EECS 394
SOFTWARE DEVELOPMENT

Chris Riesbeck

Developing Mobile/Web Apps
Next Tasks

- Starting project reports
- Building your mobile web application
Due 9am every Monday, Thursday

Special report due noon, Wednesday, April 4

Short but useful

Everyone writes, someone collates and sends

Collator task rotates

Subject: EECS 394 Team X Progress Report

Always CC the entire team
First report is different. Has two main components:

- **Product vision**
  - Product box (text is fine)
  - Core user scenario(s) for MVP
  - If you don't already have these, you're probably doing a lot of wasted work.

- **Technical platform**
  - PHP, R on R, Django, ....
  - Source control
Everyone writes at most a line or so for each the following:

- Done: what did you finish since the last report?
- Doing: what are you working on now?
- Obstacles: anything in your way?
- Other: other substantial project-related tasks (meetings, computer setup, investigating options, writing report)
- Include partner (pairing or swarming) information
Done and Doing always in terms of what users can or will be able to do

- Bad: "wrote code to calculate remaining budget"
- Good: "users can now see remaining budget"

Done means tested and deployed.

- No partial credit. No 90% done.
Doing same tasks in several reports.
Doing a task reported done before.
All work under Other.

What problems might these be warning signs of?
Done: Implemented dataTables jquery plugin to handle displaying, sorting and filtering tabular alumni data

Doing: teleconference with managers at noon.

Doing: Started work on css for the whole project page.

Doing: 24 hour coding marathon tonight.

Doing: "As a current MPD2 student I can edit my profile name, location, profile picture, company, title, personal program reflection, personal quote, and email address."
Which are the better reports?

Doing: Integrate the slideshow element into the design theme for improved Visitor experience.

Doing: Communicate with client on font + font colors and placements / about any design revisions

Done: "public users can send email to students"

Doing: Hook up forms to fields in the database

Doing: Project gallery will load real content from database

Wednesday, April 4, 2012
MOBILE CHOICES
Native Apps

Native app
- access to all phone features
- best look and feel and performance

BUT non-portable development platforms:
- iPhone: Objective-C
- Android: Java
MOBILE APP REQUIREMENTS

- iPhone requirements:
  - Mac with Snow Leopard (Lion?)
  - XCode IDE and libraries
  - iPhone SDK, developer license
  - iPhone to deploy

- Android requirements:
  - Windows / Linux / Macs
  - Eclipse IDE recommended
  - Android SDK
  - phone recommended but emulator works
PhoneGap / Cordova

- Portable framework for iPhone, Android, Blackberry, ..
- Runs as a native app, but coded HTML5 + CSS + Javascript
- Same requirements as native development
  - iPhone: XCode, iPhone SDK, iPhone...
  - Android: Eclipse, Java, Android SDK, ...
RESOURCES: PHONEGAP

- http://www.phonegap.com/
- http://incubator.apache.org/cordova/

Videos
- Early, non-slick 3 minute spiel on PhoneGap
- 9-minute demo converting HTML to iPhoneApp

Discussion groups
- http://groups.google.com/group/phonegap
- http://phonegapforum.com/

(books) http://www.amazon.com/s/ref=nb_sb_noss_1?url=search-alias%3Daps&field-keywords=phonegap
Web Pages

- Web page
  - multi-platform, easily tested
  - only needs HTML/CSS/JavaScript skills
  - BUT limited access to phone features
  - feels like a web page
MOBILE WEB PAGES

- HTML5 + CSS + Javascript
  - support touch gestures
  - access to some phone features
  - can adapt to multiple screen sizes
  - provides local data store – critical for offline use
  - animations can be hardware optimized
Mobile Web Page Resources

**MOBILE WEB PAGE FRAMEWORKS**

- (book ad) [http://www.sitepoint.com/books/mobile1/](http://www.sitepoint.com/books/mobile1/)
WEB APP TIPS

- Develop and test in Webkit browsers
  - Chrome or Safari
  - `<!DOCTYPE HTML>`
- Validate constantly!
  - HTML: http://validator.w3.org/
  - CSS: http://jigsaw.w3.org/css-validator/
  - Javascript: http://www.jslint.com/
HAS THIS EVER HAPPENED TO YOU?

- You select an item from a menu.
- The browser shows data about that item.
Has this ever happened to you?

You select an item from a menu.
The browser shows data about that item.

