - *build native applications using the native SDK*
  - *testing of applications using emulators, simulators...*

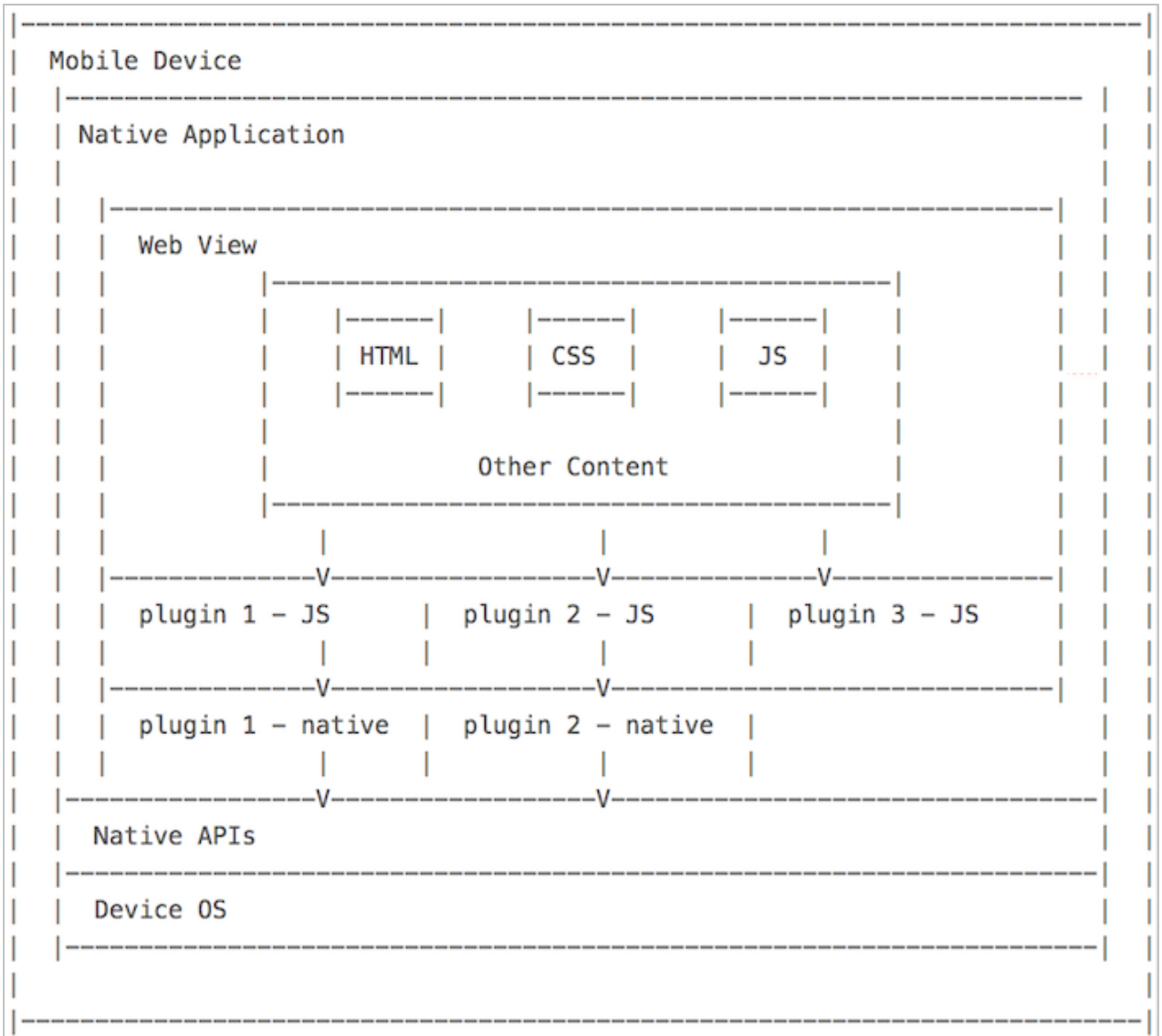
Cordova Design - architecture

native APIs

- benefit of working with Cordova is access to device
  - *installed applications, native APIs, underlying native hardware features...*
  - *camera, compass, contacts, sound, video...*
- provides access to this native functionality by using JavaScript APIs
  - *enables our web application to access native device capabilities*
  - *travel from the web content to the native device, and back again*
- APIs
  - *one for the web container and one for the native container*
- JavaScript library exposes the native capabilities to the web container
  - *uses the native code to access the required native part of the API*
- two ways we can manage plugins in Cordova
  - *Cordova CLi and Plugman*

Cordova Design - architecture

diagram



Cordova - Architecture

Cordova Design - architecture

JS plugins

- outline architecture includes the option for JavaScript only plugins
- JS plugins in Cordova normally a bridge from our web container to the native APIs
  - *useful way to expose native device functionality to the web application*
- use and develop plugins purely in JS
  - *add an existing library to help with data visualisations, graphics...*
- create our own focused plugins
  - *abstraction of application features and logic, other specific requirements...*
- greater support for native functionality at the web application level

Cordova Design - architecture

web container - part 1

- Cordova development uses many of the same underlying technologies as standard web application development
  - *a few limitations relative to network access that we need to consider*
- hybrid mobile application with Cordova
  - *a web application needs to be written as a self-contained application*
  - *needs to be able to run completely within web container on native device*
  - *Fetching external resources is not considered particularly good practice*
  - *external resources an issue if we lose a network connection*
- `index.html` file will normally be the only HTML file we use
  - *separate pages will be containers within this file*

Cordova Design - architecture

web container - part 2

- rethink our approach to building such web applications
  - *help us leverage the inherent capabilities of Cordova*
- self-contained applications need to ensure
  - *any application files and data are initially available*
  - *allows the application to launch and load on the native device without remote server*
  - *load the application and render the UI*
- application can then optionally fetch data
  - *remote server, API, search query, stream media...*

Cordova Design - building apps

SDKs and OSs - part 1

- build our Cordova applications
  - *including default Cordova APIs or additional APIs*
  - *each app has to be packaged into a native application*
  - *allows app to run on the host native device*
- each native SDK has its own set of custom or proprietary tools
  - *building and packaging their specific native applications*
- build our Cordova applications for a native device
  - *web content portion of app is added to a project*
  - *applicable to the chosen mobile platforms, such as Android, iOS, and Windows Phone*
  - *project is then built for each required platform*
  - *using Cordova CLI, for example*
  - *uses each of the applicable platform specific set of tools to help build*

Cordova Design - building apps

SDKs and OSs - part 2

mobile platform	supported desktop OS
Android	Linux, OS X, Windows
BlackBerry	Linux, OS X, Windows
Fire OS	same as Android (a few caveats...)
iOS	OS X
Ubuntu	Linux / Unix
Windows Phone	Windows

- Cordova API - Platform Guides

Cordova App - CLI recap

build initial project

```
cd /Users/ancientlives/Development/cordova
cordova create basic com.example.basic 422Basic
cd basic
```

- creates new project ready for development

```
cordova platform add android
cordova build
```

- adds support for native SDK, Android
- then builds the project ready for testing and use on native device

```
cordova emulate android
```

- outputs current project app for testing on Android emulator

Cordova App - structure recap

app directory

- quick recap of app's structure
- new project includes the following default structure

```
|- config.xml
|- hooks
|- README.md
|- platforms
   |- android
   |- platforms.json
|- plugins
|  |- android.json
|  |- cordova-plugin-whitelist
|  |- fetch.json
|- www
|  |- css
|  |- img
|  |- index.html
|  |- js
```

