3: Building the MIT Mobile Web

MIT Mobile Project Team
Albert Chow, Sonya Huang, Eric Kim, and Brian Patt

Information Services & Technology

Agenda

- Introduction
- Designing the MIT Mobile Web
- Building the MIT Mobile Web
- Technical architecture
- Browser detection
- Content generator
- Sample modules
- Q&A

Designing the MIT Mobile Web

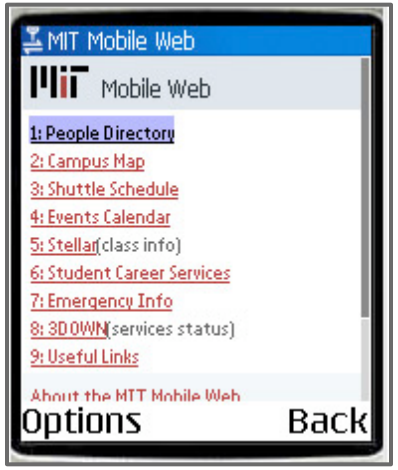
The mobile web?

Designing for the mobile context

Designing for the mobile context

- Physical constraints
  - Small, dense screens
  - Limited input and feedback
- Technical constraints
  - Data is still relatively slow and expensive
  - Limited security and personalization
  - Wide variety of physical form factors
    - Screens: from 128x160 up to 480x640 pixels
    - Input: touchscreen, numeric keypad, full QWERTY keypad
    - Data speed: 3G “high-speed”, low-speed (dialup-class)
  - Limited web browsers with very wide range of capabilities and standards compliance

Device and browser differences

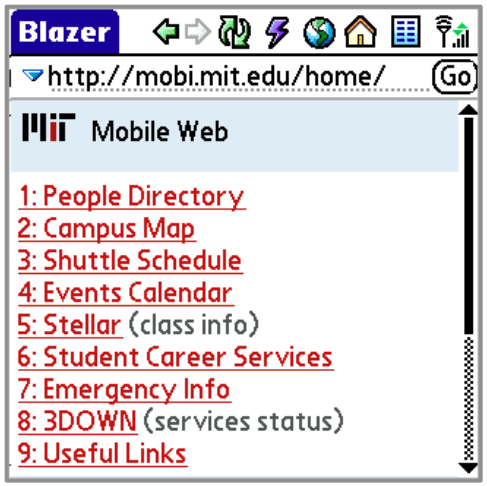


Nokia N70

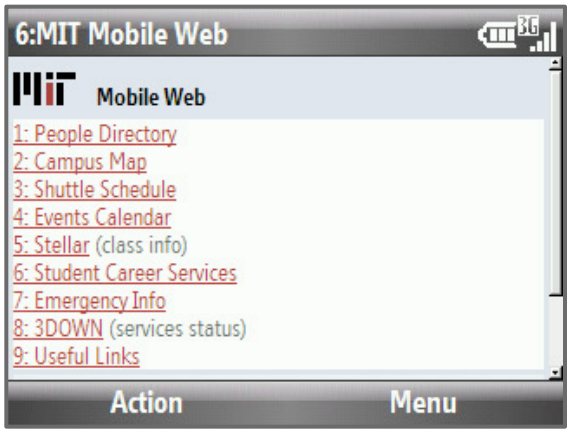
Sony Ericsson K750i

Motorola RAZR v3i

BlackBerry 8830



Palm Treo 755p



Motorola Q9h (Windows Mobile)

Device and browser differences

Designing for the mobile context

- Attention considerations
  - Seconds vs. minutes
  - Sovereign vs. transient tasks
  - *In situ* activities
- Mobile = new opportunity
  - Reach your community where they are most of the day
  - Meet low-level needs in an unexpectedly useful and fun way

Early design decisions at MIT

- Carefully curated content and functionality
- Narrow and shallow structure
- Simple navigation with as many accelerators as possible
  - Consistent global navigation
  - Consistent application-level navigation
  - Accesskeys
- Segmentation and adaptation

Segmentation and adaptation

- Impossible to design for every device
- Didn't want lowest-common-denominator experience
- Solution: device segmentation

