EVALUATING USER INTERFACES

Is this any good?

My Online Services

- Transcripts
- My Record
- Placement
- Degree Audit
- e-Learning
- Evaluations
- Financial Aid
- University Bursar
- Registration

Registration Prep
Degree Audit, Reg Start Time, Current Holds

Register Now
- Spring
- Summer
- Fall

My Schedule
- Spring
- Summer
- Fall

Request to Withdraw From All Courses
- Withdraw
- Status

My Textbooks
- Spring
- Summer
- Fall

WELCOME TO ISIS

Available 7:00am - 4:30am EST, Monday - Saturday; 10:00am - 4:30am Sunday.
Questions? Contact Us

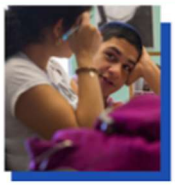
Special Announcements

- **Monday, January 20:** The Office of the University Registrar will be closed in observance of Martin Luther King Jr. holiday.
- **Fall 2013 Graduates:** Did I graduate?
- **Graduation Survey:** Fall graduates can help tell the Gator Story by completing the mandatory graduation survey. Completing the survey provides valuable information to UF and provides access to your transcript.
- Course Syllabi
- **Required Alcohol Education:** New undergraduates must complete the alcohol education tutorial during the first semester.
- Update your local address

Spring 2014

- January 17: Fees due by 3:30 pm, University Bursar
- January 17: Residency reclassification deadline
- January 24: S-U grade option deadline
- January 31: Degree application deadline
- January 31: Deadline to withdraw (all courses) with 25 percent refund (W assigned)
- April 11: Deadline to drop or add a course by college petition
- April 11: Drop/Withdraw deadline (without failing grades, W assigned)
- April 23: Classes end
- April 23: Honors theses due to college advising offices
- April 26, 28-30 and May 1-2: Final exams
- **May 2-4: Commencement**
- Dates and deadlines: Undergraduate and Graduate
- Information for veterans or their spouse/dependents

Summer 2014



- March 24 - May 6 and May 8: Summer A/C advance registration (at or after assigned start time)
- March 24 - May 6, then May 8 - June 26: Summer B advance registration (at or after assigned start time)
- May 9: Summer A/C regular registration ($100 late fee after 5:00 pm deadline)
- May 12: Summer A/C begins; classes end June 20 (A) and August 8 (C)
- May 14: Summer A degree application deadline
- June 27: Summer B regular registration ($100 late fee after 5:00 pm deadline)
- June 30: Summer B begins; classes end August 8

My Resources

- Academic Info
  - Alcohol Education Tutorial
  - Calendars/Deadlines
  - Catalog - Undergrad
  - Combined Degrees
  - Commencement Info
  - Course Evaluations
  - Course Syllabi
  - Distance Ed Courses
  - Exam Schedules
  - Forms - Registrar
  - Prestigious National Scholarships
  - Schedule of Courses
  - Study Abroad Ops
  - Ugrad Experience
- Advising Info
- Financial Info
- Grad Student Info
- Personal Info
- Other Info

FAQ

Goal

- You Should Be Able To
  - Explain common **approaches** to evaluating interfaces
  - Explain **when** to evaluate interfaces
  - Identify common **metrics** and **questionnaires** used in evaluating interfaces

Introduction

- Why evaluate?
  - Designers can become too entranced/opinionated. The focus is then on what the designer likes rather than what does the user like.
  - Experienced designers know extensive testing is required
- Factors affecting how to evaluate include
  - **stage** of design (early, middle, late)
  - **novelty** of project (well defined vs. exploratory)
  - **number** of expected users
  - **criticality** of the interface (life-critical medical system vs. museum exhibit support)
  - **costs** of product and finances allocated for testing
  - **time** available
  - experience of the design and evaluation team

Introduction

- How do you test…
  - a web site?
  - an air traffic control system?
- How much would you budget for testing? (http://en.wikipedia.org/wiki/List_of_most_expensive_video_games_to_develop)
- When do you test?
- Are you required to test? (e.g. military, government, safety)

What does testing **not** do?

- Guarantee perfection
- Hard to “finish” testing
- Difficult to test unusual situations
  - Military attack
  - Heavy load (e.g. voting)
  - Case study: Project ORCA
- Simulate accurate situations
  - E.g. driving games, military games, medical sims

Project ORCA

- http://arstechnica.com/information-technology/2012/11/inside-team-romneys-whale-of-an-it-meltdown/
- Web app for 37,000 volunteers at polls, 800 central trackers to track voters – who still needed to vote
  - Training
    - 90 minute training for 800 central volunteers
    - No training for field volunteers
  - Testing – local, not “real world”
    - System crash during actual load usage
  - Usage
    - Users’ first usage was 6 AM on election day
    - Confusion about “app” (where would you look for it)?
    - http vs https: no redirect to the secure site
    - Wrong pins and passwords
    - VoIP phones incorrectly configured – volunteers unable to provide support
  - Automated vs. Human testing
  - Difference in FL, OH, VA, CO was 520,000 votes

Expert Review

- Colleagues or Customers
  - Informal
  - Natural starting point
  - Ask for opinions
- Expert reviews
  - Formal
  - Considerations
    - What is an expert? User or designer?
    - How long? Half day to week
  - Different review methods from which to choose

Heuristic Evaluation

- Experts evaluate interface on design heuristics
  - Shneiderman's Eight Golden Rules
  - Nielsen's Heuristics
    - https://www.youtube.com/watch?v=6Bw0n6Jvwxk
    - https://www.youtube.com/watch?v=Y6B-u173hdQ
- Specific to application area
  - Heuristics for gaming (Pinelle et al 2008)
    - Provide consistent responses to user's actions (1)
    - Allow users to customize video and audio setting, difficulty, and game speed (2)
    - Provide users with information on game status (3)
    - …etc

Eight Golden Rules of Interface Design

1. Strive for consistency
2. Cater to universal usability
3. Offer informative feedback (e.g. button clicked)
4. Design dialogs to yield closure (e.g. confirmation page)
5. Prevent errors (e.g. gray out menu items)
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load (e.g. 7 ± 2)

Cognitive Walkthrough

- Experts “simulate” being users going through the interface
- Tasks are ordered by frequency
- Good for interfaces that can be learned by “exploratory browsing” (Wharton 1994) [novices]
- Usually walkthrough by themselves, then report their experiences (written, video) to designers
- Useful if application is geared for a group the designers might not be familiar with:
  - Military, Assistive Technologies

Formal Usability Inspection

- Experts hold courtroom-style meeting
- Each side gives arguments (in an adversarial format)
  - Merits and weaknesses of interface
- There is a judge or moderator
- Good for novice designers and managers
- Extensive and expensive
  - Requires time to prepare and more personnel than other approaches
- An example of formal usability inspections in practice at Hewlett-Packard
  - http://www.sigchi.org/chi95/proceedings/intpost/cg_bdy.htm

Expert Reviews

- Can be conducted at several times in the design process
- Focus on comprehensiveness over specific features or random improvements
- Rank recommendations by importance
  - HIGH: Shorten log in procedure – because users are over busy
  - MIDDLE: Reordering sequence of displays, removing nonessential actions, providing feedback.
- Less vital fixes and novel features for future releases

