

# Assignment 2

Specification 2014

# Assignment 2

- DUE: Sunday May 18th 2014
- Weight: 60%
- Goal:
  - Demonstrate ability to develop a scalable on-line service, delivered via a cross-platform app
- Deliverables:
  - Source code (in specified folder structure)
  - Design documentation
  - Live deployment

# Assignment: Concept

- App/Service concept must be one of:
  - **Your own idea/ itch/ need** - preferred!
  - **Social** - e.g. sharing photos, links, etc.
  - **Local** - e.g. restaurants near me, hail a taxi, etc.
  - **Mobile** - e.g. bus times, recipes, quick info, etc.
  - **Utility** - e.g. weather, unit converters, etc.
  - **Service** - wrap your favourite service API,
    - e.g. twitter, eventbrite, basecamp, etc.

# Assignment: Design

- **Wireframes**
  - demonstrate user interaction flow
  - cover all main functions
- **Service API**
  - follows REST principles
  - Well-documented with examples
- **Use-Cases**
  - Actor; Goal; Scenario (steps 1,2,3,...)
- **Diagrams**
  - Explain relationships, system architecture
    - Structure - objects, "classes", components, network,
    - etc. Interaction - events, sequences, states, data flows, etc.

# Assignment: Client

- **Mobile Web**
  - preferably crossplatform
  - should be usable offline and bookmarkable
- **Hybrid**
  - preferably cross-platform
  - uses capabilities not available with Mobile Web
- Goal: Same code base serves mobile web and hybrid
  - code base can be dynamic - e.g. deliver different code depending on user-agent
- Cross-platform:
  - layout is responsive to platform - adjusts size etc.
    - e.g. tab bar at top for Android (if you have a tab bar!)
  - Physical devices:
    - *Please indicate physical devices that you had access to*
    - iOS: desktop safari can be considered a reasonable proxy if no access to Mac/iPhone
    - Android: must run in simulator at least (Eclipse is available on Win/Mac/Linux)

# Assignment: Server

- **Hosting**

- Two options:
  - hosting provider: heroku, feedhenry, nodejitsu, etc. - must be Node.js
  - Amazon cloud - higher weighting (and thus less effort needed on additional goals)

- **Service Integration**

- Use third party online services to enhance app
- e.g. urbanairship.com (push notify), parse.com (data+users management) or
- even non-mobile services like evernote, postmark, etc.
- if not using amazon, you'll probably want more than one

- **Configuration Management**

- Managing and reconfiguring your live app
  - without requiring an app store update
- Be careful - code updates are not allowed, only configuration

# Assignment: Functionality

- **Concept**

- "sufficiently" complex - at least twice the code volume of Assignment 1
  - It's OK to just replicate something like instagram
    - so long as it's your code!
    - design deliverables will need to reach a higher standard

- **Social Media Interaction**

- Login via social media sites is recommended
- Use of social media APIs in some form is expected

- **Device Hardware**

- Make some use of device hardware over and above Mobile Web apps, e.g. camera  
If not required by app concept, you will need to compensate with more depth in social media and third party integrations

- **Analytics**

- Capture some form of analytics from your app
- Give some insight into user behaviour

# Assignment: Folder Structure

- **site**
  - **public**
    - *HTML files*
    - **js**
      - *JavaScript files*
    - **css**
      - *CSS files*
    - **img**
      - *Image files*
  - **node**
    - **lib**
      - *Server-side JavaScript files*
    - **node\_modules** (as create by npm)
  - **conf**
    - *Configuration files*



# Assignment: Tips

- **Verify all services you intend to use first!**
  - create accounts, read API documentation, code up some quick tests
- Use version control - preferably git so that you can use github.com
- Complete features in desktop Safari first
  - but don't leave device/simulator/emulator testing till the end
- **Use a build process**
  - You'll need some way to share files between web and hybrid folders - even if it's just a simple batch file
- Use ad hoc console testing
  - especially useful with weinre on devices
- Use unit testing - not required but keeps you sane
- Cross-platform code is hard
  - when all else fails deliberately break features to verify assumptions

# Assignment: Resources

- Books and Websites given in Lectures
- Google Groups and IRC:
  - Node.js
  - PhoneGap
- Stackoverflow.net
  - Do not hesitate to use
- Primary Sources:
  - [aws.amazon.com](http://aws.amazon.com)
  - [developer.apple.com](http://developer.apple.com)
  - [developer.android.com](http://developer.android.com)

# Grading Spectrum

(indicative only)

<i>Range</i>	<i>Design: 25%</i>	<i>Client: 25%</i>	<i>Server: 25%</i>	<i>Functionality: 25%</i>
Baseline	Wireframes	Mobile Web	Application Environment Hosted	Concept Depth
Good	+ API Documentation	Cross-Platform	OR: Amazon Hosted	+ Social Media Integration
Excellent	+ Use-Case Analysis	Hybrid Single Platform	+ Service Provider Integration	+ Device Hardware Usage
Outstanding	+ Architectural Diagrams	Hybrid Cross-Platform	+ Configuration Management	+ Analytics