- initially, our main focus will be the www directory

Cordova App - structure recap

www directory

```
|- www
|   |- css
|      |- index.css
|   |- img
|      |- logo.png
|   |- index.html
|   |- js
|      |- index.js
```

index.html

```
<html>
    <head>
        <meta http-equiv="Content-Security-Policy" content="default-src 'self'
        data: gap: https://ssl.gstatic.com 'unsafe-eval'; style-src 'self'
        'unsafe-inline'; media-src *">
        <meta name="format-detection" content="telephone=no">
        <meta name="msapplication-tap-highlight" content="no">
        <meta name="viewport" content="user-scalable=no, initial-scale=1,
        maximum-scale=1, minimum-scale=1, width=device-width">
        <link rel="stylesheet" type="text/css" href="css/index.css">
        <title>Hello World</title>
    </head>
    <body>
        <div class="app">
            <h1>Apache Cordova</h1>
            <div id="deviceready" class="blink">
                <p class="event listening">Connecting to Device</p>
                <p class="event received">Device is Ready</p>
            </div>
        </div>
        <script type="text/javascript" src="cordova.js"></script>
        <script type="text/javascript" src="js/index.js"></script>
    </body>
</html>
```

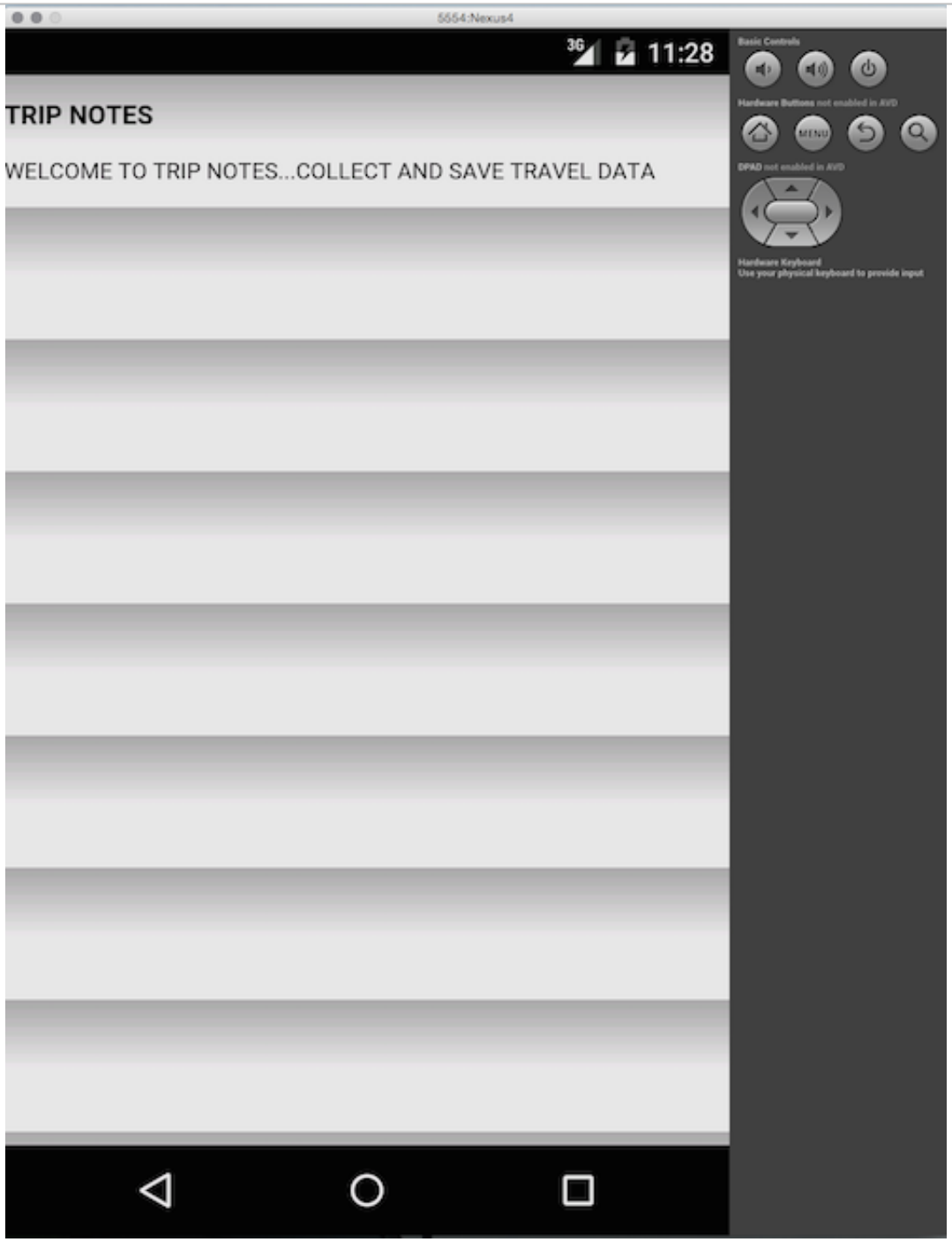
Cordova App - basic app - part 2

index.html

```
<body>
    <div>
      <h3>Trip Notes</h3>
      <p>
        welcome to trip notes...collect and save travel data
      </p>
    </div>
    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript" src="js/index.js"></script>
</body>
```

- lack of styling will be an issue...

Image - Cordova App - Basic v0.01



Trip Notes - v0.01

add Cordova specifics

- Cordova container for the application exposes native APIs to web application running in WebView
- most APIs not available until applicable plugin added to the project
- container also needs to perform some preparation before the APIs can be used
- Cordova informs us when the container, and associated APIs, are ready for use
- fires a specific event, called the `deviceready` event
- application logic requiring use of Cordova APIs
  - *should be executed after receipt of `deviceready` notification*