Conceive Design Develop Test & Refine Alpha Beta Post-Release

Expert Review

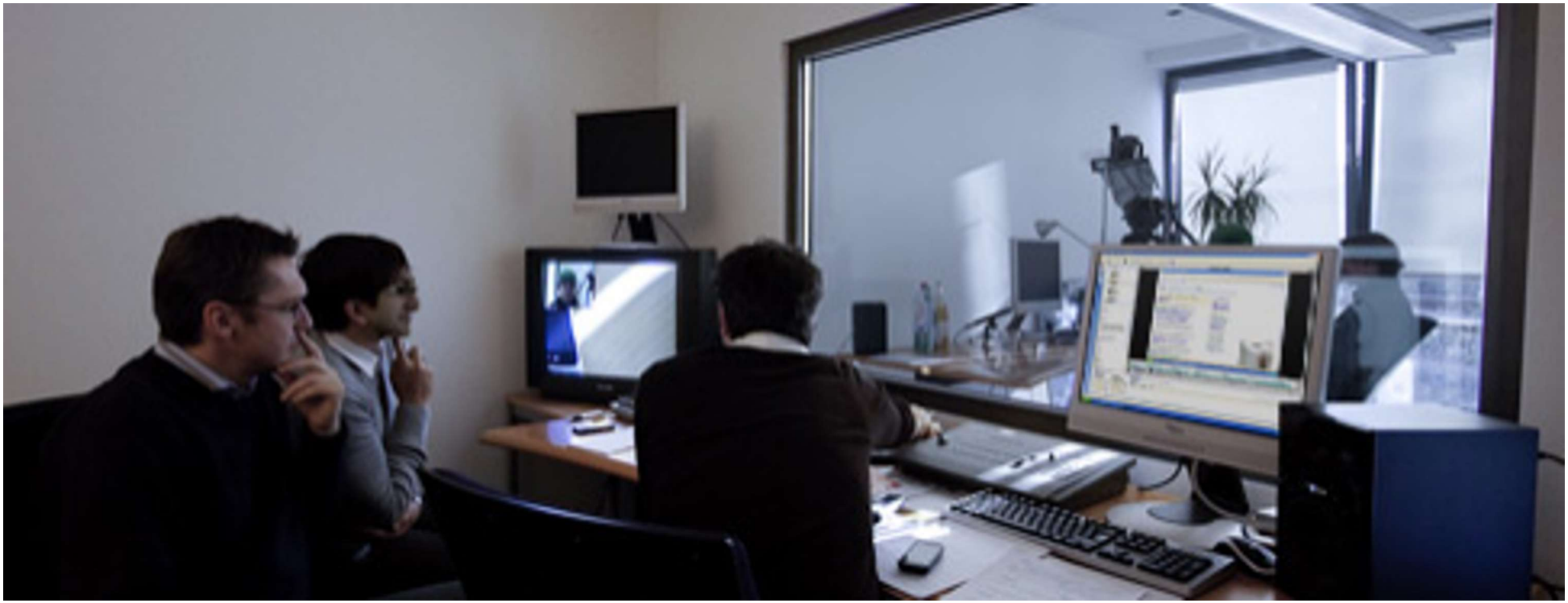
- Placed in situation similar to user
  - Take training courses
  - Read documentation
  - Take tutorials
  - Try the interface in a realistic work environment (complete with noise and distractions)
- Bird's eye view
  - Studying a full set of printed screens laid on the floor or pinned to the walls
  - Easier to evaluate topics such as consistency
- Software tools to speed analyses
  - WebTango

Usability Testing and Labs

- 1980s, testing was luxury (but deadlines crept up)
- Usability testing was incentive for deadlines
  - Sped up projects*
  - Cost savings*
- Labs are different than academia
  - Less general theory
  - More practical studies

Usability Labs

- IBM early leader
- Microsoft next (>25 labs)
- Now hundreds of companies



From http://www.ergosign.de/

Staff

- Expertise in testing (Psych, HCI, CS)
- 10 to 15 projects per year
- Meet with UI architect to plan testing (Figure 4.2)
- Participate in early task analysis and design reviews
- t – 2-6 weeks, creates study design and test plan
  - Task lists
  - Participants – number, types and sources (current customers, in company staff, temp agencies, advertisements)
- t – 1 week, pilot test (procedures, tasks, and questionnaires)
  - 1-3 participants

Participants

- Labs categorize users based on:
  - Computing background
  - Experience with task
  - Motivation
  - Education
  - Ability with the language used in the interface
- Controls for
  - Physical concerns (e.g. eyesight, handedness, age)
  - Experimental conditions (e.g. time of day, physical surroundings, noise, temperature, distractions)

Recording Participants

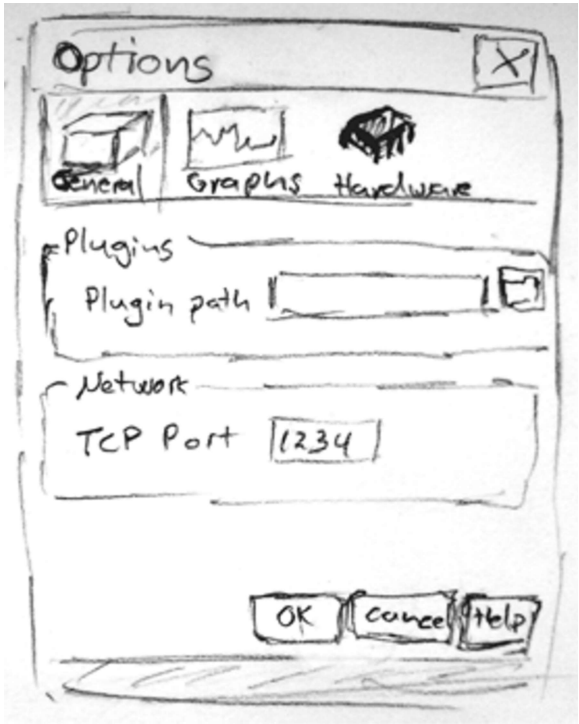
- Logging is important, yet tedious
  - Software to help (Live Logger, Morae, Spectator)
  - Powerful to see people use your interface
  - New approaches: eye tracking
    - Tobii (infrared to generate reflection patterns on corneas)