Feature phones

Flip, bar, slider phones with relatively basic browsers and smaller screens

Smartphones

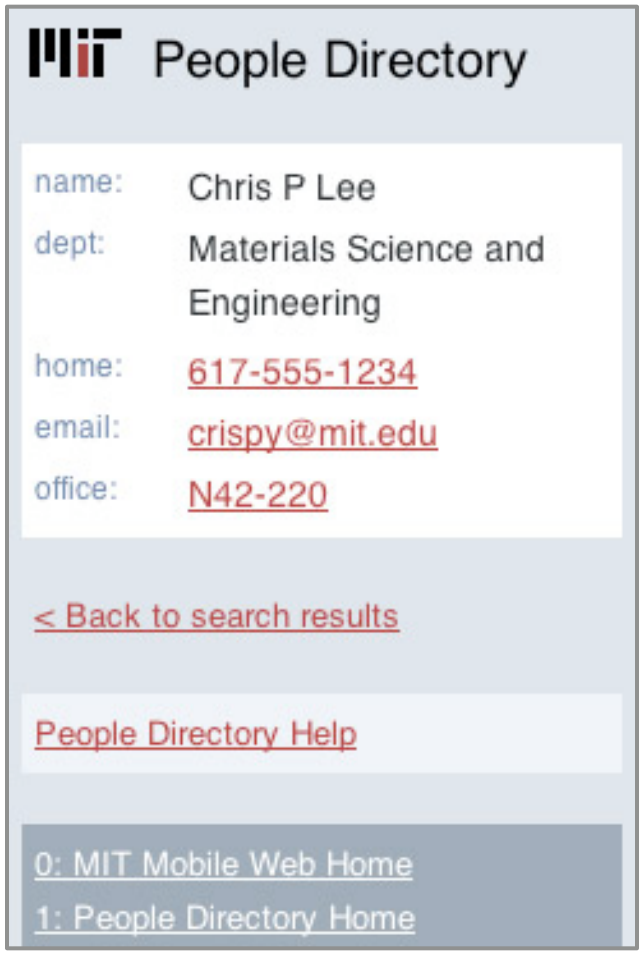
BlackBerry, Windows Mobile, PalmOS

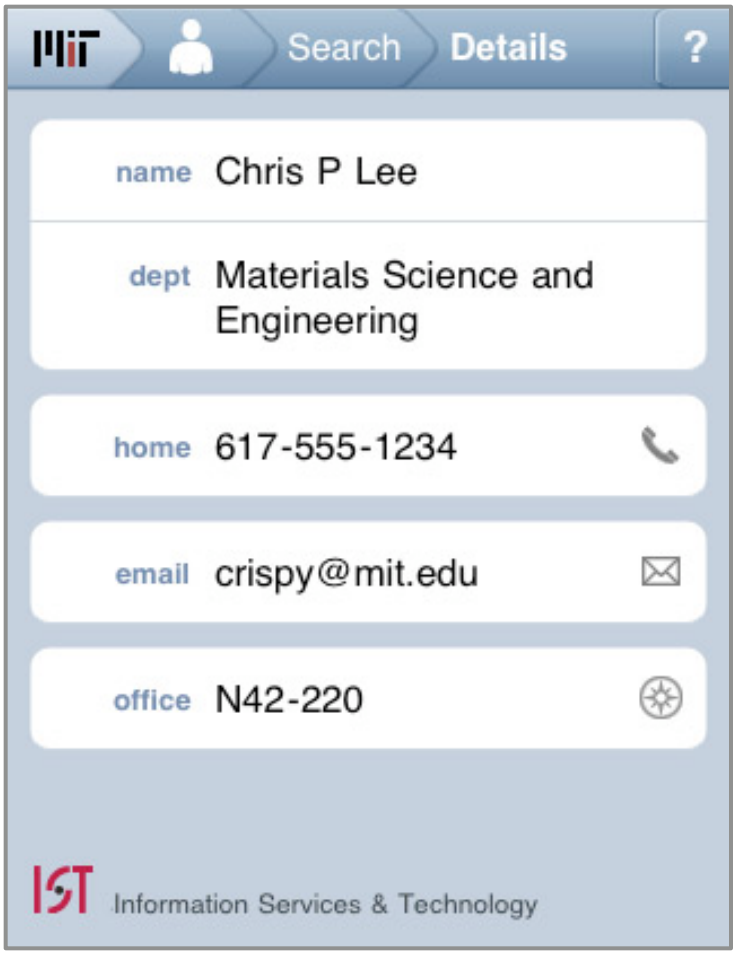
iPhone/iPod touch

Segmentation and adaptation

Same content & functionality across device classes

Different presentation and navigation for each device class





Key differences between “buckets”

- Level of:
  - Rich, semantic structure (HTML)
  - Style (CSS)
  - Interactivity (Javascript)
- Amount of content shown per page (pagination)
- Image sizes
- Location, organization and presentation of navigation

General design principles

- Feature phone and smartphone: converged on a simple, robust, text-centric design optimized for keypad navigation
- iPhone and iPod touch: take advantage of unique capabilities of the touch-driven platform
- Minimize distance to user goals
- Be great where we can
- Don’t suck anywhere

iPhone-specific design features

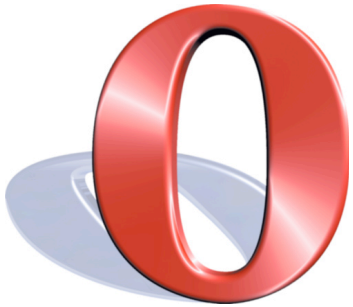
- Best standards compliance – things work the way they're supposed to
- Support for advanced CSS, including absolute positioning, layering, rounded corners, PNGs, transparency
- Full-screen mode
- Asynchronous data loading
- Layout re-flowing on device orientation change
- Apple's guides, HIG, code samples