Cordova App - basic app - part 4

add some jQuery

- add to foot of body

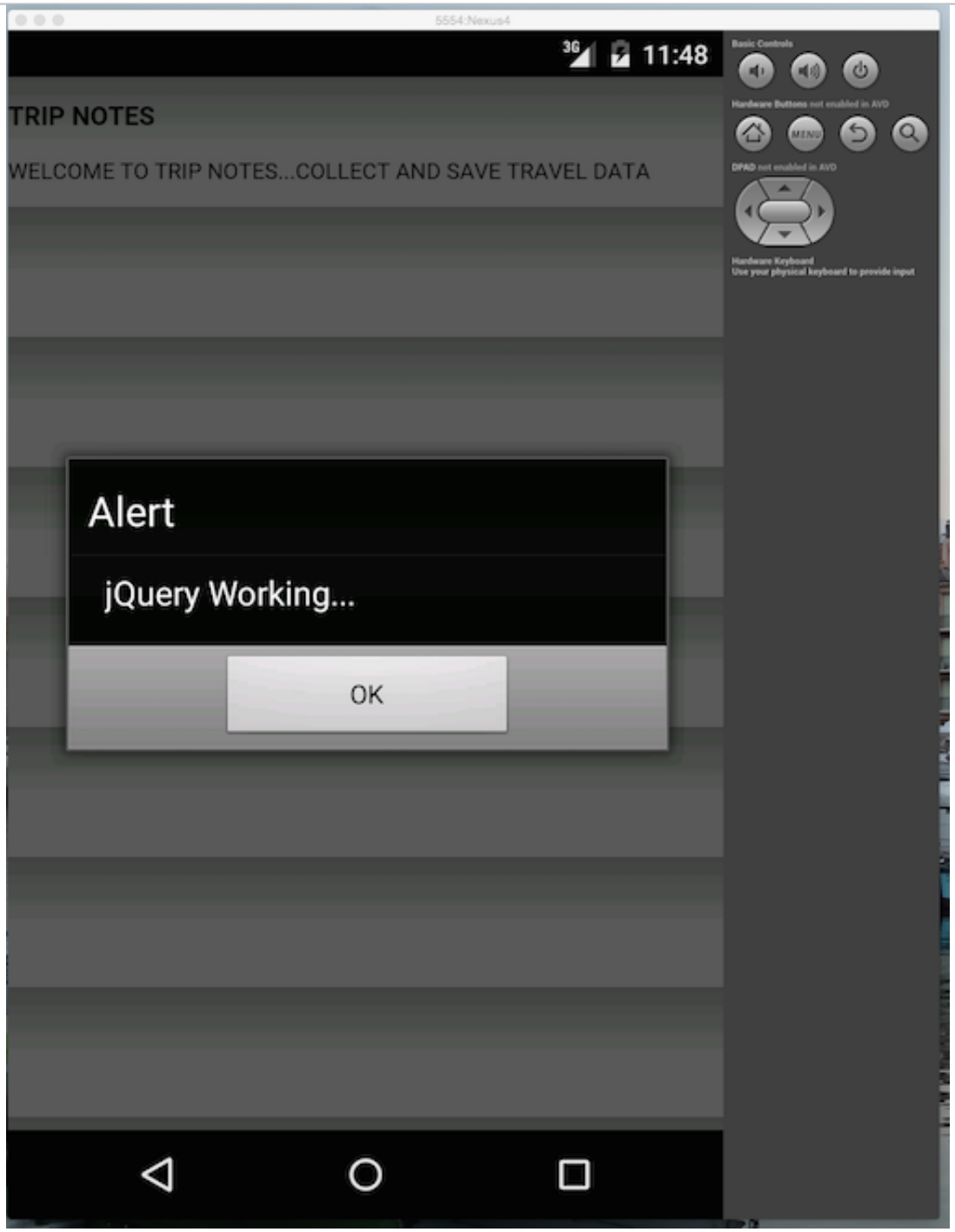
```
<script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>
```

- add test to `trip.js` file

```
function tripNotes() {
  alert("jQuery Working...");
}

$(document).ready(tripNotes);
```

Image - Cordova App - Basic v0.02



Trip Notes - v0.02

Cordova App - basic app - part 5

add some jQuery Mobile

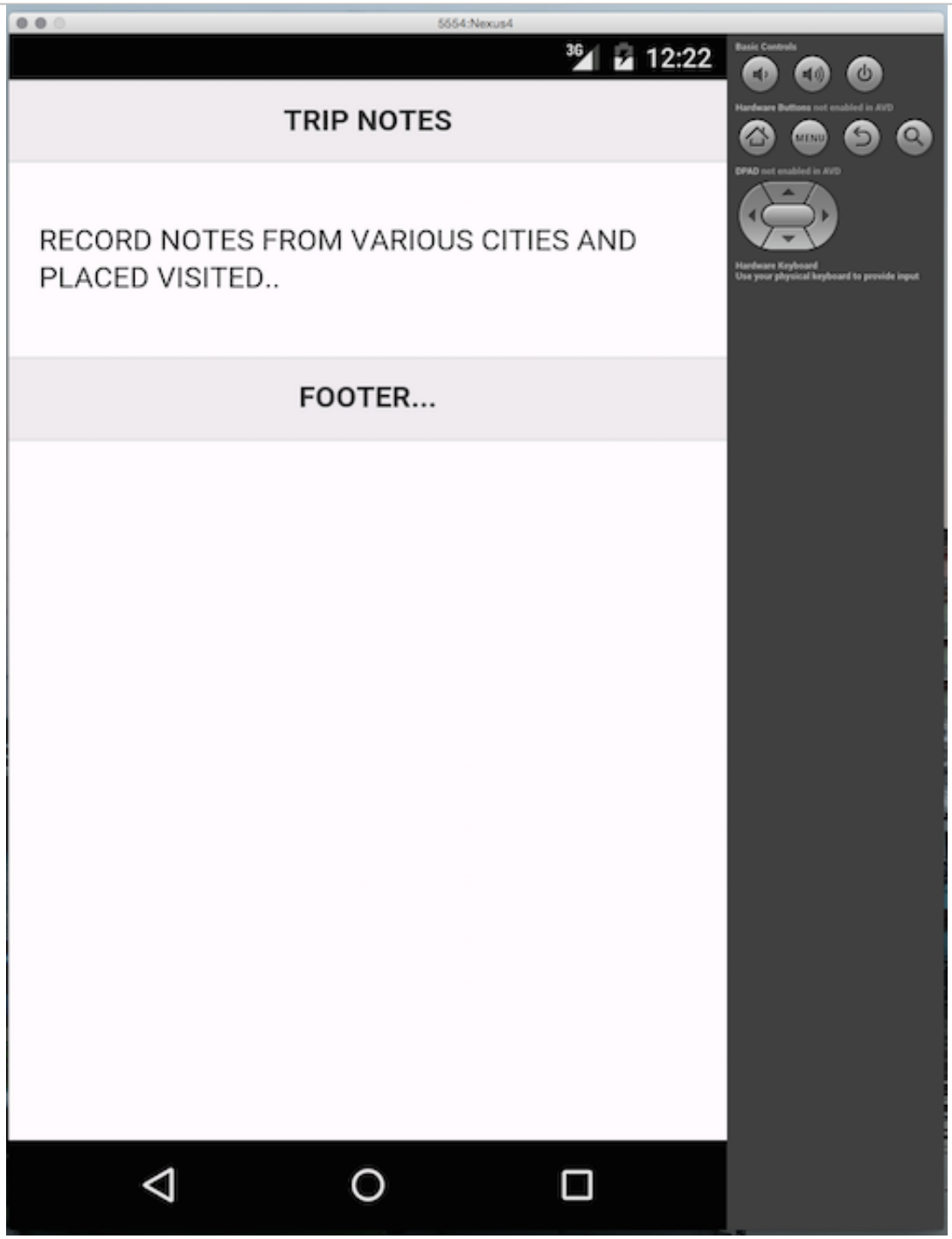
- update head with local jQuery Mobile CSS

```
<head>
...
<link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.4.5.min.css" />
</head>
```