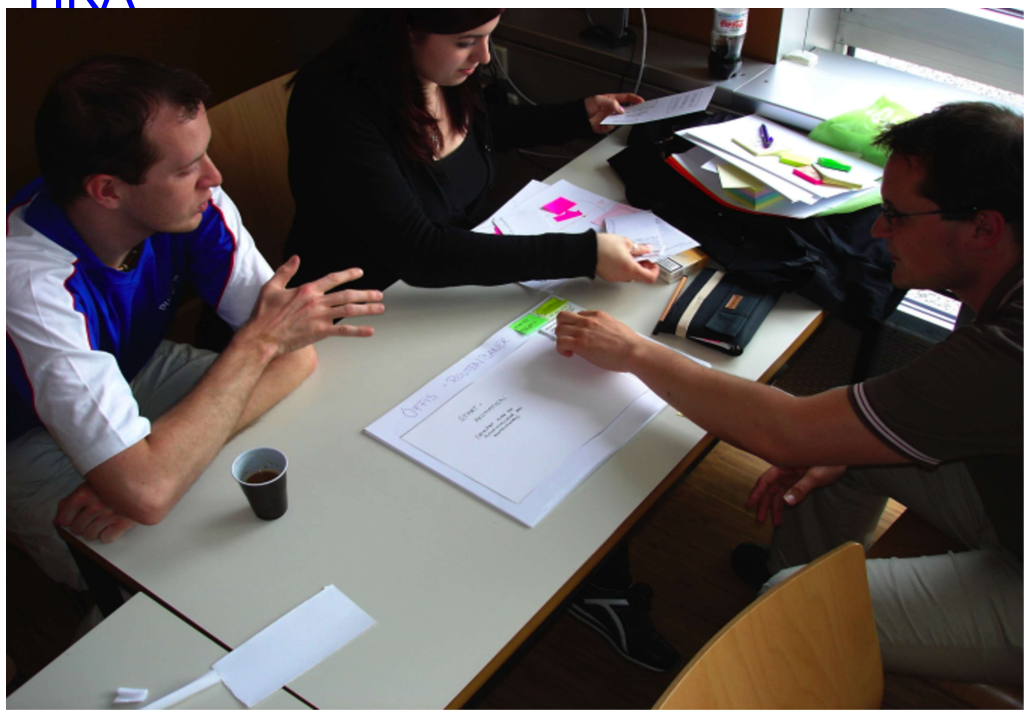
Thinking Aloud

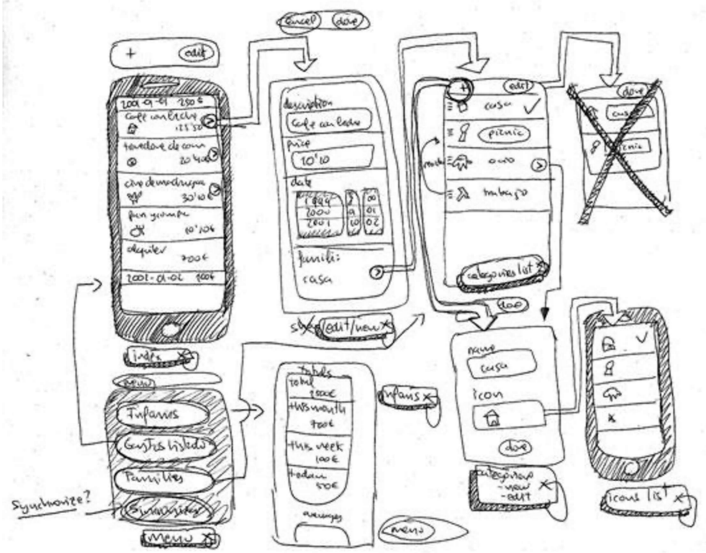
- Concurrent think aloud
  - Invite users to *think aloud*
  - Nothing they say is wrong
  - Don’t interrupt, let the user talk (sometimes need to prompt)
  - Spontaneous, encourages positive suggestions
  - Can be done in teams of participants
- Retrospective think aloud
  - Asks people afterwards what they were thinking
  - Issues with accuracy (remembering)
  - Does not interrupt users (timings are more accurate)
- https://www.youtube.com/watch?v=IHmaS0hXx4w

Types of Usability Testing

- Paper mockups and prototyping
  - Inexpensive, rapid, very productive
  - Low fidelity is sometimes better (Synder, 2003)
  - https://www.youtube.com/watch?v=9wQkLthhHKA







http://images.sixrevisions.com/2010/10/28-02_iphoneapp.jpg
http://expressionflow.com/wp-content/uploads/2007/05/paper-mock-up.png
http://user.meduni-graz.at/andreas.holzinger/holzinger/papers%20en/

Types of Usability Testing

- https://www.youtube.com/watch?v=3Qg80qTfzgU

- Discount usability testing
  - Test early and often (with 3 to 6 testers)
  - Pros: Most serious problems can be found with few testers. Good for formative evaluation (early)
  - Cons: Complex systems can't be tested this way. Not good for summative evaluation (late)

- Competitive usability testing
  - Compares new interface against prior or competitor's versions
  - Experimenter bias, be careful to not "prime the user"
  - Counterbalancing order of presentation of interfaces important
  - Within-subjects is preferred because can make comparisons

Types of Usability Testing

- Universal usability testing
  - Test with highly diverse…
    - users (experience levels, ability, etc.)
    - platforms (mac, pc, linux)
    - hardware (old (how old is old?) -> latest)
    - networks (dial-up -> broadband)
- Field tests and portable labs
  - Tests UI in realistic environments
  - Use of logging software an advantage
  - Beta tests – 10s or 1000s of users evaluate beta versions and asked to comment

Types of Usability Testing

- Remote usability testing (via web)
  - Recruited via online communities, email
  - Pros: large *n*, adds to realism
  - Difficulty in logging, validating data
  - Software can help (NetMeeting, WebEx, Sametime)
- Can You Break this Test
  - Challenge testers to break a system
  - Games, security, public displays

Limitations

- Focuses on first-time users
- Limited coverage of interface features
  - Testing only 1-3 hours – can't know performance after week or month
    - Rarely used features
- Difficult to simulate realistic conditions
  - Ex: Military, first response
    - Stress in real usage situation can lead to inability to process info (Rahman 2007)
- Yet formal studies on usability testing have identified
  - Cost savings
  - Return on investment (Sherman 2006; Bias & Mayhew 2005)

Survey Instruments

- Questionnaires
  - Paper or online (e.g. Qualtrics, SurveyMonkey)
  - Easy to grasp for many people
  - The power of many can be shown
    - 80% of the 500 users who tried the system liked Option A
    - 3 out of the 4 experts like Option B
- Success depends on
  - Clear goals in advance
  - Focused items

- Advantage of pilot testing – define and refine the survey questions!!

Designing survey questions

- Ideally
  - Based on existing questions
  - Reviewed by colleagues
  - Pilot tested with small sample
  - Have built in methods to verify respondents represent population
    - Age
    - Gender
    - Experience
    - Etc…

Likert Scales

- Most common methodology
  - Strongly Agree, Agree, Neutral, Disagree, Strongly Disagree
- Typically 5, 7, 9-point scales
- Examples
  - Improves my performance in book searching and buying
  - Enables me to search and by books faster
  - Makes it easier to search for an purchase books

Most used Likert scales

- Questionnaire for User Interaction Satisfaction

- System Usability Scale (SUS) – Brooke 1996
  - 10 questions: 5 pos worded, 5 neg worded (reverse coding)
- Post-Study System Usability Questionniare
- Computer System Usability Questionniare
- Software usability Measurement Inventory
- Website Analysis and MeasureMent Inventory
- Mobile Phone Usability Questionnaire
- Questionnaire websites
  - Gary Perlman's website, Jurek Kirakowski's website

Acceptance Tests

- Set goals for performance
  - Objective
  - Measurable
- Requirements document
  - We want the software to be “user friendly”
    - Vague and misleading criterion
- How could we rephrase it?
  - Use a metric such as Shneiderman’s goals for interface design
    - Time for users to learn specific functions
    - Speed of Task performance
    - Rate of Errors
    - User retention of commands over time
    - Subjective satisfaction

Examples: food shopping website

- Test A
  - The participants will be
    - 35 adults (25-45 years old)
    - Native speakers with no disabilities
    - Hired from an employment agency
    - Moderate web-use experience (1-5 hours/week) for at least one year
  - ≥30 of the 35 should complete the benchmark tests within 30 minutes
- Test B
  - 10 participants will be recalled after one week
  - Carry out new set of benchmark tests
  - In 20 minutes, at least 8 should be able to complete tasks correctly

Acceptance Tests

- Completing the acceptance tests
  - Can be part of contractual fulfillment
  - Demonstrate objectivity
- Different than usability tests
  - More adversarial
  - Neutral party should conduct that
- Central goal
  - To verify requirements adhered to
  - Not to detect flaws

Evaluation during active use

- Interviews with individual users
  - Pro: Get very detailed on specific concerns
  - Cons: Costly and time-consuming
- Focus group discussions
  - Pro: identify patterns of usage or hidden problems
  - Con: Certain people can dominate or sway opinion
- Targeted interviews and focus groups
  - Experienced or long-term users
  - Identify different issues than novice users

Evaluation during active use

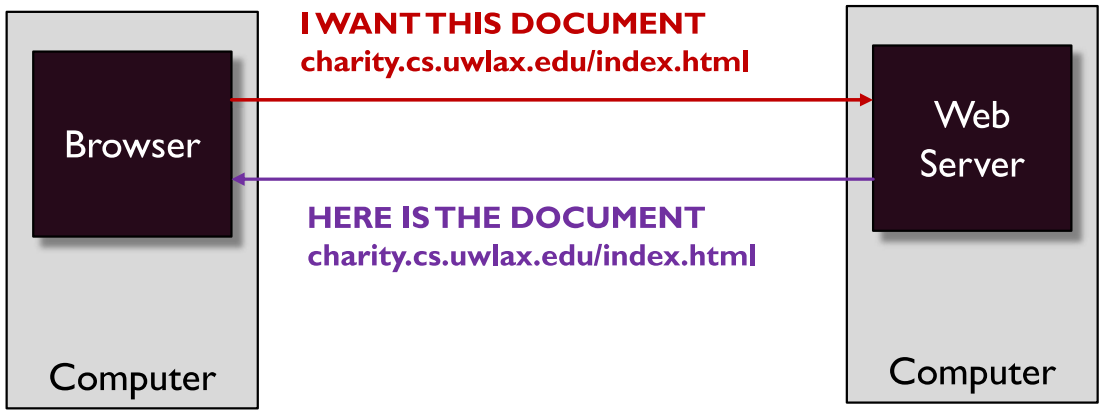
- Continuous logging
  - The system itself logs user usage
    - Video game example
  - Other examples
    - Track frequency of errors** (gives an ordered list of what to address via tutorials, training, text changes, etc.)
    - Speed of performance
    - Track which features are used and which are not
    - Web Analytics
- Ethical considerations
  - Privacy? What gets logged? Opt-in/out?
  - What about companies?