iPhone-specific design features

iPhone-specific design features

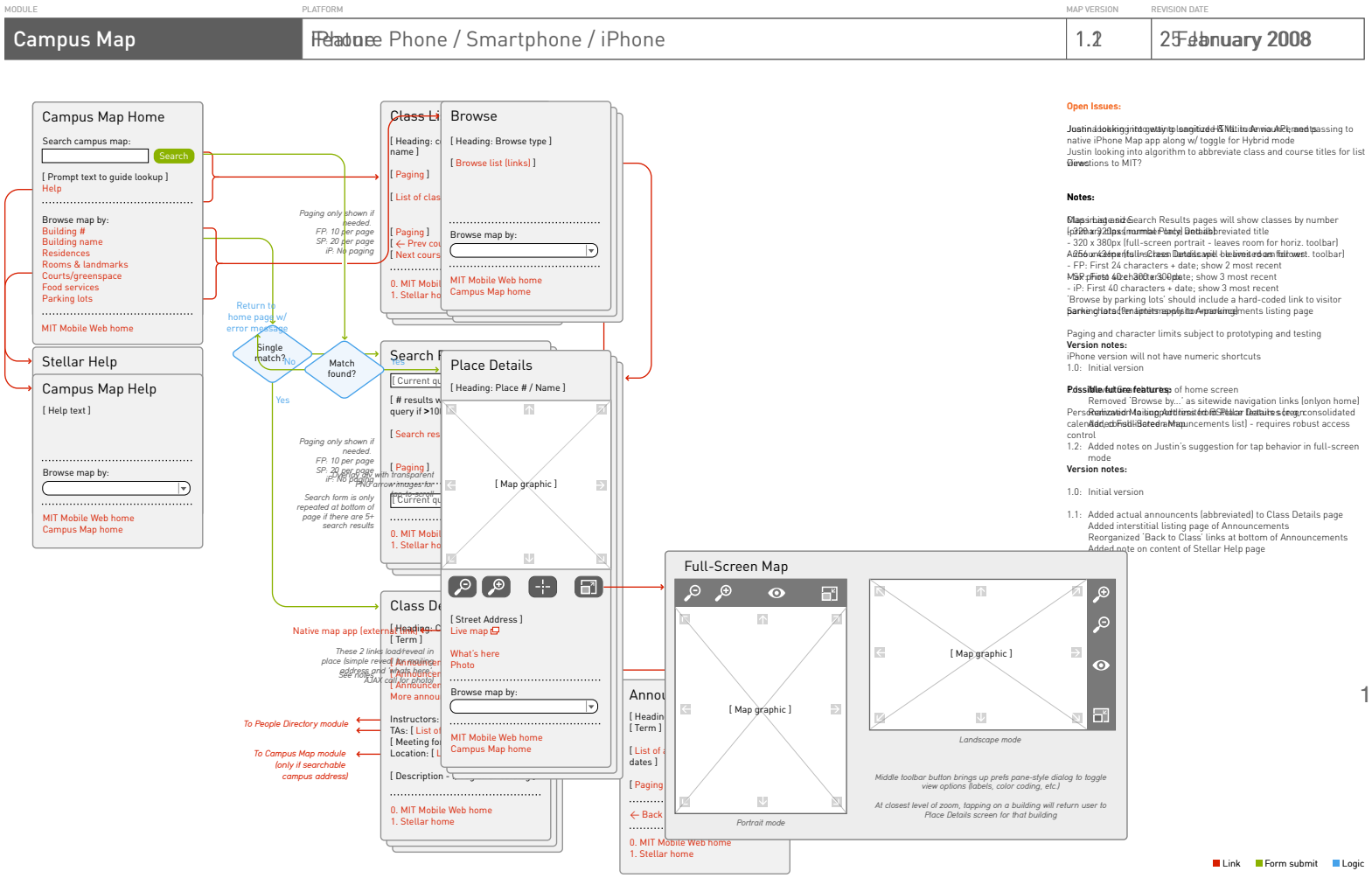
Segmentation and adaptation

- Challenge: Devices are always changing, previous category lines blurring and shifting
- Future: segment by browser capability, not hardware category
- Better exploit strengths of emerging advanced devices and platforms

Design process

1. User-experience (UX) mapping



Design process

1. User-experience (UX) mapping
2. Visual design and prototyping
3. Initial implementation
4. Usability testing
5. Remaining implementation and testing
6. Ongoing design and development

Conclusion

- Mobile web provides real-world daily utility for the college and university community
- Comes with unique and constantly evolving challenges
- Strategies for success:
  - Design for the mobile context
  - Segment target devices and adapt content, navigation and functionality
  - Test and refine
  - Continue to track, re-think, re-design and re-test your site as the mobile market evolves

Further reading

- *Mobile Web Design:* http://mobilewebbook.com/
- dotMobi Mobile Web Developer's Guide: http://dev.mobi
- Global Authoring Practices for the Mobile Web: http://www.passani.it/gap
- WURFL: http://wurfl.sourceforge.net
- Device Atlas: http://deviceatlas.com
- Device Anywere: http://deviceanywhere.com
- Apple iPhone web and application guidelines (free Apple Developer Connection account required): http://developer.apple.com/documentation/AppleApplications/Reference/SafariWebContent http://developer.apple.com/webapps

Building the MIT Mobile Web

Building the MIT Mobile Web

How to start building a mobile web site:

1. Select modules
2. Find data sources
3. Select devices to support
4. Set up server

…or use the MIT Mobile Web, which will be open sourced

MIT Mobile Web modules

Criteria for module selection

- What's most useful on the go
- Most readily available data sources
- Avoided "screen scraping"
  - Takes more development time
  - Not robust

System architecture

Browser Detection

iPhone | Smartphone | Feature Phone | Other

Content Generator

People Directory | Campus Map | Shuttle Schedule | Events Calendar | Stellar | Student Careers | Emergency Info | 3Down

MIT LDAP Server | MIT IMS Server | NextBus | MIT Events Server | MIT Stellar Server | Careers Office | Emergency .mit.edu | 3Down .mit.edu

System configuration

- Hardware (Virtual Server):
  - CPU: Intel Xeon 3GHz
  - Memory allocation: 2 GB
  - Storage: 15 GB
  - VMWare
- Software:
  - Operating system: Red Hat Enterprise Linux 4
  - Web server: Apache 2.0.52
  - PHP 5.2.5, MySQL 5.0.18, Python 2.4.4

Browser detection

Browser Detection

iPhone | Smartphone | Feature phone | Computer | Spider

WURFL

Wireless Universal Resource File (WURFL)

- Open source data source
- Attempts to catalog all mobile devices
- Initial goals:
  - Determine screen size
  - Determine phone features
- Current implementation:
  - Distinguishes phones from computers
  - Updated database of User-Agent strings
  - Keywords to pick smart phones (iPhone, iPod, BlackBerry, …)

