Website Design
Small Screen Design
Responsive Design
How Do We Really Use the Web?

- Do users **carefully read** or **scan and click**?
- Facts of life:
  - **We don’t read** pages, **we scan** them
  - We don’t make optimal choices, we **choose** the **first reasonable option**
    - Little downside for wrong guesses
  - We **muddle through** without always understanding how things work
Browse or Search?

- **Browsing**
  - Versus the real world – **no sense** of scale, direction, or location

- **Searching** - users are really not that good at forming effective queries

- So **help the user** find the desired page
  - Auto complete
  - Auto suggest to disambiguate
  - Suggest keywords
Site Evolution

- **Informational** sites:
  - Balance display density of useful information with learnability for infrequent users
  - Full screen content with good page navigation

- **Transactional** sites
  - Properties of informational sites plus functional behaviors
  - Efficient structured navigation based on an “information architecture” page content organization

- **Web application** sites:
  - Desktop-like more complex applications
  - “Views” more than “pages” – not a “document” metaphor
  - Asynchronous server communications
Designing for the Web

Current landscape ...

- **HTML5 + CSS3 + JavaScript** to build a wide array of sophisticated “rich Internet applications”
- **Reusable GitHub** based open source **UI components**; e.g., Bootstrap, jQuery
- Modern **browsers efficiently** process HTML and JavaScript
- Widely accepted design **conventions** and **patterns** have evolved
- Good foundational design principles still apply!

[W3Schools Web Tutorials](http://www.w3schools.com)
Some Web Design Guidelines

<table>
<thead>
<tr>
<th>Home Page</th>
<th>Page Layout</th>
<th>Navigation</th>
<th>Information Presentation</th>
</tr>
</thead>
</table>
| • Sites value and purpose  
• Positive first impression  
• Limit to one screen  
• Keep it simple  
• Access to site’s major features | • Consistent grid – visual hierarchy, element alignment  
• Header and footer boundaries | • Page navigation supports associative information architecture  
• Primary(global) sections at top, secondary(local) on left, “utilities” at bottom  
• A way to search  
• Page and link names match  
• Navigation markers (e.g., breadcrumbs)  
• No dead end pages  
• List of contents for long scrollable pages  
• Good link affordances | • Simple background images for readability  
• Distinguish important images from aesthetics  
• Have clear and useful reasons for using multimedia |
Typical Page Layout

- **Site ID**
- **Local Navigation**
- **Sections**
- **Page Name**
- **Search**

---

**Amazon.com**

- **Header**
  - Logo
  - Navigation links
- **Main Content**
  - Search bar
  - Section links
  - Books at Amazon
- **Category Links**
  - Shop by Category
  - Popular in Books
  - More in Books
- **Promotions**
  - Prime Day deals

---

**R.I.T Software Engineering**

S. Ludi/R. Kuehl
Single Page Web APPs

- What are they – **single** [long] scrollable **page** with associated **client side code** for **server side data access**
- More appropriate for smaller sites
- **Claim** is **simpler** user navigation and understanding, simpler design, deployment, and maintenance
- However …
  - May be **more complex** – e.g., may have to duplicate in code existing browser services such as history for back button refresh
  - May be **performance tradeoffs** – initial load latency, variable client performance
  - More code means more opportunity for bugs
  - Less effective search engine optimization
Usability and Small Screens

SE444
“The phrase ‘mobile usability’ is pretty much an oxymoron. It's neither easy nor pleasant to use the Web on mobile devices.”

“designing for mobile is hard”

“It’s not enough that a site will display, can the user get things done?”

 Jacob Nielsen, useit website
Small Screens

- We will focus on **consumer mobile devices** such as smartphones and tablets.
- However, there is another large category of **embedded devices** to be considered as well.
- Design guidelines apply equally to those devices with more constraints such as safety.
Mobile Usability Problems (Opportunities)

- **Small screens** (inherent)
- **Awkward input** (“fat finger syndrome”)
- **Network performance and reliability**, especially for downloads (but getting better)
- **Mis-designed websites** – designed for the desktop just makes it worse (but getting better)
### Small Screen Design Guidelines

<table>
<thead>
<tr>
<th>Screen Layout</th>
<th>Navigation</th>
<th>Information Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• To preserve screen real estate ...</td>
<td>• Limit navigational hierarchy, especially global to contextual transition</td>
<td>• Rapid serial presentation of text, important information first (progressive disclosure);</td>
</tr>
<tr>
<td>• Use transparency; e.g., widgets</td>
<td>• Apply Fitt’s Law: large objects for navigation (touch) versus hypertext</td>
<td>• Minimize extended scrolling or paging</td>
</tr>
<tr>
<td>• Vertical or horizontal screen navigation</td>
<td>• Apply screen layout design patterns; E.g.</td>
<td>• Optimize</td>
</tr>
<tr>
<td>• Use images sparingly</td>
<td>• Carousels</td>
<td>• 14pt fonts</td>
</tr>
<tr>
<td>• Minimize use of footers, breadcrumbs, progress indicators, menu bars</td>
<td>• Stacks</td>
<td>• Organize text with headings</td>
</tr>
<tr>
<td>• Most frequently used controls at bottom</td>
<td>• Lists</td>
<td>• Minimize search to avoid complex data entry (voice okay)</td>
</tr>
<tr>
<td>• Consider the physical feel – ergonomics,</td>
<td>• Rapid serial presentation of text, important information first (progressive disclosure);</td>
<td></td>
</tr>
<tr>
<td>• The use of the (one) hand – fingers</td>
<td>• Minimize extended scrolling or paging</td>
<td></td>
</tr>
<tr>
<td>• Finger tip target size guides (e.g., iPhone 44 pixels)</td>
<td>• Optimize</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 14pt fonts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Organize text with headings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Minimize search to avoid complex data entry (voice okay)</td>
<td></td>
</tr>
</tbody>
</table>
Responsive Design for Web Applications
What is the Problem?