Online and Telephone Help

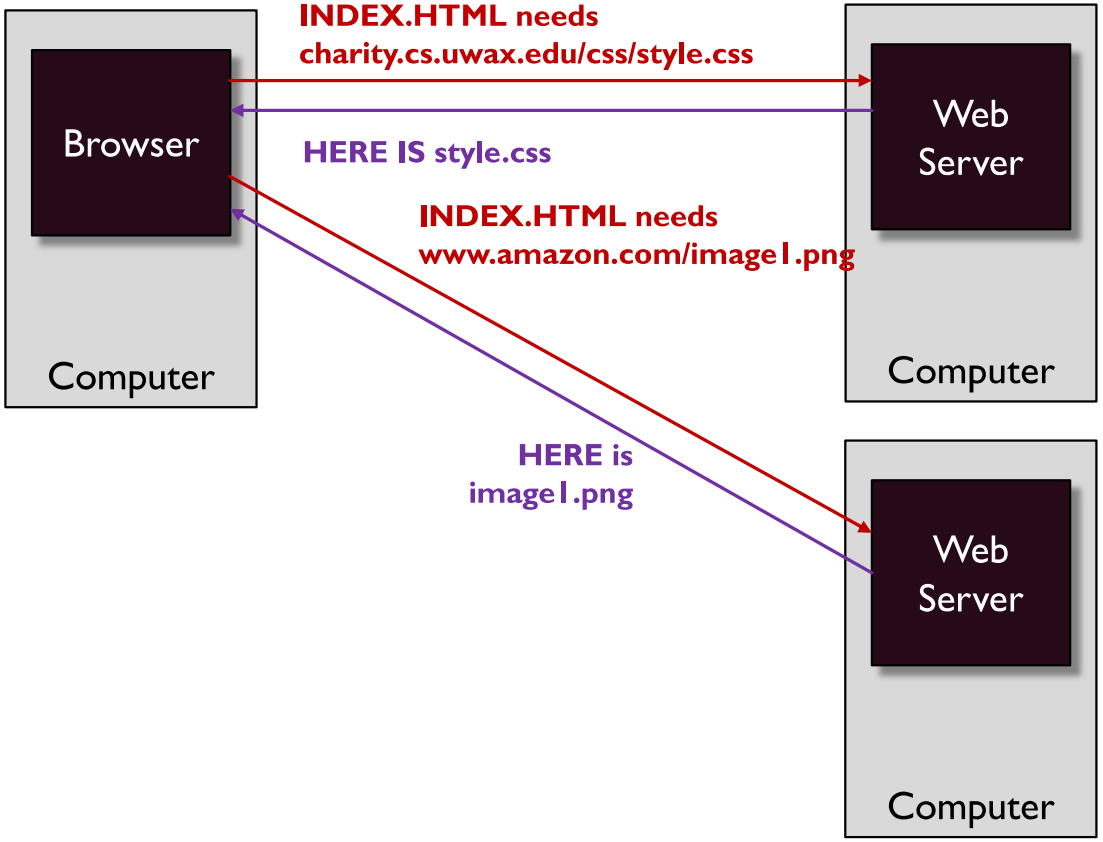
- Users feel reassured having people ready to help (real-time chat online or via telephone)
- E.g. (2008) Netflix has 8.4 million customers, how many telephone customer service reps?
  - 375
  - Expensive, but higher customer satisfaction
- Cheaper versions are bug report systems
  - Windows, Chrome, Bugzilla

BEYOND DESKTOP GUI

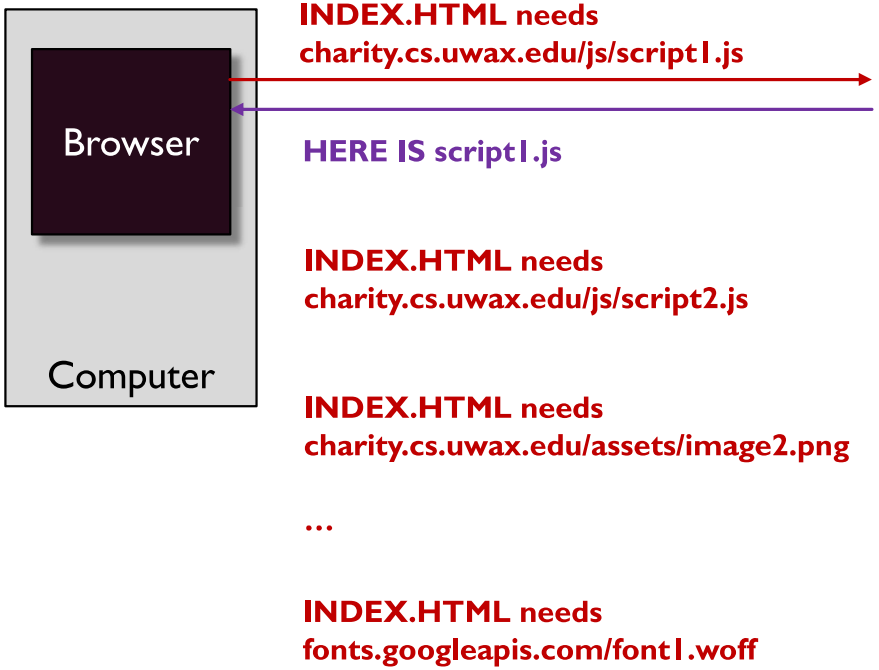
WHAT HAPPENS WHEN LOADING A WEB PAGE?