- update body for basic app

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h3>trip notes</h3>
    </div><!-- /header -->
    <div role="main" class="ui-content">
      <p>record notes from various cities and placed visited..</p>
    </div><!-- /content -->
    <div data-role="footer">
      <h5>footer...</h5>
    </div><!-- /footer -->
  </div><!-- /page -->
  <script type="text/javascript" src="cordova.js"></script>
  <script type="text/javascript" src="js/index.js"></script>
  <script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>
  <script type="text/javascript" src="js/jquery.mobile-1.4.5.min.js"></script>
  <script type="text/javascript" src="js/trip.js"></script>
</body>
```

Image - Cordova App - Basic v0.03



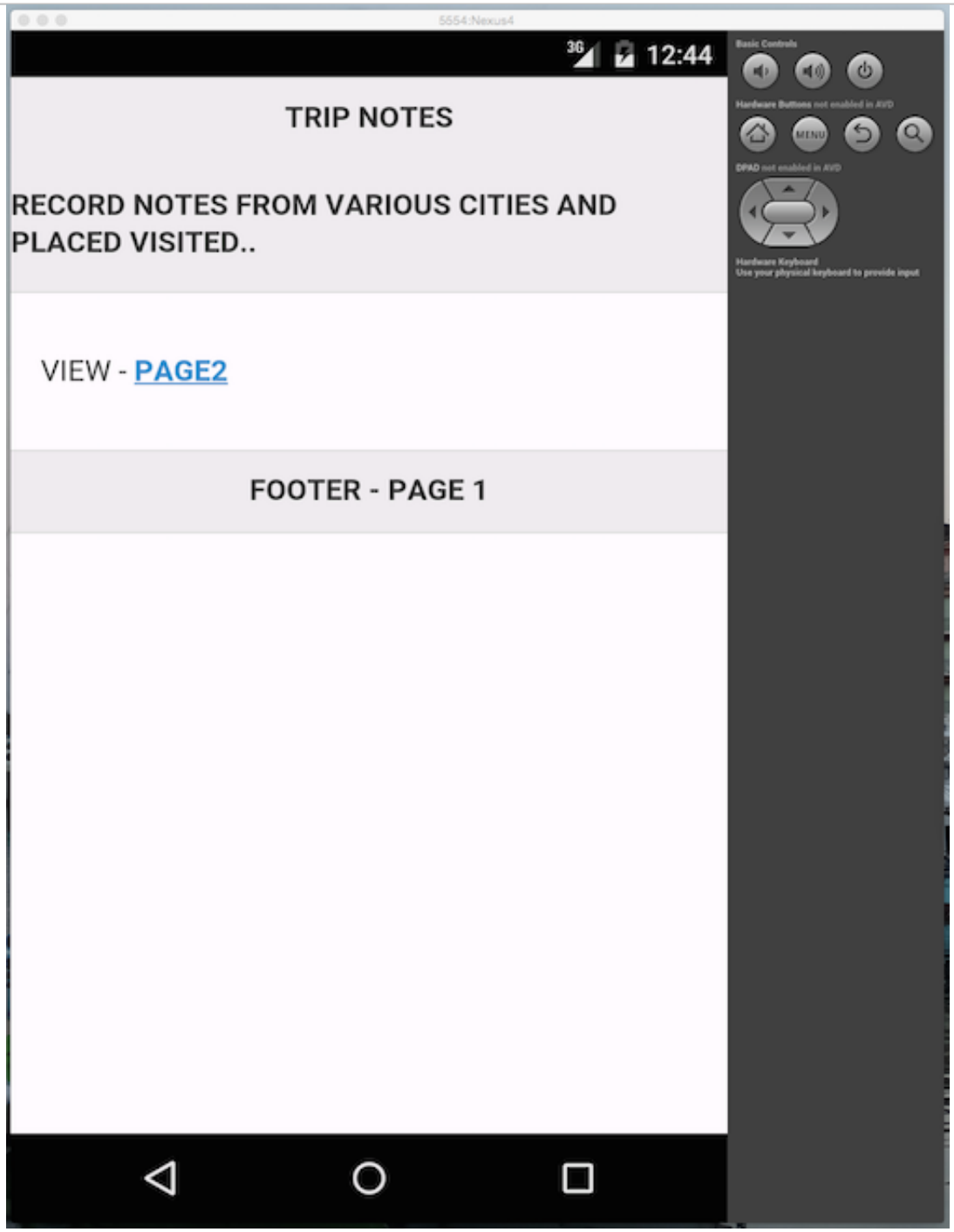
Trip Notes - v0.03

jQuery Mobile - test transitions

- update `index.html` to add page containers, transitions...

```
<!-- page1 -->
<div data-role="page" id="page1">
  <div data-role="header">
    <h3>trip notes</h3>
    <p>record notes from various cities and placed visited..</p>
  </div><!-- /header -->
  <div role="main" class="ui-content">
    <p>View - <a href="#page2" data-transition="slidedown">page2</a></p>
  </div><!-- /content -->
  <div data-role="footer">
    <h5>footer - page 1</h5>
  </div><!-- /footer -->
</div><!-- /page1 -->
<!-- page2 -->
<div data-role="page" data-dialog="true" id="page2">
  <div data-role="header">
    <h3>page 2</h3>
  </div><!-- /header -->
  <div role="main" class="ui-content">
    <p><a href="#page1" data-rel="back">Cancel</a></p>
  </div><!-- /content -->
  <div data-role="footer">
    <h5>footer - page 2</h5>
  </div><!-- /footer -->
</div><!-- /page2 -->
```

Image - Cordova App - Basic v0.04



Trip Notes - v0.04

Image - Cordova App - Basic v0.05

Trip Notes - v0.05

jQuery Mobile - navigation - part 1

intro

- navigation within our apps
- navigation is thankfully **asynchronous**
- jQuery Mobile navigation loads pages into DOM using Ajax
- modify the page's content, then re-render for display to the user
- includes a set of aesthetically pleasing, and useful, animations
  - *help inform the user of changes in state, and appropriate updates in the content*
- navigation system effectively hijacks a link within a page's content container
  - *routes it through an Ajax request*
- benefit for developers is simple approach to asynchronous navigation
- still able to support standard concepts such as **anchors** and **back** button
  - *without breaking coherence and logic of the application*

jQuery Mobile - navigation - part 2

intro - continued

- jQuery Mobile is able to load and view groups of disparate content
  - *in page content containers within our initial home document*
- support for core JavaScript event handling
  - *URL fragment identifiers with `hashchange` and `popstate`*
- allows the application to persist navigation history, at least temporarily
  - *a record of user navigation and paths through the content*
- tap into this internal history of the application
  - *hijack certain patterns to help us better inform the user*
  - *add details about state changes, different paths, content, and so on...*

jQuery Mobile - navigation - part 3

example navigation

- example of using the jQuery Mobile standard method, `$.mobile.navigate`
- used as a convenient way to track history and navigation events
- set our record information for the link
  - *any useful information for the link or affected change in state*
- log the available direction for navigation
- url for the nav state, and any available hash
  - *in our example the simple hash, `#nav1`*
- Demo - nav

jQuery Mobile - using widgets - part 1

- add standard HTML elements within our content containers
  - *<p>, <h1>, <h2>..., li, <section>...*
- jQuery Mobile includes a wide-range of widgets
- simply add the widgets to our applications
- touch friendly widgets
  - *eg: collapsible elements, forms, responsive tables, dialogs...*
- `pageContainer` widget for a content container

jQuery Mobile - using widgets - part 2

listviews

- style, render, manipulate standard data output and collections
- render lists as interactive, animated views
- lists are coded with a now familiar `data-role` attribute,

```
data-role="listview"
```

- we can also set links on our lists
  - *rendered with styling and link icons*
  - *add new page, add extra styles...*
- demo1 - jQuery Mobile listview 1
- demo2 - jQuery Mobile listview 2

jQuery Mobile - using widgets - part 3

listviews - example

```
<!-- listview example -->
<div>
  <ul data-role="listview">
    <li>Cannes</li>
    <li>Marseille</li>
    <li><a href="#page3" data-transition="slide">Monaco</a></li>
    <li>Nice</li>
  </ul>
</div>
```

- simple listview with slide transition

```
<!-- page3 -->
<div data-role="page" id="page3">
  <div data-role="header">
    <h3>page 3</h3>
  </div><!-- /header -->
  <div role="main" class="ui-content">
    <p><a data-rel="back" class="ui-btn">Return</a></p>
  <section class="image-view">
    <img src="media/images/monaco1.jpg">
  <section>
  </div><!-- /content -->
  <div data-role="footer">
    <h5>footer - page 3</h5>
  </div><!-- /footer -->
</div><!-- /page3 -->
```

- new page for Monaco image
- demo3 - jQuery Mobile listview 3

jQuery Mobile - using widgets - part 4

listviews

- use listviews to add filtering and live search options to our lists
- set a simple client-side filter
  - *add an attribute for* `data-filter`
  - *then set the value to* `true`

```
data-filter="true"
```

- also set some default, helpful text for the input field
  - *prompts user to interact, and use this feature correctly*

