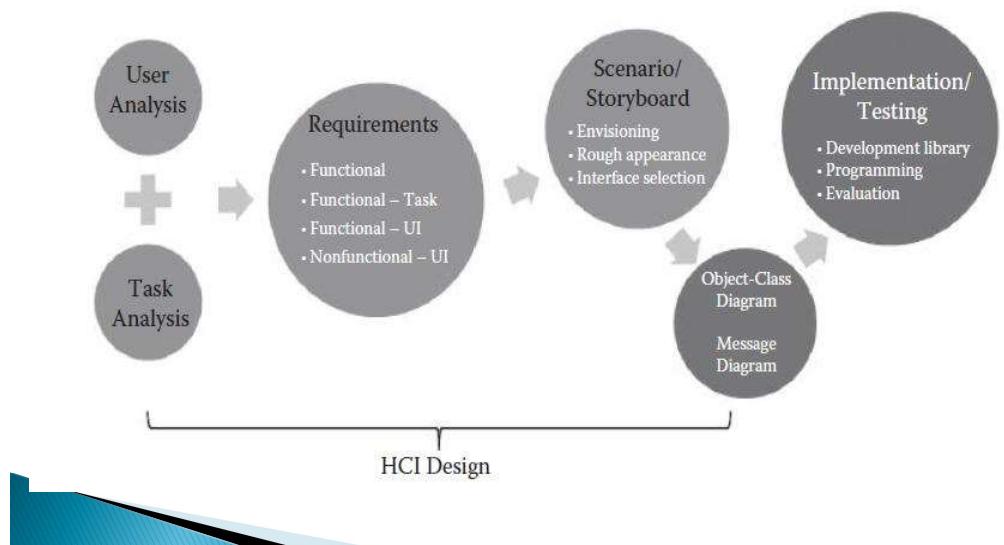


# overall iterative HCI design process

# HCI DESIGN

## Module-4

### Interface selection–Wire framing–Naïve design



## *Requirements analysis*

- ▶ Requirements analysis
- ▶ User analysis
- ▶ Scenario and task modeling
- ▶ Interface selection and consolidation

- ▶ functional-task requirements
  - User-centered focus
  - Specific actions–inputs (e.g., clicking a button, entering text) and the expected outputs (e.g., displayed information, system response)
  - Workflow considerations–sequence of actions and interactions needed to reach completion.
  - Accessibility considerations–Requirements should also address the needs of users with disabilities, ensuring the system can be navigated and used effectively by everyone.

- ▶ Examples of functional-task requirements in HCI:
  - ▶ "A user should be able to search for a product by entering keywords in the search bar and view a list of relevant results on the product page."
  - ▶ "When adding items to a shopping cart, the system should display a pop-up confirmation with the item details and the updated cart total."
  - ▶ "Users should be able to filter search results based on price range, size, and color options."
  - ▶ "The system should provide clear visual feedback to the user when an action is being processed, such as a loading indicator."

- ▶ *functional-UI requirements*
  - functions that are important in realizing certain aspects of the user experience

- ▶ Functional UI requirements (what the UI should do):
  - ▶ Users should be able to navigate between different sections of the website using a clear menu structure.
  - ▶ The search bar should provide relevant results based on user input. Error messages should be displayed in a user-friendly format when invalid data is entered.
  - ▶ Forms should have clear labels for each input field.
  - ▶ Users should be able to filter and sort data in a list view.

- ▶ Non-functional UI requirements (how the UI should perform):
  - ▶ The website should load within 3 seconds on average.
  - ▶ The UI should be accessible to users with disabilities.
  - ▶ The design should be responsive and adapt to different screen sizes.
  - ▶ The UI should be consistent across different pages and platforms.
  - ▶ The system should be able to handle a high volume of concurrent users without performance degradation.

- ▶ *User analysis:*
- ▶ The results of the user analysis will be reflected back to the requirements, and this could identify additional UI requirements (functional or non-functional). It is simply a process to reinforce the original requirements analysis to further accommodate the potential users in a more complete way.
- ▶ *Scenario and task modeling*
- Equally important to user analysis is task analysis and modeling. This is the crux of interaction modeling: identifying the application task structure and the sequential relationships between the different elements. With a crude task model, we can also start to draw a more detailed scenario or storyboard to envision how the system would be used and to assess both the appropriateness of the task model and the feasibility of the given requirements

## Scenario and stroy board

- ▶ A scenario is a narrative that describes how a user interacts with a product, while a storyboard is a visual representation of that scenario. Both are techniques used in user experience (UX) design to understand users' needs and improve the user experience.