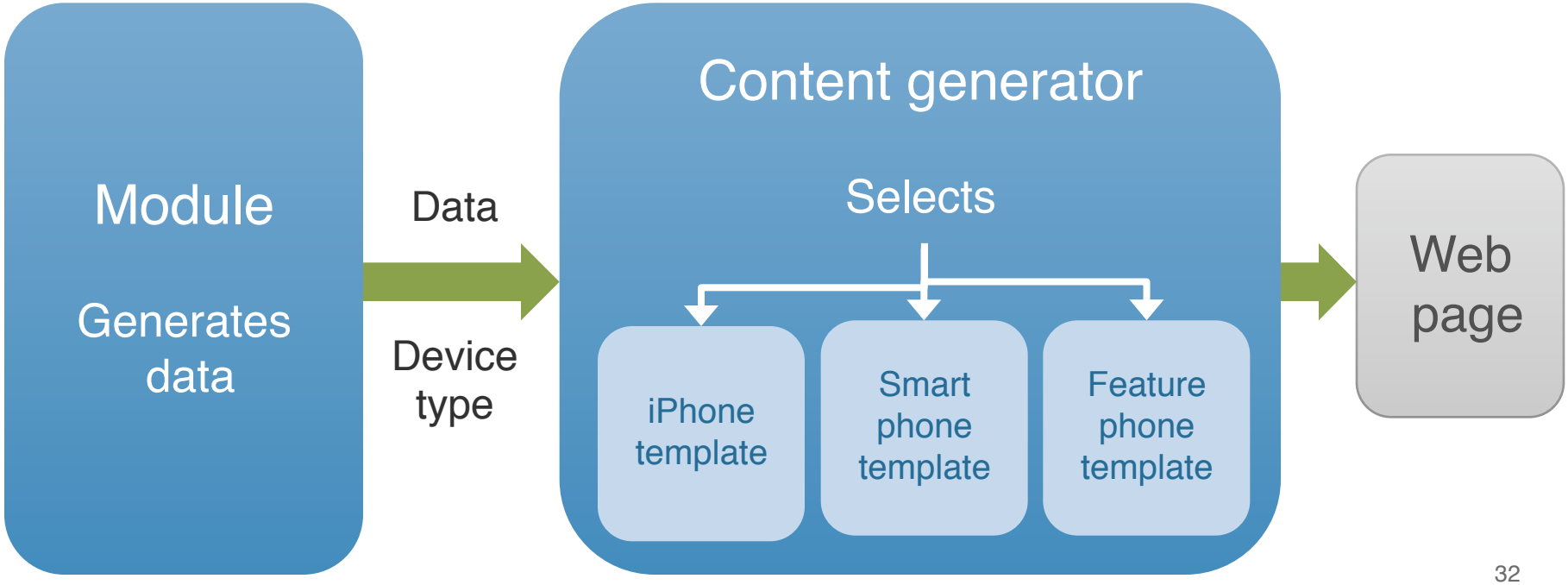
User agent strings

Complicated and non-standardized

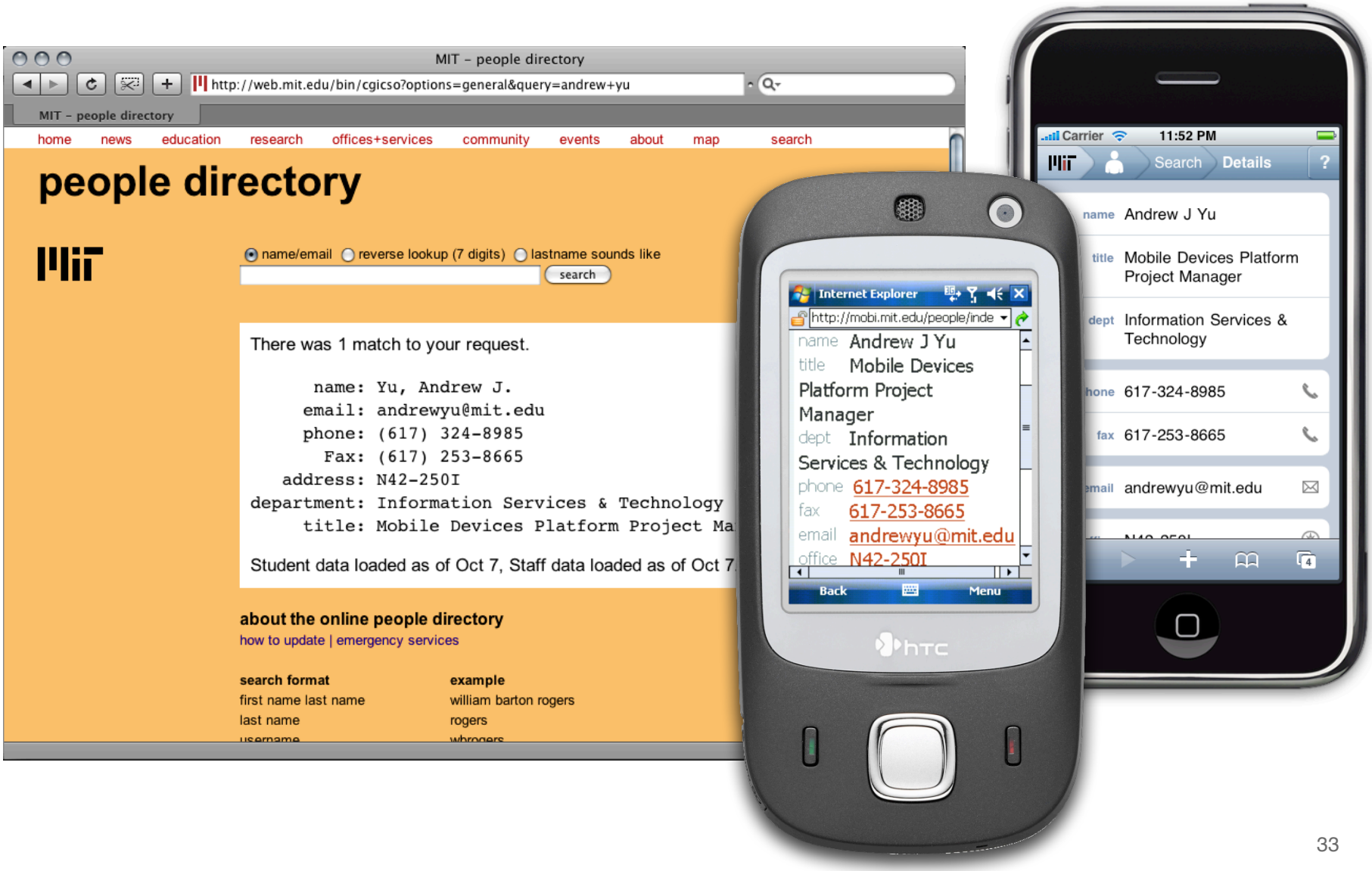
- iPhone:
  Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+(KHTML, like Gecko) Version/3.0 Mobile/1A538a Safari/419.3
- BlackBerry:
  BlackBerry8130/4.3.0 Profile/MIDP-2.0 Configuration/CLDC-1.1 VendorID/105
- Search for closest match

Content generator

- Three different HTML templates – one for each "bucket" of devices
- Each HTML template includes its own CSS (style sheets) and images