```
data-filter-placeholder="Search Cities"
```

- tidy up the presentation of our list, add an inset using the attribute

```
data-inset="true"
```

- demo4 - jQuery Mobile listview 4

jQuery Mobile - using widgets - part 5

listviews - adding some formatted content

- fun aspects of working with a framework such as jQuery Mobile
  - *simple way we can organise, format our data presentations and views*
- grouped dataset can still be presented using lists
  - *add informative headings*
  - *links to different categories within this dataset*
  - *add simple styling to help differentiate list components*
- structure the list as normal, with sub-headings, paragraphs, and so on
  - *jQuery Mobile option for setting list content as an aside element*

```
<p class="ui-li-aside">1 image</p>
```

- many similar tweaks, additions for listviews...
  - *visit jQuery Mobile API for further details*

jQuery Mobile - using widgets - part 6

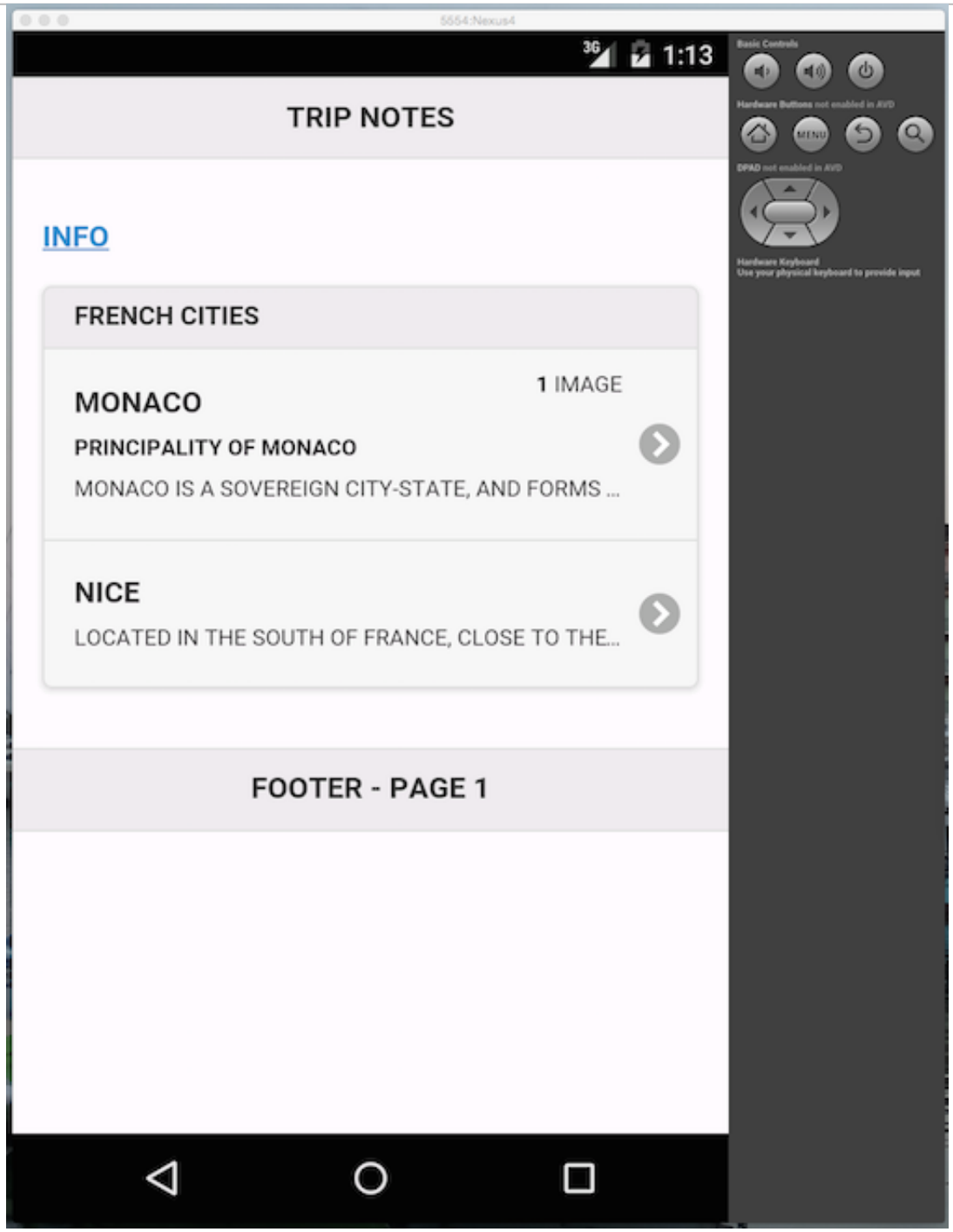
listviews - example