- **Mobile** web access is ubiquitous
- One interface design does not fit all screens for optimal user interaction
  - Mobile users may have different needs from desktop users
- So why not make designs flexible to dynamically match the screen environment?
Web App vs. Native App?

- **Web App**
  - Develop once, lower support costs
  - Cross device platform support
  - Dependent on a network connection
  - May be functional limitations

- **Native app**
  - More expensive to develop and support
  - Not portable
  - Better performance and security
  - Use local hardware
  - Better UX?
  - App store distribution

- **Hybrid app?** Native app accesses website data
Responsive Web Design

- Create a **single website** that works effectively on the **desktop** as well as **mobile devices**
- Responsive websites **reorganize themselves automatically according to the device displaying them**
  - **Desktops/laptops** get the full experience – video, images, animation
  - **Smartphones** get a simplified experience that works quickly – app-like
  - **Tablets** – something in between
Responsive Web Design

- More than altering the layout based on viewport*
  - Invert the process of web design
    - Design for the smallest viewport first
    - Progressively enhance the design and content for larger viewports

* Viewport is display area versus physical screen size
Responsive Design Example

http://www.andthewinnerisnt.com/

Check out the CSS File – look for @media
Responsive Design Guidelines

**Group similar devices** by screen size/media type to establish target size “breakpoints” for design

For each media type, identify **unique properties** and **shared properties** that will vary by value (e.g. font size)

Design **adaptive layouts** – e.g., large menu bar on the desktop, dropdown menu on smartphone

Use “fluid grids” - **proportional layouts** scaled by screen dimensions (scale factor)

**Tailor** the amount and type of **content** by screen size

Use CSS3 (@media query) and HTML5 encoding
Cascading Style Sheet (CSS) Media Queries

- **CSS3 “@media” query** – query “screen” as media type with **screen properties** such as **size** and **resolution**
  - All following style sheet rules apply to that media type
  - **Substitute** different **layout commands** or a **tailored CSS file**
  - **Scale** to match device screen resolution and size
  - **Transform** screen layout – e.g., number of columns of content
  - **Adjust object size** such as for links (Fitt’s Law)
  - **Adjust typography** – e.g., font size, line width and length

```css
@media print {
  body { font-size: 10pt }
}
@media screen {
  body { font-size: 13px }
}
@media screen, print {
  body { line-height: 1.2 }
}
```

https://www.w3schools.com/css/css3_mediaqueries.asp
CSS Media Query Example

- @Media rule. What happens?

```css
body {
    background-color: grey;
}
@media screen and (max-width: 960px) {
    body {
        background-color: red;
    }
}
@media screen and (max-width: 768px) {
    body {
        background-color: orange;
    }
}
@media screen and (max-width: 640px) {
    body {
        background-color: yellow;
    }
}
@media screen and (max-width: 320px) {
    body {
        background-color: green;
    }
}
```
CSS Media Queries for Popular Form Factors

Smartphones
Portrait and Landscape
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) { ... }
Landscape
@media only screen and (min-width : 321px) { ... }
Portrait
@media only screen and (max-width : 320px) { ... }

Tablets, Surfaces, iPads
Portrait and landscape
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) { ... }
Landscape
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape) { ... }
Portrait
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait) { ... }

Desktops, laptops, larger screens
@media only screen and (min-width : 1224px) { ... }
Large screen
@media only screen and (min-width : 1824px) { ... }
Gotcha – Cross Browser Compatibility and/or Obsolescence

- **Graceful degradation** – design for modern browsers but assure a useful experience on older browsers

- **Progressive enhancement** – start with standard markup for all browsers and enhance the experience for more capable browsers – recommended

- Modernizr – open source JavaScript library that feature tests a browser’s capabilities

- Polyfill – downloadable code that provides capabilities missing from the native browser (e.g., HTML5 features)
Web App References

- “Don’t Make Me Think”, Steve Krug
- “Designing for Conversion; Evaluating decision making through HFI’s PET Design™”, Mona Patel
- “About Face”, Cooper, Reimann
Mobil App References

- Lari Kärkkäinen and Jari Laarni, "Designing for Small Display Screens", NordiCHI, October 19-23, 2002
- uxmatters.com: Usability for Mobile Devices
- Josh Clark, Tapworthy – Designing Great iPhone Apps, O’Reilly Media, 2010
Responsive Design References

- Marcotte, Ethan (May 25, 2010). "Responsive web design". A List Apart
- Frain, Ben, Responsive Web Design with HTML5 and CSS3 (eBook)