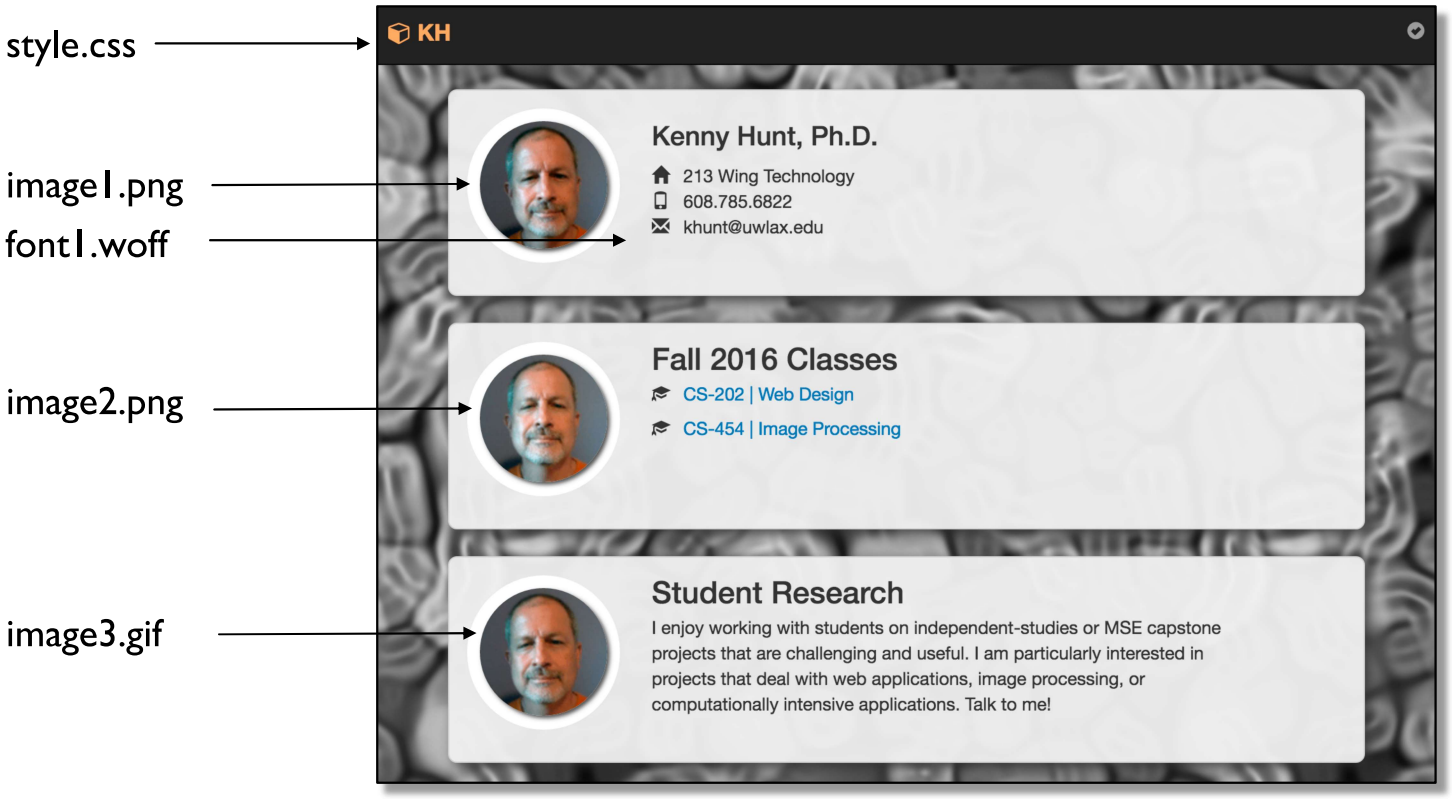
WHAT HAPPENS WHEN LOADING A WEB PAGE?



WHAT HAPPENS WHEN LOADING A WEB PAGE?



RENDER THE DOCUMENT

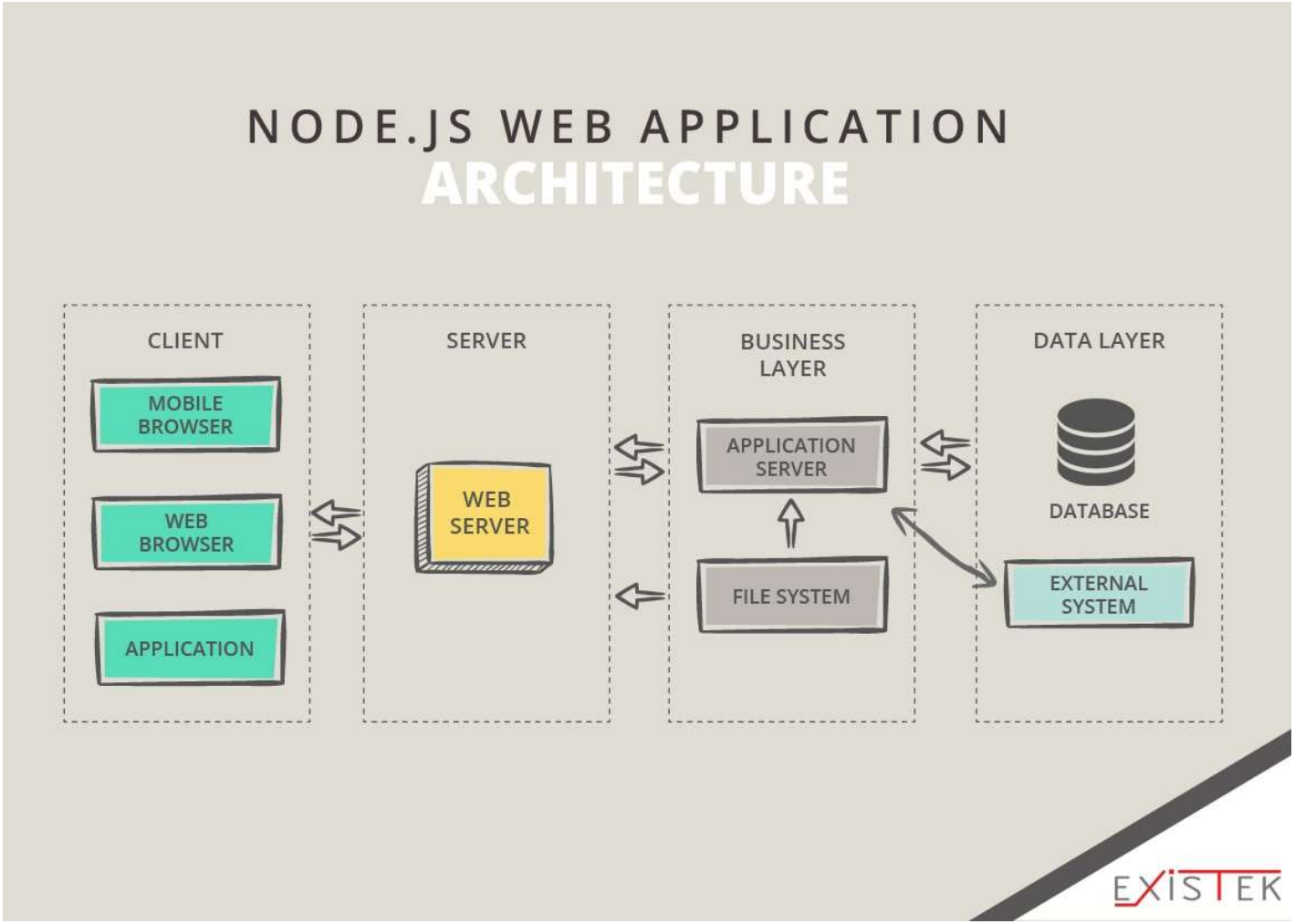


REVIEW THE PROCESS

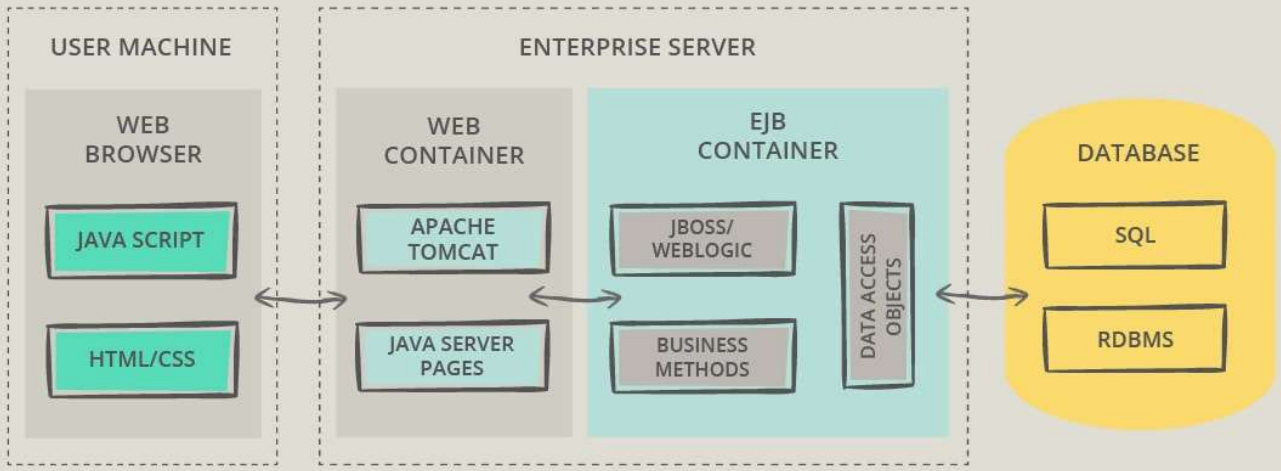
- User runs a web browser program. This is the client.
- A user enters: "***http://charity.cs.uwlax.edu/index.html***"
  - Server : named in the URL (charity.cs.uwlax.edu)
  - Resource : named in the URL (index.html)
- The browser requests the resource
  - A socket is opened using TCP/IP
  - The socket carries an HTTP request.
- The web server responds with the resource
  - The socket carries an HTTP response with the requested resource.

