

QUICK RECAP

A reminder of what we have done during Lecture 06





LAST TIME...

- Client side development
 - What are connectors?
 - How to use shared state?
- RPC
 - When should RPC be used?
 - How to call methods using RPC?
- Add-ons
 - How to use custom add-ons?

BEST UX PRACTICES AND MOBILE FIRST

Development of Modern Web Applications (with Vaadin)

Lecture 07

OVERVIEW

- Best UX practices
 - <http://www.designforfounders.com/web-app-ux/>
 - Written by Heidi Pungartnik
 - To some extent also applicable to desktop apps
- Mobile First
 - A development paradigm for web apps
- Responsive design
- No shoutbox this time
 - Just kidding 😊

TIMELESS RULES FOR CREATING INTUITIVE WEB APPS

As collected by Heidi Pungartnik

<http://www.designforfounders.com/web-app-ux/>





DON'T REINVENT PATTERNS

- Do not fight user habits
 - Red means error
 - *Next step* button is on the right
- Lowers learning curve
- Principle of least astonishment
 - Windows users: what does F1 do?



GROUP RELATED ELEMENTS

- Place related things close to each other
- Use shared styles
 - `addStyleName("user-settings")`
- Users should be able to guess

LESS IS BETTER

- UI
 - Limit distractions
 - Focus on what is relevant
- Functionality
 - Do one thing well
- Example: shell tools
 - Build your own pipeline
 - Each tool does one thing really well



PLAN BEFORE DEVELOPING

- Design interaction before implementing
- Wireframes
 - Step-by-step actions
 - Focus on essentials
- Get a second pair of eyes
 - Software developers design crappy UI
- User stories
 - As a {foo} I want to {do things}



PROVIDE FEEDBACK

- Acknowledge user's action
 - Notification
 - Cursor change
- Very little effort, great benefits



UNOBTRUSIVE HELP

- Make help available
- Make users aware of that 😊



HELP USERS DECIDE

- Make next steps obvious
 - Make sure it is consistent
- User stories, again



FOCUS ON TARGET USERS

- Design for your audience
 - Assume some common knowledge
 - Removes distractions
- Focus on the functionality



WHERE AM I?

- Do not let users feel lost
 - Clearly mark the progress
 - Allow navigating back and forth
- Use Navigator
 - Make good use of navigation parameters
 - Display breadcrumbs or something similar
- State changed? Change URL

• WHERE DID THAT COME FROM?

- Show transitions between elements
 - No surprises to the user
- Example: Vaadin Notifications
- CSS is a standard and supports transitions
 - "Standard"





DESIGN FOR NO DATA

- First-time user has no data
 - Display something, instead of void
- This is a surprisingly good rule
- Only one chance to make the first impression
 - That applies not only to software development 😊



BE CONSISTENT

- UI should be about the same all the time
 - At least the main elements
- Navigator needs a view port
 - Which can be any component container
 - Which means general UI look-and-feel is not changed



LOAD QUICKLY

- Do not let users wait
 - Or at least give them some feedback
 - Loading screens can be fun, too
 - SimCity 4 Loading Messages 😊
- Progressive Web Applications
 - Kind of a related topic

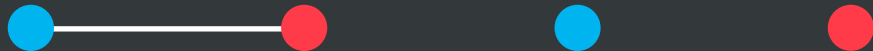


TEST

- Ideally, by someone else
- As said, developers are crappy UX designers
- You know how your software works
 - What about others?
- Vaadin TestBench
 - Part of ProTools
 - JUnit, but for UI
 - Awesome for acceptance and regression tests

MOBILE FIRST

What is it, why you need it and how you can fail it?





SHOUTBOX

- How was the UI planned?
 - Text at the top, button to the right
 - Content at the bottom
- How does it look?
 - Okayish on a monitor
 - All desktop monitors are horizontal
- Did I ever resize the browser window?
- Did I even think about it?



MOBILE WEB BROWSING

- Software developers use computers
- Other people... not that often
 - When was the last time your mother used a computer to browse the internet?
 - Assuming your mother is not a software developer 😊
- Modern phones are always connected
 - Many people do not need a computer
 - Everyone has a phone
- What is the shape of a mobile device screen?