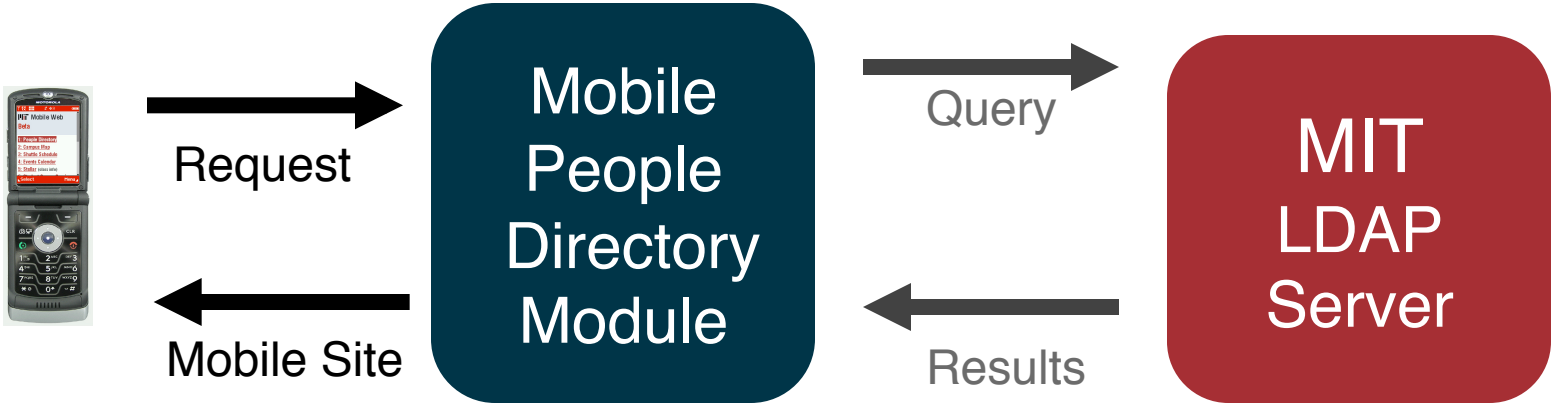


People Directory



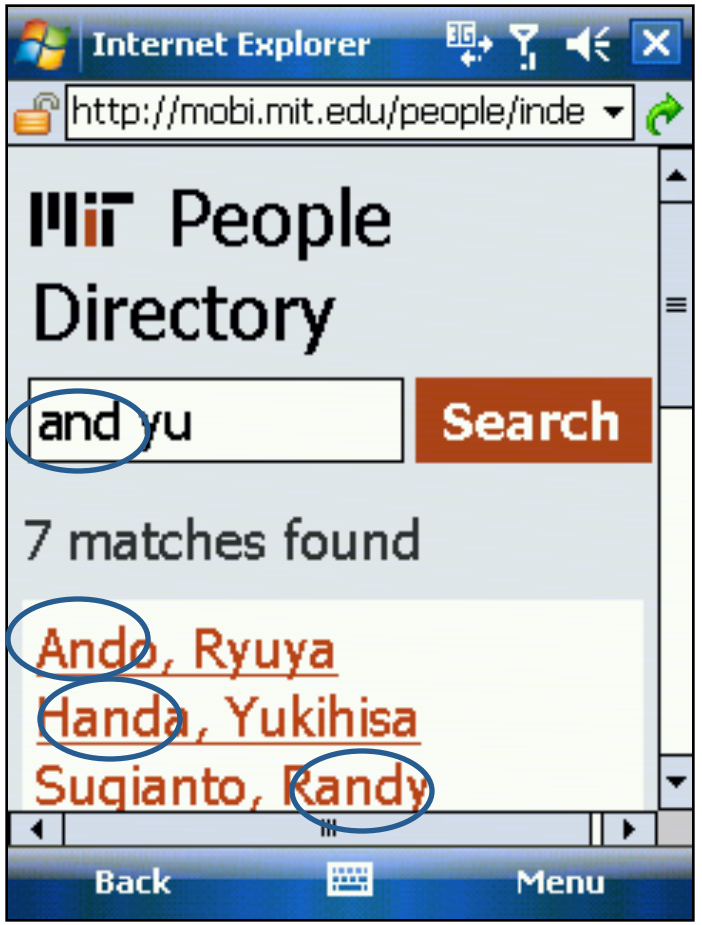
People Directory mobile backend

- LDAP (Lightweight Directory Access Protocol)
  - Uses standard query language
  - Widely used
  - PHP has a built in interface



Optimized for mobile experience

- Difficult/slow to type on mobile devices
- Optimize search results:
  - Single letter = first initial of first or last name
  - Default sub string search
  - Exact email matches at the top



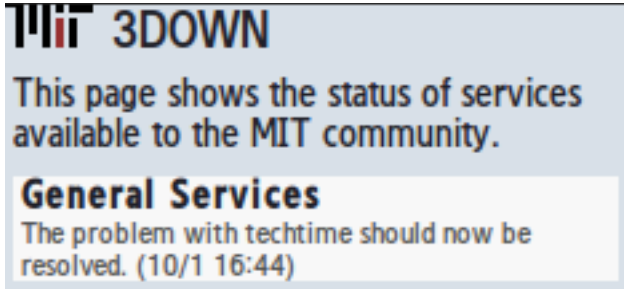
3DOWN

- Status of major services at MIT, such as internet, email, and telephone
- Backend: RSS (Rich Site Summary) Feed
  - Contains title and content
  - Widely used, such as Wordpress and New York Times