```
<ul data-role="listview" data-inset="true">
  <li data-role="list-divider" role="heading">French Cities</li>
  <li>
    <a href="#page3" data-transition="slide">
      <h3>Monaco</h3>
      <p><strong>Principality of Monaco</strong></p>
      <p>Monaco is a sovereign city-state, and forms part of the French Riviera...</
      <p class="ui-li-aside"><strong>1</strong> image</p>
    </a>
  </li>
  <li>
    <a href="#">
      <h3>Nice</h3>
      <p>Located in the south of France, close to the border with Italy...</p>
    </a>
  </li>
</ul>
```

- demo5 - jQuery Mobile listview 5

Cordova App - basic - part 7

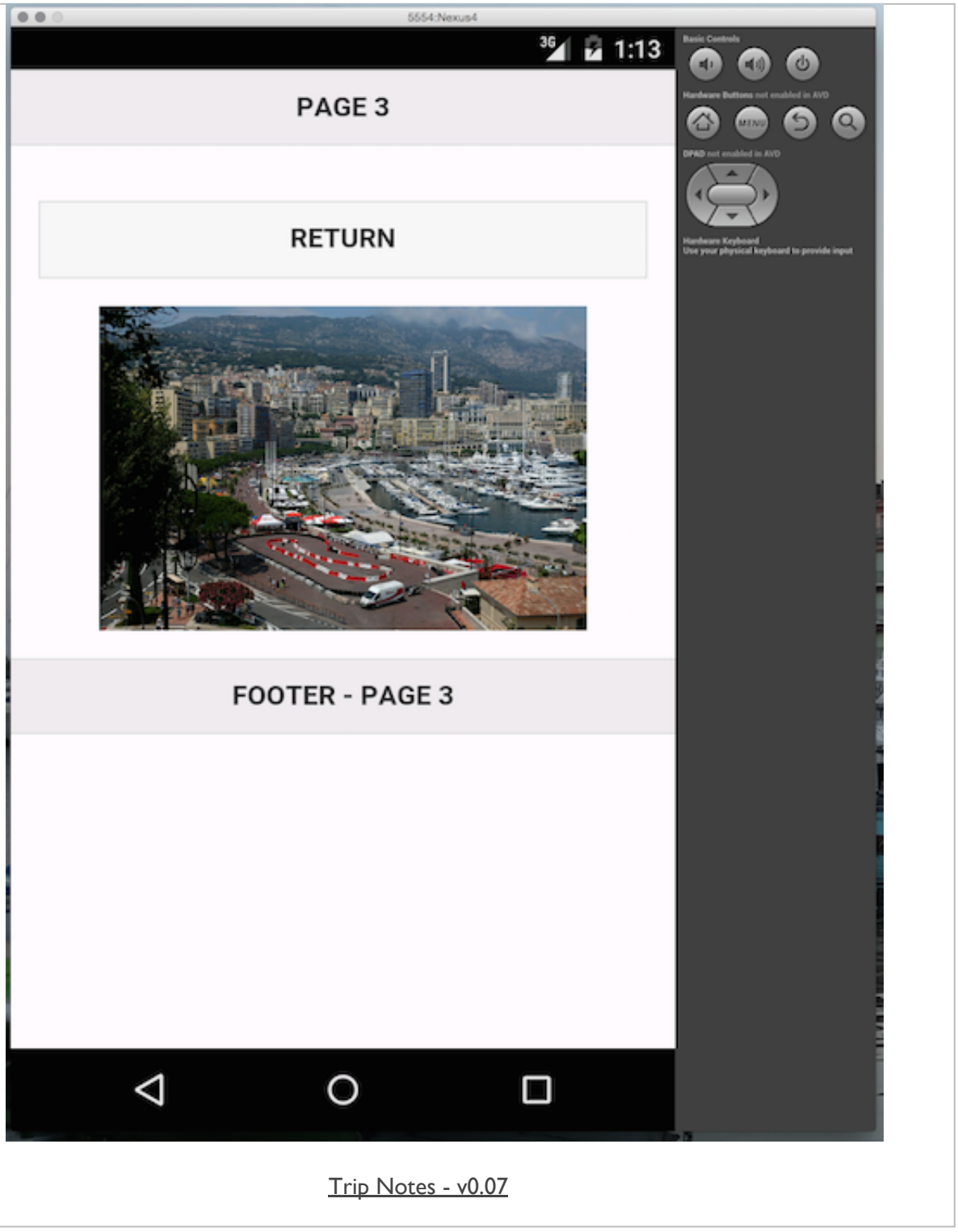
jQuery Mobile - add some organisation



Trip Notes - v0.06

Cordova App - basic - part 8

jQuery Mobile - add some organisation



Trip Notes - v0.07

Design and interface - intro

- consider some of the concepts, challenges, and options for interface design on the client-side
- important to remember the very nature of these applications
- these apps are inherently
  - *highly interactive*
  - *display content from myriad sources, including databases and streaming APIs*
  - *communicate with other systems*
  - *now more dynamic than ever*
- often designed and implemented with more than one activity in mind
- often represent actions such as finding results and records
  - *whilst also managing that data*
- access to and awarenesss of real-time data and streams
  - *strongly influencing our design and development*
  - *from news to banking*

Design and interface - goals

- an issue with app design is often focusing on both functionality, and complementary aesthetics
- look at client-side design trends in general
  - *ubiquity of digital applications has led to a reduction for many early design conventions*
  - *rare to see a site still use browser defaults for links*
  - *actual design or aesthetic choice*
  - *a lack of design for design's sake, to leave them set to a blue with underline*
- breadth and diversity of devices and network connected applications
  - *also see a dizzying number of evolving patterns and standards*
- consider Apple's or Google's design guidelines, then compare to Microsoft's
- no single pattern for use, no unified visual language outside of prescribed ecosystems
- want applications we design and use to be more than simply utilitarian

Design and interface - design as a guide

- interfaces simply allow us to mediate communication options and associated interaction
  - *through screens and available networks*
- definite need for a clear visual language
  - *contains signs and symbols to help inform our users*
  - *provide complementary direction and feedback*
- not as simple as just presenting the data as various forms of information

> *...primary technique to achieve improved visual communication is to use clear, distinct, consistent visible language...*
>
> *Marcus, Aaron. Graphic Design for Electronic Documents and User Interfaces*

- consider detailed, complex visual interfaces
  - *can observe the many messages conveyed on a single screen*
- challenge for design is to create some semblance of direction, order, and pattern
  - *help users simply make sense of what they see*

Design and interface - communication

- can be considered as involving
  - *a sender*