GUI FOR WEB APPLICATIONS

- Clients use the browser as the GUI
- A web app requires multiple languages and often duplicate coding on server/client.
  - HTML (client and server)
  - CSS (client)
  - JS (client)
  - Java (server)
- The application starts by issuing an HTTP request for an HTML document. In modern apps, the entire application UI is contained within this HTTP document.
- Subsequent requests typically load only the data used to populate the UI.

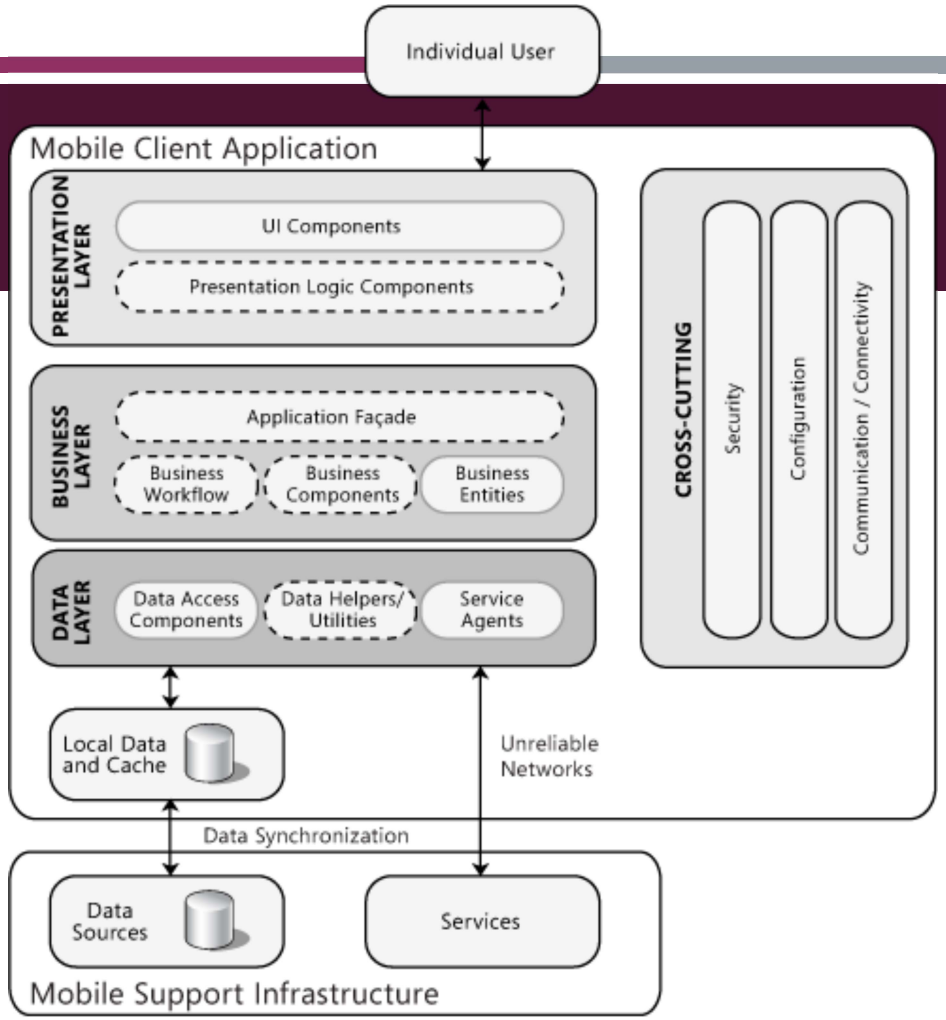

NODE.JS WEB APPLICATION
ARCHITECTURE
CLIENT
MOBILE BROWSER
WEB BROWSER
APPLICATION
SERVER
WEB SERVER
BUSINESS LAYER
APPLICATION SERVER
FILE SYSTEM
DATA LAYER
DATABASE
EXTERNAL SYSTEM
EXISTEK