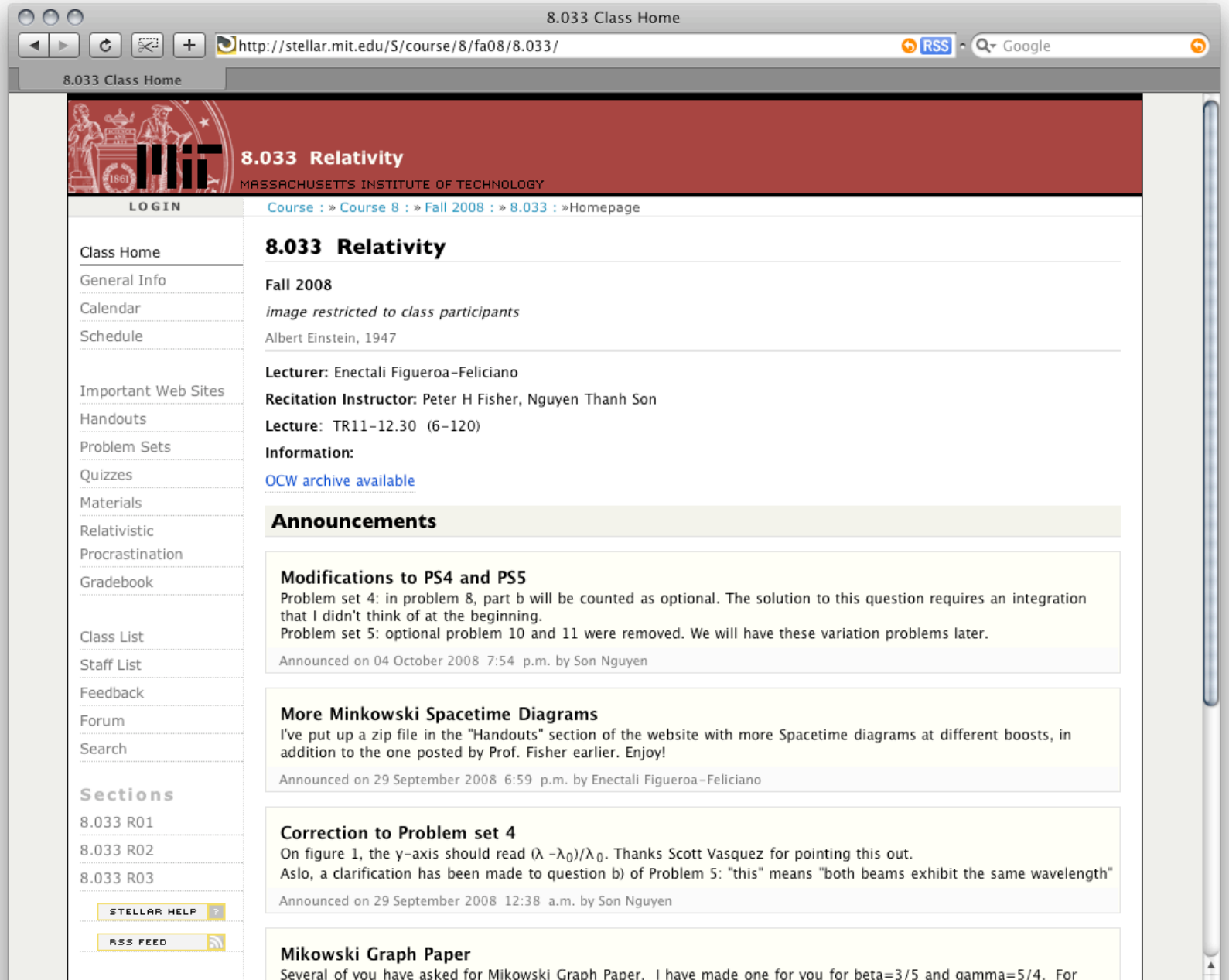
RSS good for mobile devices

- RSS is list oriented
- Phones are good at displaying lists
- Easy to reformat content for mobile devices

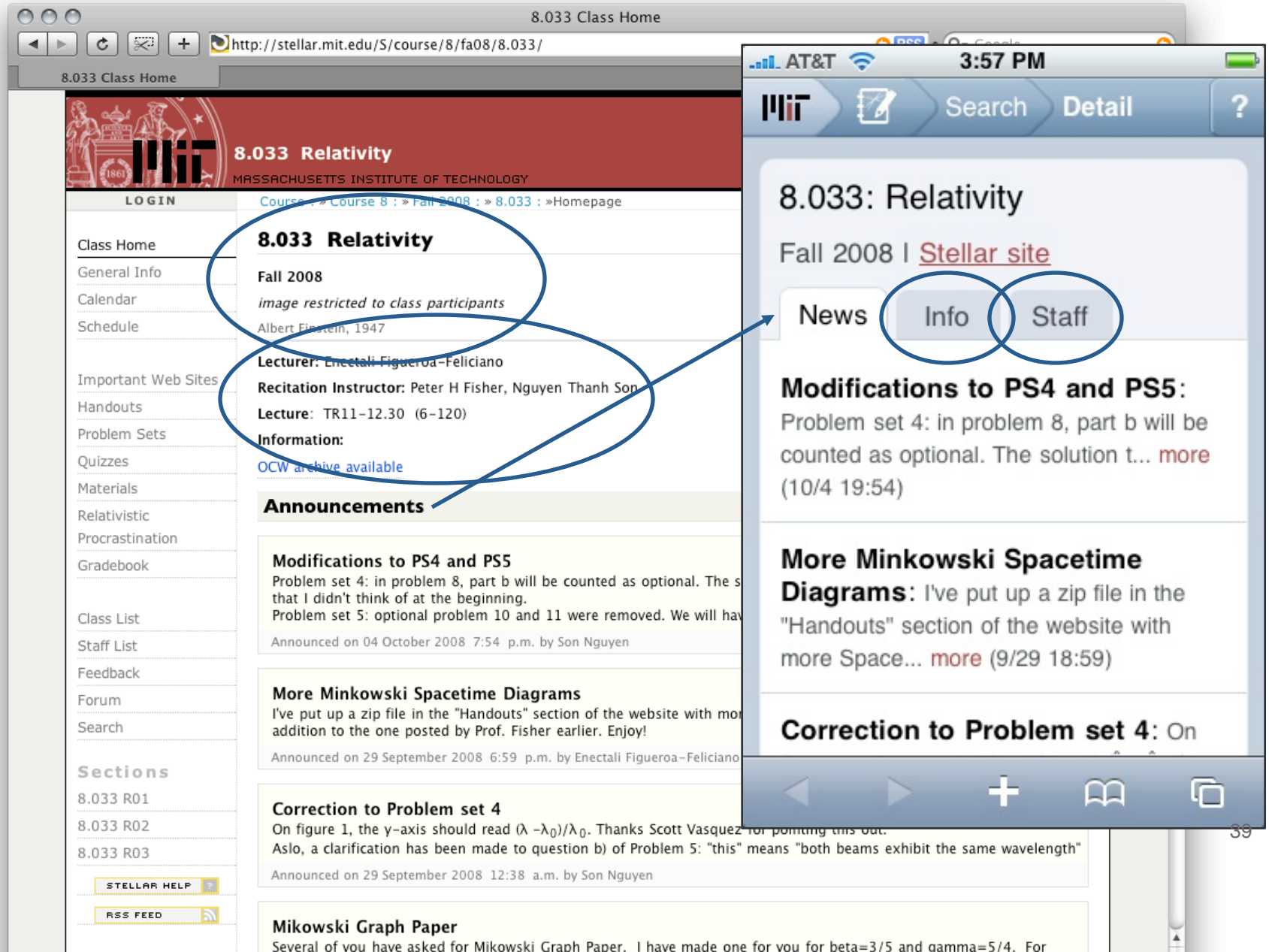
```
<ttl>5</ttl>     <item>
  <title>
  General Services
  </title>
  <link>http://3down.mit.edu/</link>
  <description><![CDATA[
  The problem with techtime should now be resolved.
  ]]>
  </description>
  <pubDate>Wed, 1 Oct 2008 16:44:13 EDT</pubDate>
```