  - *a message*
  - *a conveying signal or carrier for the message*
  - *a receiver or viewer who needs to interpret the message*
- readily observe as designers and developers
  - ***we are not able to control the entire process***
- interface design
  - *the very act of selecting elements with user expectations in mind*
  - *then the combination of these elements*
  - *with appropriate and useful visual signals that users actually understand*
  - *makes it more likely a target audience of users will successfully understand and interpret our message*
- need interfaces to help us successfully manage increasingly complex nature of data

Design and interface - direction and principles

- a basic framework, a set of underlying principles we can follow or use
  - *a basic template for how we think and act as designers*
- start designing our applications with a more informed decision-making process
  - *helps us bridge form and function*
  - *provide a sense of the beautiful with the useful*
  - *such considerations are not mutually exclusive*
- underlying principles we can consider, and apply, to our designs
  - *inherently help inform good practice and design choices for our development work*
- principles will focus upon
  - ***consistency***, ***hierarchy***, *and basic design* ***personality***
- consider these underlying principles in a similar vein to syntax or language
- **Consistency** and **hierarchy** are often seen as analagous to a language's grammar
  - *a user learns whilst using an application.*
- visual **personality** of our design
  - *visual characteristics, notable traits in effect, of our design become the words we use to convey our message*
- such principles can hold true even as technology continues to evolve
- design aesthetics and principles can remain as a footprint of our work

Design and interface - tools of the trade

- consider visual tools of our trade
  - *the nuts and bolts of visual design*
- **tools** that help us layout and construct our interfaces for users
- need to define and outline the various visual tools of application design
  - *affordances*
  - *colour*
  - *controls*
  - *imagery*
  - *layout*
  - *type*

Design and interface - common mistakes

- consider some of the common mistakes
  - *affect our ability to design*
  - *implement consistency within our interfaces*

- consider interfaces that achieve consistency
  - *colours appropriate for the criteria or usage environment*
  - *consistent use of colours*
  - *consistent standards for typography*
  - *consistent implementation and styling of controls*
  - *elements correctly organised and aligned*
  - *elements placed in a logical position for users*
  - *ie: where users expect to find them*
  - *fonts used appropriate to a given situation, event...*
  - *grouping of similar, contiguous elements*

Design and interface - consistency, consistency...

- we need to establish rules for placement and usage of interface elements
- need to consistently adhere to these prescribed rules
- mix and match visual interface characteristics without confusing, and annoying, our users
- designer's visual language, like natural language
  - *requires a set of rules to be applied consistently*
  - *these rules can then be recognised and interpreted*