GRACEFUL DEGRADATION

- Start with a fully-blown app
 - Optimise for desktop
 - Huge screen
 - Infinite processing power
 - Countless extra features
- Get rid of features as the display gets smaller
 - But keep the functionality as-is



PROGRESSIVE ENHANCEMENT

- Start with a fully-blown app
 - Optimise for mobile
 - Minimal screen
 - Slow CPU
 - No extras
- Add extra features as the screen gets bigger
 - Keep the functionality as-is



WHY IS IT GOOD?

- Focus
 - Minimal requirements
 - Essentials, and essentials only – but working
 - Maximum functionality
- Portable
 - Using extra screen space is not a problem
- Extendable
 - More space? More features!

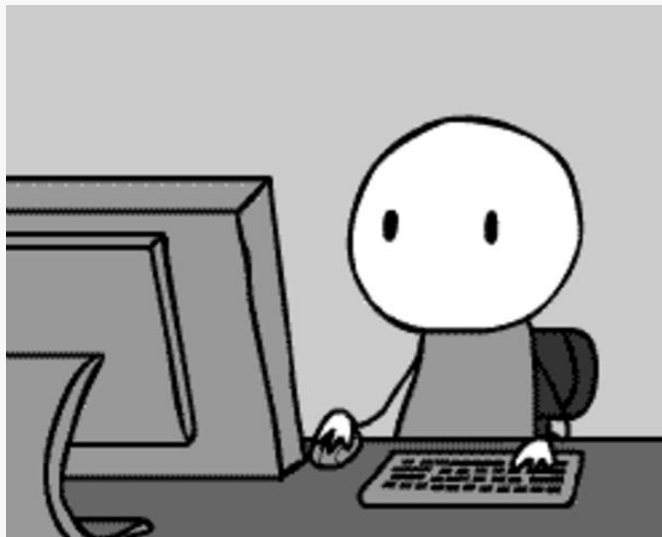


WHY IS IT BAD?

- More constraints right from the start
 - Small screen
 - No extras
 - Focused content
- Different behavioural patterns
 - More tapping, scrolling, etc.
 - Vertical instead of horizontal
 - Must work instantly

BEHAVIOURAL PATTERNS?!

Using a computer



Using a phone



• SO, IT IS NOT BAD?

- It is more difficult
- Apply it by default
 - Unless the app will never be used by a mobile user
 - Yeah, right
 - Unless you do not care for mobile users
 - ~1/4 of the internet users use mobile most of the time
- Apply it when it makes sense
 - Not everything has to be mobile-friendly
 - Not everything has to be a web app

RESPONSIVE DESIGN

Arranging content to fit different screen types





WHAT?

- Web design technique
 - Content rearranged according to available space
 - Context-aware elements
 - Image scaling
- Quite an advanced approach
- Really great results
 - Good readability, regardless of device
 - Information hierarchy



VAADIN + RESPONSIVE

- An extension to any component
 - `Responsive.makeResponsive(Component...)`
 - Most likely used on a CSS layout 😊
 - Remember to `addStyleName()`
- Enables responsive SCSS syntax
 - `width-range~="0-1000px"`
 - `height-range~="0-200px"`
 - The upper bound does not have to be present
 - The ranges may overlap
 - All matching are active
- All resources must be in the same domain

DEMO!

Shoutbox step 10

<http://github.com/vaadin-miki/shoutbox>

end branch: **step-10**





THE PLAN

- Responsive Shoutbox
 - For small screens
 - Remove room navigation
 - And some captions, too
 - Move text field and button to the bottom
 - Current view ok for big screens
- Top-level layout will be responsive

SUMMARY

What did we do today



• LESSONS OF TODAY (HOPEFULLY)

- Web app design principles
 - Why should they be applied?
- Mobile First
 - What is it?
 - What are the benefits and drawbacks?
- Responsive Design
 - What is it?
 - How to use it with Vaadin?



COMING UP NEXT

- Declarative UI / Vaadin Designer
- Web Components and Vaadin Elements
- Quality, debugging and testing
- Progressive Web Applications

THE END

SUGGESTIONS?
QUESTIONS?

miki@vaadin.com

t: @mikiolsz