Stellar – web version



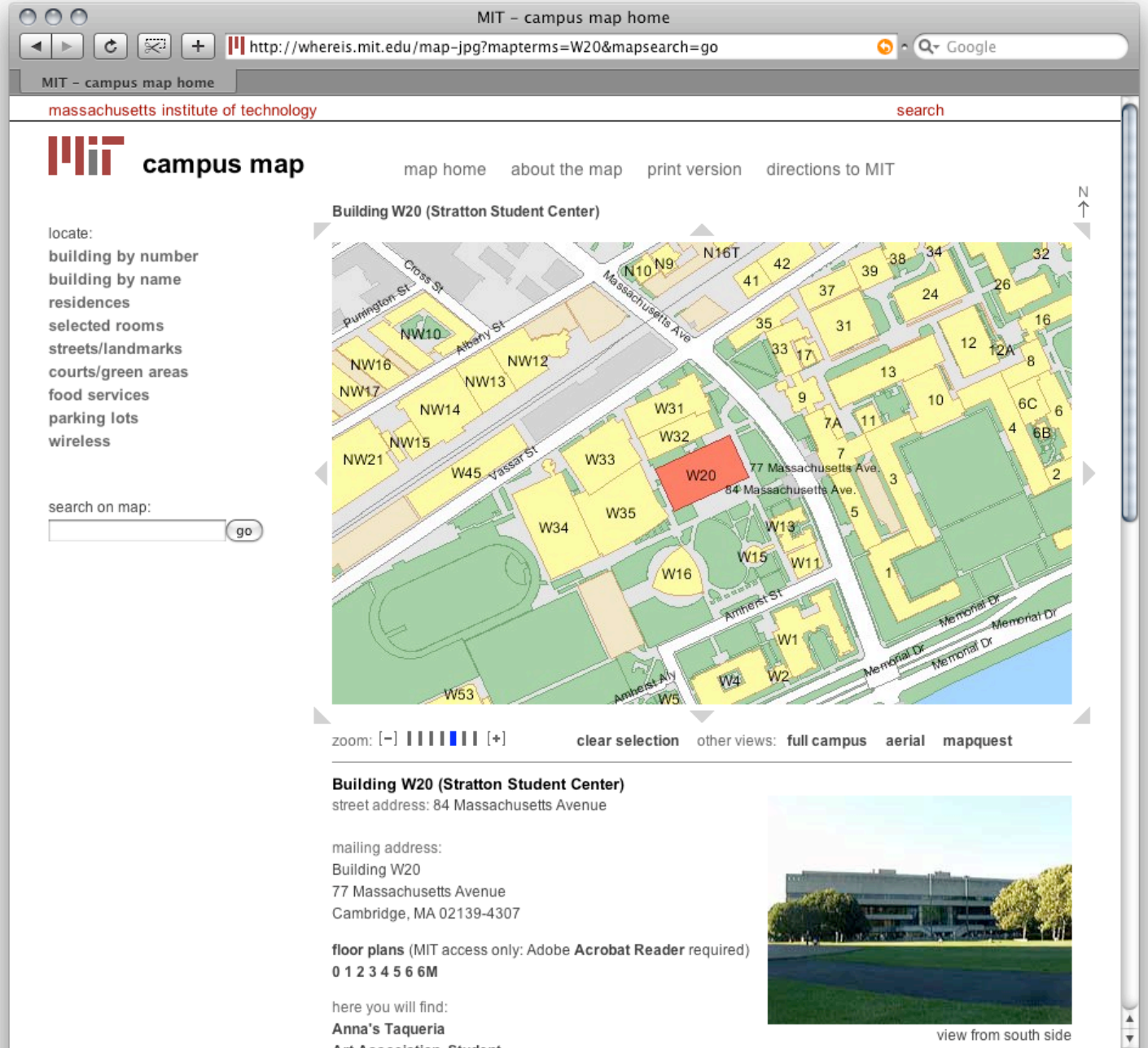
Stellar – mobile version



Stellar backend

- Stellar uses several different data sources
- Aggregate into a custom XML (eXtensible Markup Language) feed
  - Course guide is an XML feed
  - One RSS per class for course announcements
  - Not given access to protected information
- Why use XML?
  - Easy to inspect
  - Easy to expand to any text based data
  - Widely used for internet applications

Campus Maps – web version



Campus Maps – backend

Web Map Services (WMS)

- Generates a map
- Based on picture size, frame size, and map elements to display
- Parameters passed via URL
- Widely used standard
- Requires a two step process:
  - Request frame
  - Request picture

Campus Maps – backend

- Example: Map of W20 (Student Center)
- Parameters :