- consistency in design is rarely exciting or necessarily interesting
- it will help our users gain an innate sense of familiarity with an application
- hopefully helps drive further adoption and usage
- design consistency is simply about giving users what they can understand
  - *in essence, rely on throughout an application*

Image - design consistency example

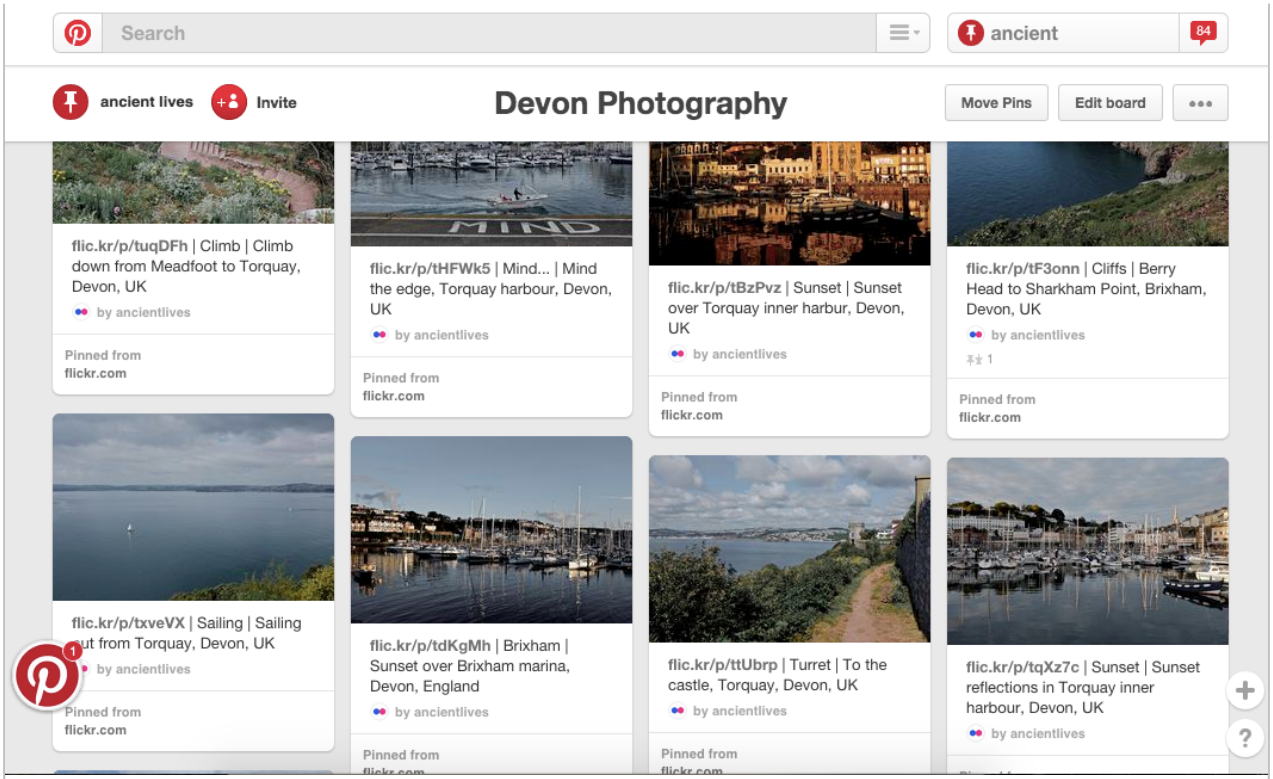
A lack of consistency - Arngren.net

Source - Arngren.net

Design and interface - considering consistency

- start to design our interfaces for applications and web sites
- then update them in response to feedback or feature changes
  - *smallest changes can cause a ripple effect throughout our application*
- applications may change and evolve, implementing new or updated technologies, options...
- still need to establish consistency in usage
  - *eg: Pinterest interface*
  - *uses an interface mechanism of continuous scrolling to display a rich variety of images*
  - *now an accepted option for an interface pattern*
- continuous scroll pattern is attempting to solve a given problem
  - *user needs to view a subset of data that is not easily displayed on a single page*
  - *application's content presented to users as focused subset*
  - *larger, seemingly endless dataset to focused view*
  - *user needs to be aware of the ongoing content*
  - *without excessive effort or hindrance to the usage experience*

Image - Pinterest continuous scroll



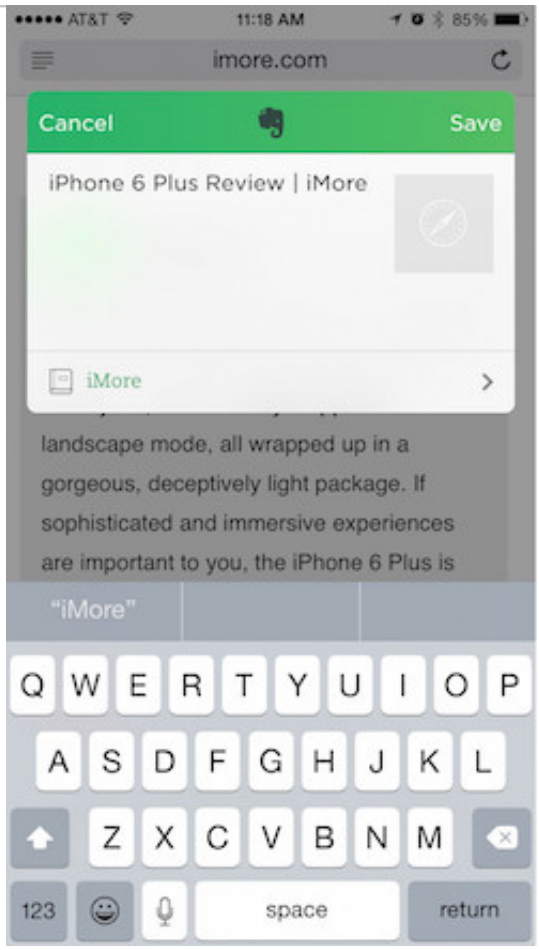
Pinterest - Continuous Scroll

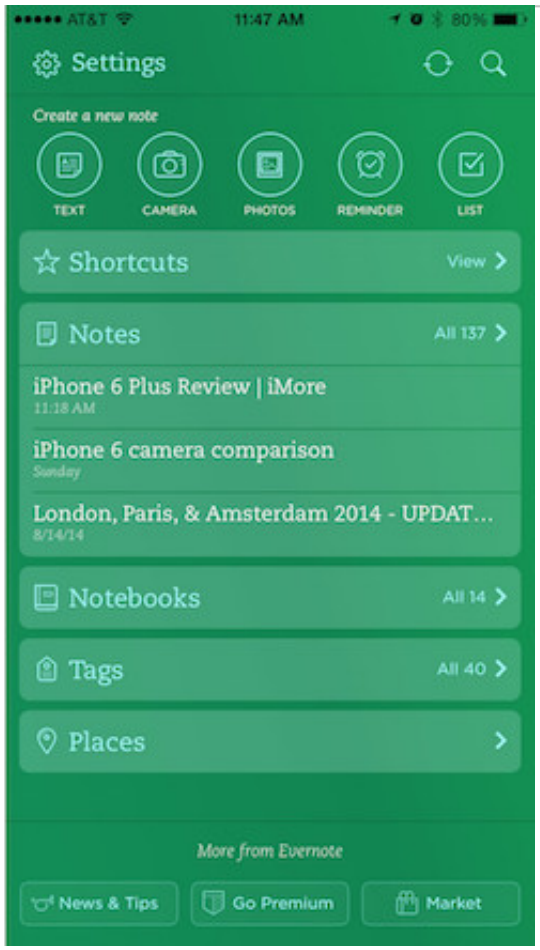
Source - Pinterest

Design and interface - establishing consistency

- help our users by starting with familiar elements and designs
  - *elements and designs people are familiar with from other examples and applications*
- users' expectations can simply be influenced by what they see onscreen
  - *naturally what they've seen in the past*
- a good reason that to review and consider many different types and examples of websites
- forms can be a good example of this type of conditioning and expectation in users
- a user sees a form for payment or credit card information
  - *they have normally seen and used other examples*
  - *examples will often follow a similar pattern*
- we can modify slightly to match specific requirements
  - *such as text, specific event or purchase details...*
- a user will normally look for familiar interface elements
  - *such as a **submit** button, input field...*
- as users, we become conditioned to use patterns on a regular basis
- consistency relies on an inherent awareness of user expectations

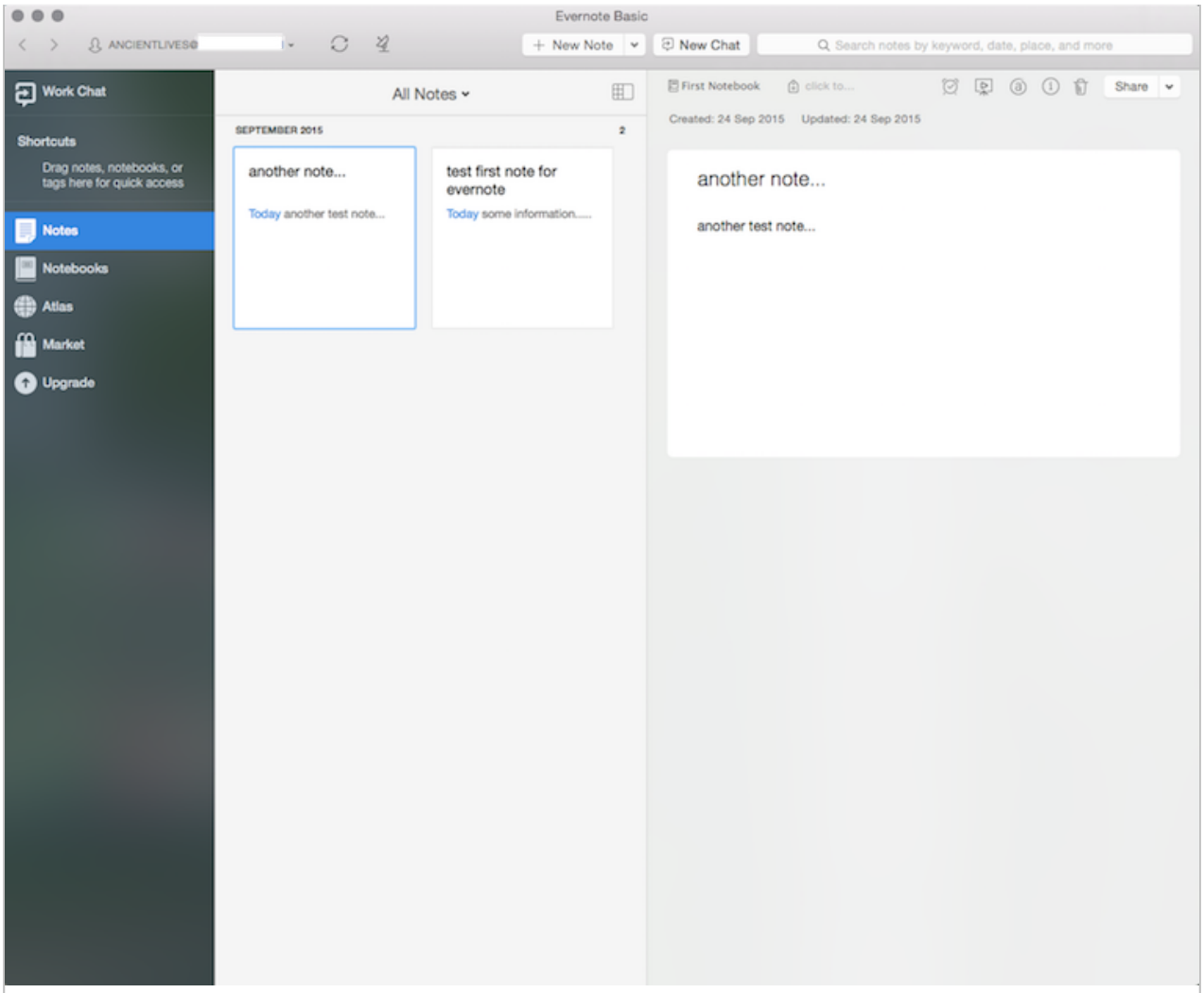
Design and interface - examples of consistency to consider...





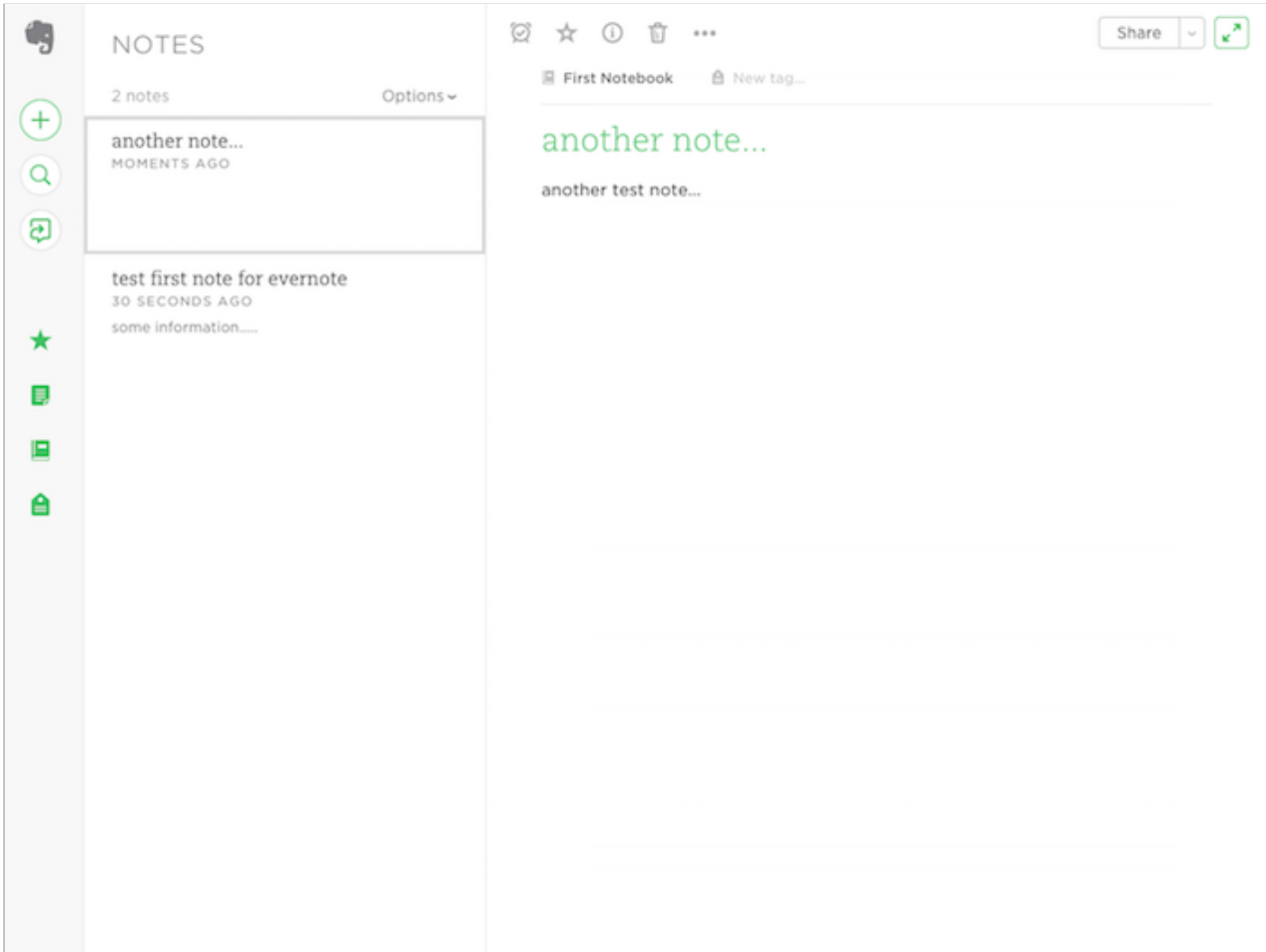
Evernote on iOS

Design and interface - examples of consistency to consider...



Evernote on OS X

Design and interface - examples of consistency to consider...



Evernote website

Demos

Cordova

- Trip Notes - v0.0.1

jQuery Mobile

- demo1 - jQuery Mobile listview 1
- demo2 - jQuery Mobile listview 2
- demo3 - jQuery Mobile listview 3
- demo4 - jQuery Mobile listview 4
- demo5 - jQuery Mobile listview 5

References

- Aaron, Marcus. *Graphic Design for Electronic Documents and User Interfaces*. ACM Press. 1992.
- jQuery Mobile - API