J2EE WEB APPLICATION ARCHITECTURE



EXISTEK

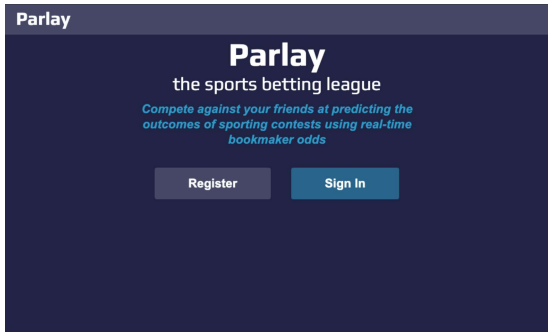
MOBILE APPLICATIONS

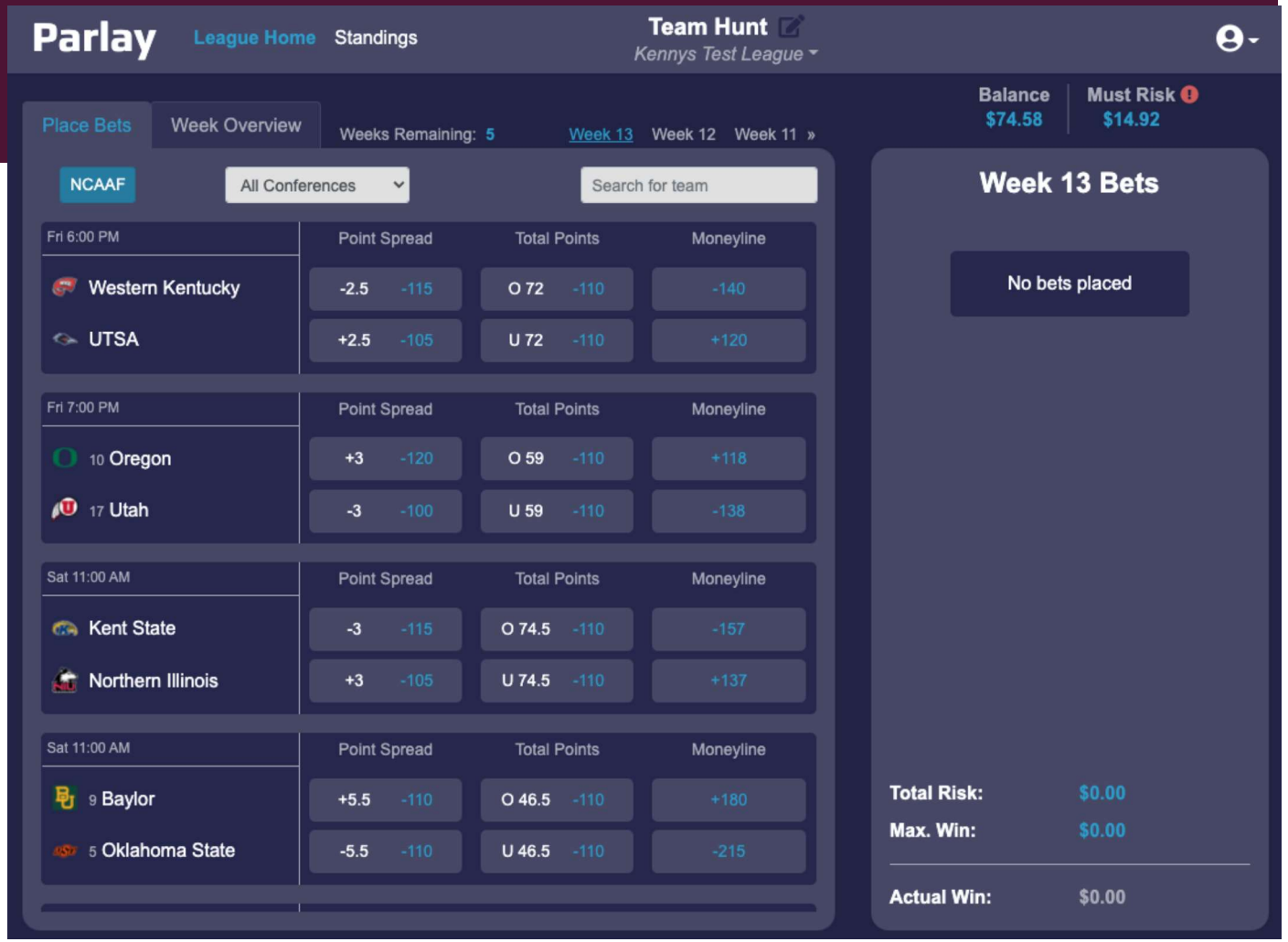


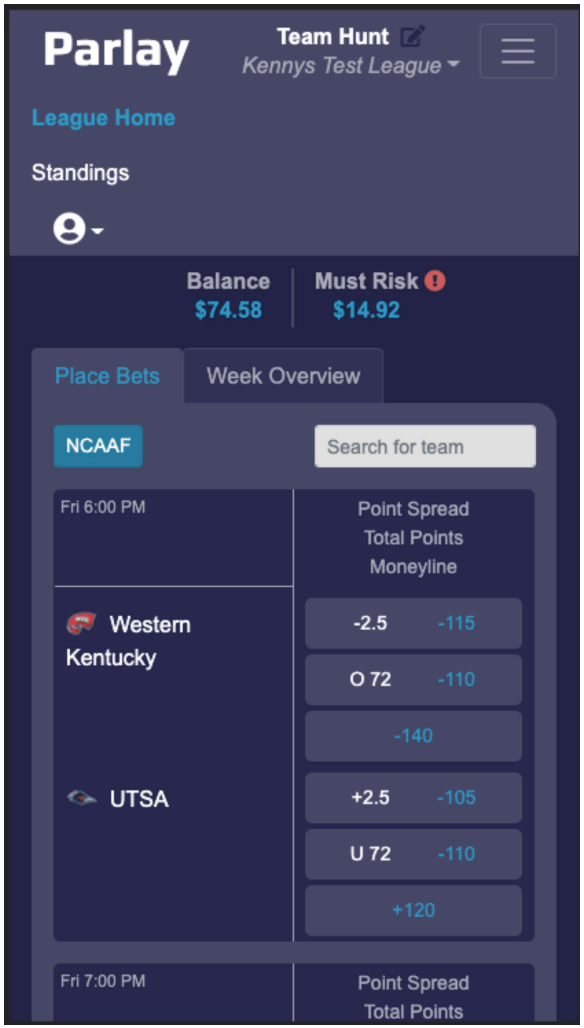
GUI FOR MOBILE APPLICATIONS

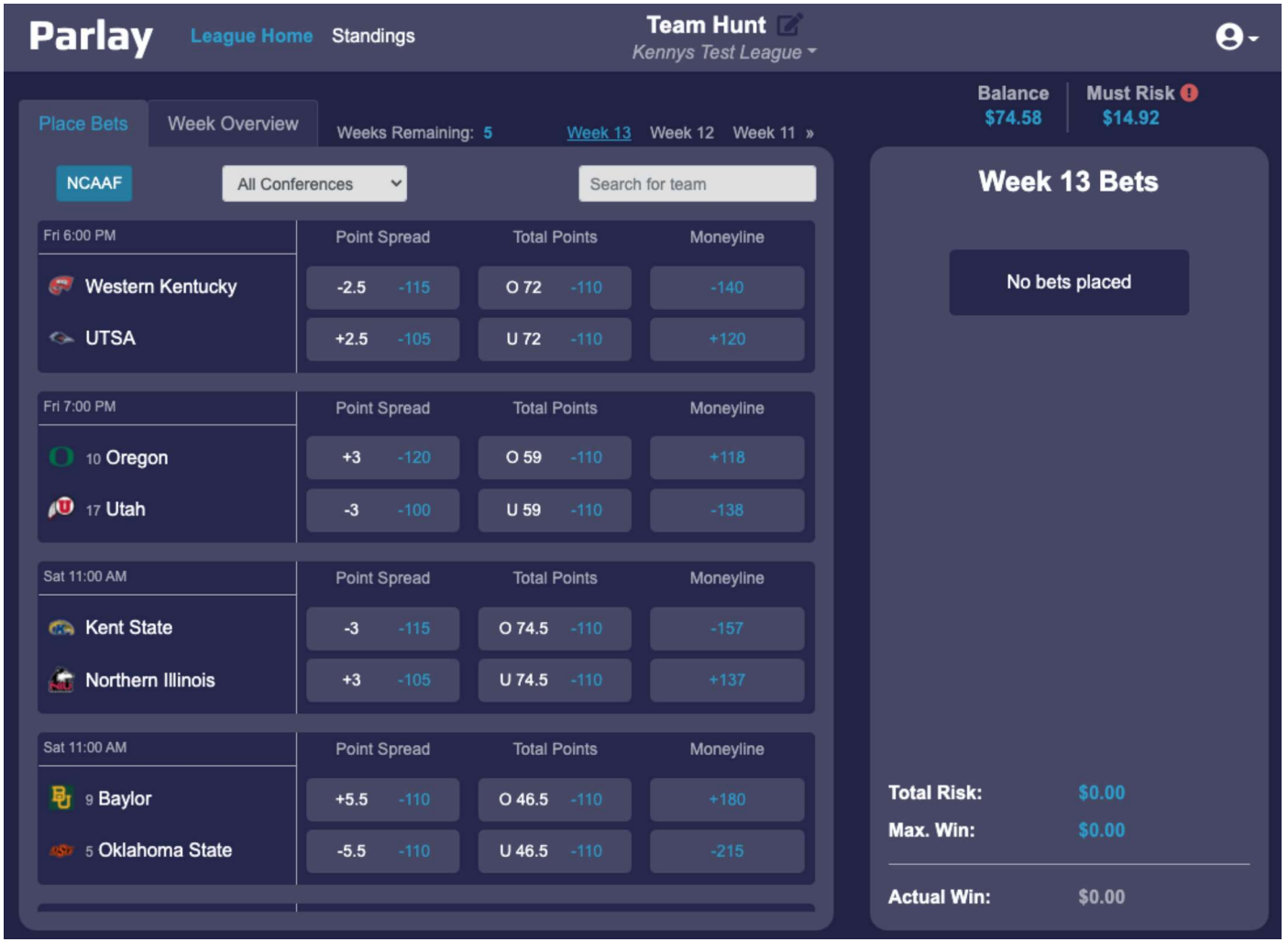
- Graphical elements must be of the right size and layout for the device. Sizes vary greatly.
- Portable on multi-platforms and multi-os
- Navigation, back-button are more important
  - A single PC screen may consume several mobile screens
  - "All on one screen" approach is almost useless for a mobile app
- Fonts/color/icon selection is critical due to size constraints
- Direct manipulation principles are expected to be used almost all the time
- Personalization
  - Requires a great deal of work for the developer
  - Must understand the application domain and the users thoroughly before designing the GUI
  - May require several iterations
- Keep it simple
  - A complex GUI for a mobile app is one of the reasons that users will tend to ignore/not-use the app

RESPONSIVE LAYOUT EXAMPLE






Parlay
Team Hunt
Kennys Test League
League Home
Standings
Balance
$74.58
Must Risk
$14.92
Place Bets
Week Overview
NCAAF
Search for team
Fri 6:00 PM
Point Spread
Total Points
Moneyline
Western Kentucky
-2.5 -115
O 72 -110
-140
UTSA
+2.5 -105
U 72 -110
+120
Fri 7:00 PM
Point Spread
Total Points