  ?request=getmap&version=1.1.1&

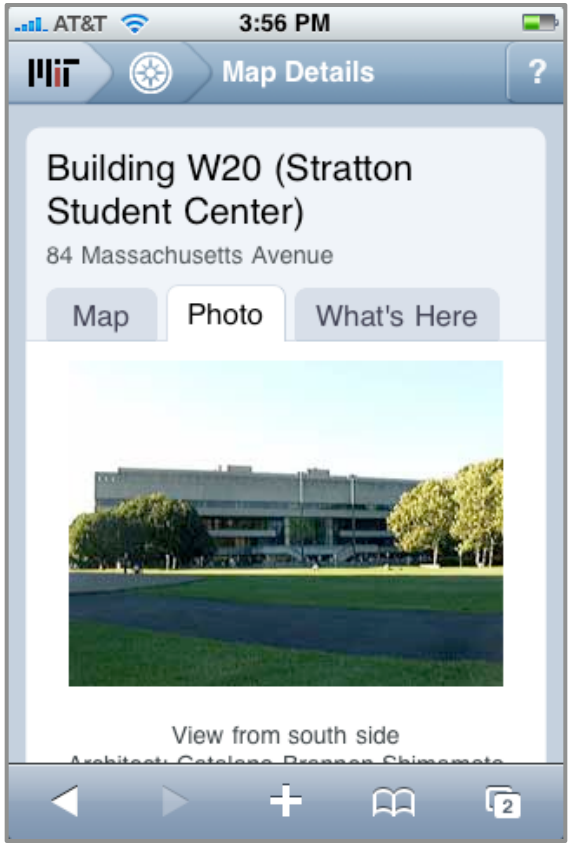
  image size *width=220&height=160&*

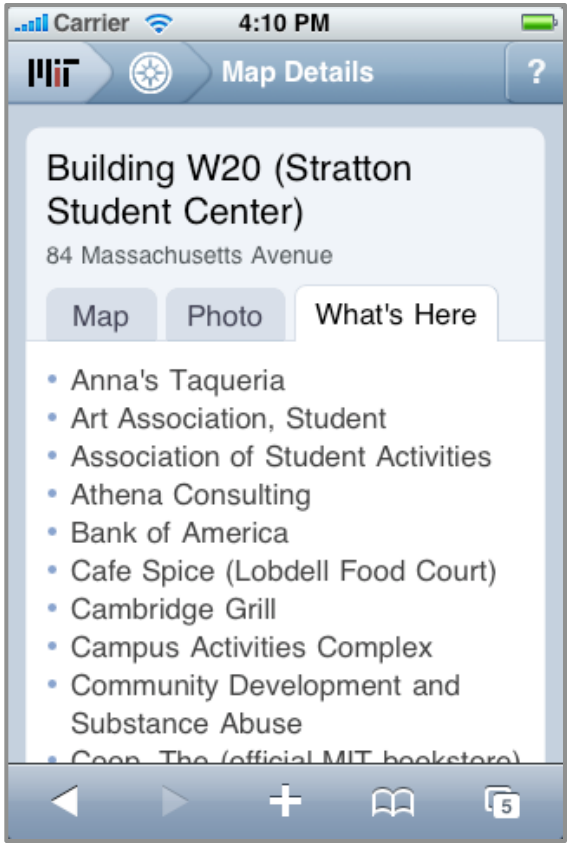
  building *selectvalues=W20&*

  frame *bbox=708882,495087,709896,495825&*

  map elements to display *layers=Greenspace,Roads,Parking, Buildings*

Campus Map – mobile version





Shuttle Schedules

- Created a data file of the published bus schedule
- Stored in MySQL database

Boston Daytime

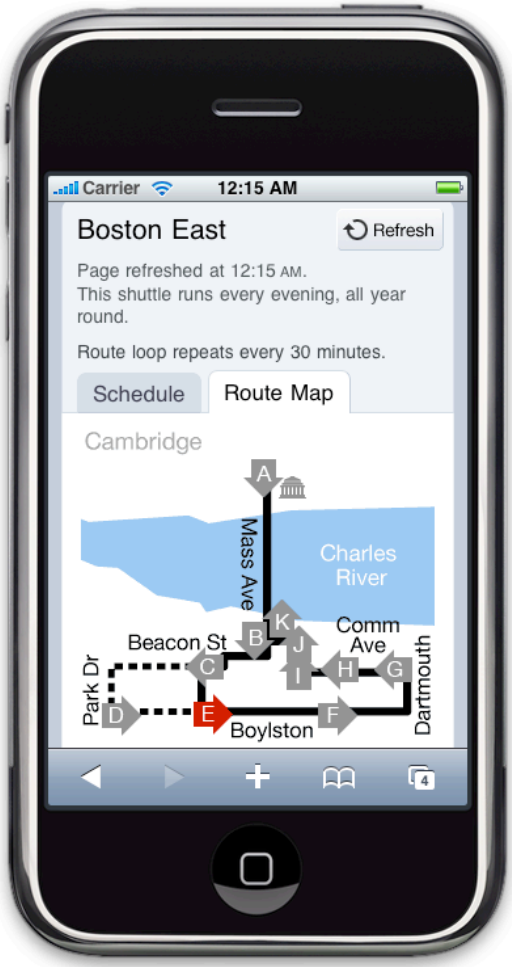
RUNS SEPTEMBER THROUGH MAY

1	84 Mass. Ave.	8:07	8:27	8:47	9:07	9:27	9:47	10:07	10:27	10:47	11:07	11:27	11:47
2	Mass. Ave./Beacon	8:09	8:29	8:49	9:09	9:29	9:49	10:09	10:29	10:49	11:09	11:29	11:49
3	487 Comm. Ave. (PSK)	8:10	8:30	8:50	9:10	9:30	9:50	10:10	10:30	10:50	11:10	11:30	11:50
4	64 Bay State (TXI)	8:11	8:31	8:51	9:11	9:31	9:51	10:11	10:31	10:51	11:11	11:31	11:51
5	478 Comm. Ave.	8:14	8:34	8:54	9:14	9:34	9:54	10:14	10:34	10:54	11:14	11:34	11:54
6	450 Beacon St.	8:19	8:39	8:59	9:19	9:39	9:59	10:19	10:39	10:59	11:19	11:39	11:59
7	77 Mass. Ave.	8											

```
$schedule
      ->route("Boston Daytime", "boston")
      ->summary("Runs weekdays 8AM-6PM, Sep-May")
      ->except_holidays()
      ->perHour(3)
      ->stops(
         st("84 Mass. Ave."                ,"mass84_d", "boston"   , '07'),
         st("Mass. Ave. / Beacon"          ,"massbeac", "cambridge", '09'),
         st("487 Comm. Ave. (PSK)"         ,"comm487" , "cambridge", '10'),
         st("64 Bay State (TXI)"           ,"bays64"  , "cambridge", '11'),
         st("478 Comm. Ave."               ,"comm478" , "cambridge", '14'),
         st("450 Beacon St."               ,"beac450" , "cambridge", '19'),
         st("77 Mass. Ave."                ,"mass77"  , "cambridge", '23'))
      ->addHours("Mon-Fri", hours("8-17"));
```

Shuttle Schedules

- Route maps hand-drawn for different screen sizes



Summary of MIT Mobile Web

- Module selection:
  1. Usefulness (e.g., People Directory, shuttle, etc.)
  2. Easy to integrate data sources = less interface code
     - Open data standards (e.g., LDAP, RSS, and XML)
- Device optimization / selection:
  - UI optimized for three categories (“buckets”)
  - iPhone, smartphone, and feature phone
- Optimized mobile experience
  - Only display essential information (device dependent)
  - Code to reduce the amount of input needed

Q&A

Thank you!

Design:

Eric J. Kim

eric@chemicalcreative.com

Development:

Albert Chow, Sonya Huang, and Brian Patt

MangoText

mangotext@mit.edu