You click the back button to see something else.

A scary dialog box asks if you want to resubmit data to the server.
Has this ever happened to you?

You select an item from a menu.

The browser shows data about that item.

You click the back button to see something else.

A scary dialog box asks if you want to resubmit data to the server.

I'm talking to you, CAESAR.
Has this ever happened to you?

- You select an item from a menu.
- The browser shows data about that item.
- You bookmark that page for future reference.
- Later, you click the bookmark.
- The browser takes you to the main page instead.
- You click the back button to see something else.
- A scary dialog box asks if you want to resubmit data to the server.

I'm talking to you, CAESAR.
Has this ever happened to you?

You select an item from a menu.
The browser shows data about that item.

You click the back button to see something else.
A scary dialog box asks if you want to resubmit data to the server.

You bookmark that page for future reference.
Later, you click the bookmark.
The browser takes you to the main page instead.

I’m talking to you, CAESAR.
I’m talking to you, Blackboard.
How the Web Works

(slightly simplified)
How the web works

(slightly simplified)

Browser

Clicking a link: \text{GET} + \text{URL}

Submitting a form: \text{POST} + \text{form data}

Server
How the Web Works

(slightly simplified)

Clicking a link: GET + URL

Submitting a form: POST + form data

HTML + CSS + media

Wednesday, April 4, 2012
GET and POST are very different actions for the browser.

Clicking a link: GET + URL

Submitting a form: POST + form data

HTML + CSS + media

GET and POST are often handled by the same code on the server.

(slightly simplified)
HTTP METHOD TYPES

GET, HEAD

POST
HTTP METHOD TYPES

GET, HEAD
Safe
Idempotent
only retrieves data
repeated calls get the same results
POST
HTTP METHOD TYPES

GET, HEAD

Safe

only retrieves data

Idempotent

repeated calls get the same results

POST

Neither

Ergo, browsers ask before repeating
The web scaled because sites are mostly repositories of self-describing resources, not applications.

Roy Fielding "Architectural Styles and the Design of Network-based Software Architectures" (PhD, UCI, 2000)
RESTFUL WEB APPS

(REpresentational State Transfer)

define as much of your system as possible in terms of resources (lists, item details, carts, user profiles, ...)

provide an initial unchanging home URL with links to other resources

make all resources available via some chain of links starting from the home page

use GET to retrieve resources

use POST (and/or PUT and DELETE) to update resources
Common Mistake #1

**Bug**

A link (GET) that updates a resource.

Example: the PHP tutorial's demo blog has a delete link next to each blog entry

A web crawler, e.g., Google, a site map generator, a broken link finder, even a browser pre-fetch loop, will delete every blog entry!

**Fix**

Use a form with POST for all actions that modify server state.
COMMON MISTAKE #2

Bug

A form (POST) that just gets a resource.

Example: a search field

Search results can't be bookmarked.

Going back to the results page triggers "do you want to resend data?"

Fix

Use form with GET method

Use POST-REDIRECT-GET pattern

Wednesday, April 4, 2012
http://blog.andreloker.de/post/2008/06/Post-Redirect-Get.aspx
POST-REDIRECT-GET

Client  

POST + form data  

Server

http://blog.andreloker.de/post/2008/06/Post-Redirect-Get.aspx
POST-REDIRECT-GET

**Client**

**Server**

POST + *form data*

Redirect to *URL*

GET + *URL*

---

http://blog.andrelroker.de/post/2008/06/Post-Redirect-Get.aspx
POST-REDIRECT-GET

Client

POST + form data

Redirect to URL

GET + URL

Server

after update, redirect browser to resource with results

no "resubmit?" on browser back

http://blog.andreloker.de/post/2008/06/Post-Redirect-Get.aspx
**Single Page Apps**

- Single Page App: all or almost all interaction occurs on a single web page
- Apparent multiple pages actually DIV's hidden and shown using Javascript and CSS
- Data retrieved using XHR (XMLHttpRequest), a.k.a. AJAX (asynchronous Javascript and XML)
- POST-REDIRECT-GET not needed with XHR
RESTFUL APIs

- URL's for XHR calls designed following RESTful principles
  - GET to retrieve data with no change in server state
  - POST to modify server state
    - Also PUT and DELETE
  - Decouples the service from the client
  - Enables wider user of the service, e.g., mashups