Parlay
League Home
Standings
Team Hunt
Kennys Test League
Balance
$74.58
Must Risk
$14.92
Place Bets
Week Overview
Weeks Remaining: 5
Week 13
Week 12
Week 11 »
NCAAF
All Conferences
Search for team
Fri 6:00 PM
Point Spread
Total Points
Moneyline
Western Kentucky
-2.5 -115
O 72 -110
-140
UTSA
+2.5 -105
U 72 -110
+120
Fri 7:00 PM
Point Spread
Total Points
Moneyline
10 Oregon
+3 -120
O 59 -110
+118
17 Utah
-3 -100
U 59 -110
-138
Sat 11:00 AM
Point Spread
Total Points
Moneyline
Kent State
-3 -115
O 74.5 -110
-157
Northern Illinois
+3 -105
U 74.5 -110
+137
Sat 11:00 AM
Point Spread
Total Points
Moneyline
9 Baylor
+5.5 -110
O 46.5 -110
+180
5 Oklahoma State
-5.5 -110
U 46.5 -110
-215
Week 13 Bets
No bets placed
Total Risk:
$0.00
Max. Win:
$0.00
Actual Win:
$0.00

RESPONSIVE DESIGN

- In the early days of web design, there were two options when designing a website:
  - You could create a *liquid* site, which would stretch to fill the browser window
  - or a *fixed width* site, which would be a fixed size in pixels.
- These two approaches tended to result in a website that looked its best on the screen of the person designing the site! The liquid site resulted in a squashed design on smaller screens and unreadably long line lengths on larger ones.

RESPONSIVE WEB DESIGN

- The term responsive design was coined by Ethan Marcotte in 2010 and described the use of three techniques in combination.
- The first was the idea of fluid grids (something that was done via JS in the early days)
- The second technique was the idea of fluid images. Using a very simple technique of setting the max-width property to 100%, images would scale down smaller if their containing column became narrower than the image's intrinsic size, but never grow larger. This enables an image to scale down to fit in a flexibly-sized column, rather than overflow it, but not grow larger and become pixelated if the column becomes wider than the image.
- The third key component was the media query. Media Queries enable the type of layout control that was previously done with JS, only now using CSS. Rather than having one layout for all screen sizes, the layout could be changed. Sidebars could be repositioned for the smaller screen, or alternate navigation could be displayed.

BACKGROUND

- Web apps use HTML documents that, when rendered, are the UI
  - An HTML document is content-driven. The HTML markups describe the semantics of the elements on the page
- HTML documents are styled via CSS
  - A CSS stylesheet is appearance-driven. The CSS stylesheet describes how the elements in an HTML document ***appear***.
  - CSS rules a) select elements and b) declare how the selected elements should appear on the screen.
- Media queries allow the different CSS rules to apply based on the size and shape of the media on which they are viewed. Note, that document may by "viewed" on a printer or a screen. The CSS rules for each may differ.
  - @media screen and (min-width: 800px) { …rules go here… } // apply these rules to screens at least 800px wide
  - @media screen and (color) { … rules go here…} // apply these rules to screens that support color
  - @media print { … rules go here…} // apply these rules when printing

CONSIDER CHANGING LAYOUTS AT CERTAIN POINTS

- Most CSS frameworks use breakpoints to alter layout
  - @media (0 <= width <= 320px) { … these rules apply to the smallest-size screen … }
  - @media (320 < width <= 640px) { … these rules apply to the next smallest-size screen … }
  - @media (641 < width <= 1280px) {… }

BOOTSTRAP 5.0

Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
X-Small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
Extra large	`xl`	≥1200px
Extra extra large	`xxl`	≥1400px

BOOTSTRAP 5.0

Each breakpoint was chosen to comfortably hold containers whose widths are multiples of 12. Breakpoints are also representative of a subset of common device sizes and viewport dimensions—they don't specifically target every use case or device. Instead, the ranges provide a strong and consistent foundation to build on for nearly any device.

These breakpoints are customizable via Sass—you'll find them in a Sass map in our `_variables.scss` stylesheet.

Copy

```
$grid-breakpoints: (
  xs: 0,
  sm: 576px,
  md: 768px,
  lg: 992px,
  xl: 1200px,
  xxl: 1400px
);
```

BOOTSTRAP 5.0

```
// No media query necessary for xs breakpoint as it's effectively `@media (min-width: 0)
@include media-breakpoint-up(sm) { ... }
@include media-breakpoint-up(md) { ... }
@include media-breakpoint-up(lg) { ... }
@include media-breakpoint-up(xl) { ... }
@include media-breakpoint-up(xxl) { ... }

// Usage

// Example: Hide starting at `min-width: 0`, and then show at the `sm` breakpoint
.custom-class {
  display: none;
}
@include media-breakpoint-up(sm) {
  .custom-class {
    display: block;
  }
}
```

LAYOUT WITH BREAKPOINTS

- Responsive sites are built upon breakpoints and ***flexible grids***. A flexible grid means that you don't need to target every possible device size and build a pixel perfect layout for it. Several options exist for defining a flexible grid.
  - CSS multicol technique.
    - *details on next slide*
  - Flexbox
    - *display* is a CSS property that, when set to "flex", describe a flexbox layout.
  - Grid
    - *display* is a CSS property that, when set to "grid", describes a grid layout.

MULTIPLE COLUMNS

- CSS allows containers to arrange children in columns
  - column-count: <number> = the number of columns in the container.
  - column-width: <number> = the ideal width of each column in the container. The container will have as many columns as can fit without any of them having a width less than this value. If the container is narrower than this value, the container will have a single column with less width.
  - column-span: all | none = an element can span across all columns. Used for headers or callouts.
  - column-gap: <length> = the amount of space (gutter) between columns.
  - column-rule: <width> <style> <color> = describes a "line" between columns.
    - <width> is a length value
    - <style> is one of none, dotted, dashed, solid, double, groove, ridge, inset, outset
    - <color> is a color!

RESPONSIVE IMAGES

```
<img src="imageURL" width="360px" height="auto" max-width="100vw">
```

- Consider an image that is 1024 pixels wide and 768 pixels tall.
  - A full-color image of that resolution is likely between 500-750 KB.
  - How is this image displayed on a 360x740 pixel screen?
- We can optimize both display and delivery of this image for various displays. In this example, we select to optimize for displays of widths 320, 512 and 1024 pixels.
  - The HTML img tag supports a "srcset" attribute that allows us to specify a set of images that the browser will use to select one image for display. That selection is based on screen resolution.

```
<img src="img_512.jpg"
    srcset="img_320.jpg 320w,
            img_512.jpg 512w,
            img_1024.jpg 1024w">
```

PIXEL DENSITY

- An alternative to hardcoding image size widths is to specify pixel density
  - Pixel density is given as pixels-per-inch. The number of physical pixels in one physical-inch of screen space
  - Displays support pixels that are not square (perhaps rectangular or circular)
  - The DOM specifies that window.devicePixelRatio provides the pixel density for a particular display. This value will likely be between 1 and 4. This value is computed as (PHYISCAL_PPI / 96 PPI).
- The img tag allows the designer to specify which image to display based on pixel density. The browser chooses the srcset image with the smallest pixel density that is at least as large as window.devicePixelDensity.

```
<img src="img_512.png" srcset="img_512.png 1x, img_1024.png 2x, img_2048.png 4x">
```

THE PICTURE ELEMENT

- The picture element allows the designer to specify which image must be selected based on media queries
  - Allows for multiple "source" elements. Each one specifying an image with a related media query
  - There must be one img child. No img is displayed without the img child. This is the fallback image to display.

```
<picture>
  <source srcset="img_2048.png" media="(min-width: 2048px)">
  <source srcset="img_1024.png" media="(min-width: 1024px)">
  <source srcset="img_512.png" media="(min-width: 321px)">
  <source srcset="img_320.png" media="(max-width: 320px)">
  <img src="img_512.png" alt="Expensive red car">
</picture>
```

IMAGES SHOULD BE OPTIMIZED

- When creating an image for the web, it should be optimized (compressed) for bandwidth.
  - Consider the chrome diagnostic tools
  - Consider using a tool such as tinypng.com