### ▶ Scenario

A story that describes how a user interacts with a product or service

Helps identify user needs, motivations, and pain points

Outlines the purpose of the storyboard

### Storyboard

- ▶ A visual representation of a user scenario
- ▶ Shows the sequence of actions, emotions, and outcomes
- ▶ Helps visualize UX ideas
- ▶ Helps clients and stakeholders remember specific user scenarios
- ▶ Helps understand what drives user behavior

## Scenario's elements

- Setting — description of the starting state of the episode and objects that are involved
- Actors
- Goals
- Actions — things that actors do
- Events — things that happen to actors
- Objects

## Scenario types

- Problem scenarios — describe current situation features (what users can do)
- Activity scenarios — propose transformation from current practice into new design features
- Information scenarios — how users perceive, interpret and make sense of information
- Interaction scenarios — physical actions and system responses that enact and respond to the users' task goals and needs

## Creating a user scenario step-by-step

- ▶ Step 1: identify your users
- ▶ Step 2: define the context



LIGHT LEVEL



LOCATION



TIME OF DAY



SOUND LEVEL



CULTURAL INFLUENCES



DEVICE TYPE

- ▶ Step 3: focus on user goals
- ▶ Step 4: map out the steps

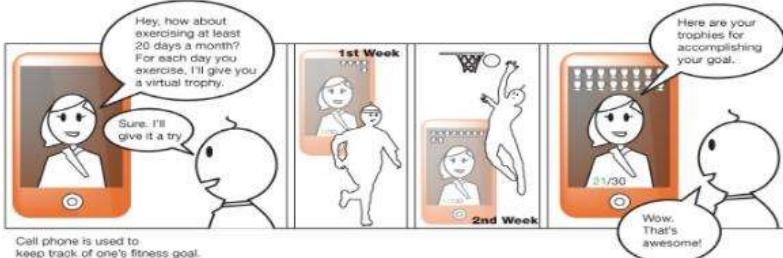
- ▶ A young professional, Sarah, is on her lunch break and wants to quickly order a salad from a food delivery app on her phone. She opens the app, searches for a nearby salad option, customizes the ingredients to remove onions, and then proceeds to checkout with her saved payment details.".

### Key points about this scenario:

- ▶ **User persona:** Sarah, a young professional.
- ▶ **Context:** Lunch break, using a mobile phone.
- ▶ **Goal:** Order a customized salad for delivery.
- ▶ **Interaction steps:** Open app, search for salad, customize ingredients, checkout with saved payment

# Storyboarding

## ■ What is it?

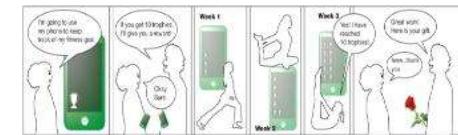


## Elements of a Storyboard

### ■ Visual storytelling

### ■ 5 visual elements

1. Level of detail
2. Inclusion of text
3. Inclusion of people & emotions
4. Number of frames
5. Portrayal of time



## How scenarios and storyboards are used

- ▶ To explore risk
- ▶ To discover new user stories
- ▶ To capture the relationship between stories
- ▶ To describe interaction steps
- ▶ To communicate ideas and solutions
- ▶ To improve the user experience

## ▶ *Interface selection and consolidation*

For each of the subtasks and scenes in the storyboard—particularly software interface components (e.g., widgets), interaction technique (e.g., voice recognition), and hardware (sensors, actuators, buttons, display, etc.)—choices will be made. The chosen individual interface components need to be consolidated into a practical package, because not all of these interface components may be available on a working platform (e.g., Android™-based smartphone, desktop PC, mp3 player).

# Interface Selection Options

- ▶ *Hardware Platforms*
- ▶ *Software Interface Components*

## *Hardware Platforms*

- ▶ *Desktop (stationary)*: Monitor (typical size: 17–42 in.; resolution:  $1280 \times 1012$  or higher); keyboard, mouse, speakers/ headphones (microphone)
  - **Suited for:** Office-related tasks, time consuming/serious tasks, multitasking
- ▶ *Smartphones/handhelds (mobile)*: LCD screen (typical size: 3.5–5 in., resolution:  $720 \times 1280$  or higher, weight  $\approx 120$  g), buttons, touch screen, speaker/headphones, microphone, camera, sensors (acceleration, tilt, light, gyro, proximity, compass, barometer), vibrators, mini “qwerty” keyboard

**Suited for:** Simple and short tasks, special-purpose tasks

- ▶ *Tablet/pads (mobile)*: LCD screen (typical size: 7–10 in., resolution:  $720 \times 1280$  or higher, weight  $\approx 700$  g), buttons, touch screen, speaker/headphones, microphone, camera, vibrators, sensors (acceleration, tilt, light, gyro, proximity, compass, barometer)
  - **Suited for:** Simple, mobile, and short tasks, but those that require a relatively large screen

- ▶ **Embedded (stationary/mobile):** LCD/LED screen (typical size: less than 3–5 in., resolution: low), buttons, special sensors, and output devices (touch screen, speaker, microphone, special sensors); embedded devices may be mobile or stationary and offer only very limited interaction for a few simple functionalities

**Suited for:** Special tasks and situations where interaction and computations are needed on the spot (e.g., printer, rice cooker, mp3 player, personal media player)

- ▶ **TV/consoles (stationary):** LCD/LED screen (typical size: >42 in., resolution: HD), button-based remote control, speaker, microphone, game controller, special sensors, peripherals (camera, wireless keyboard, Wii mote-like device, depth sensor such as Kinect)
- ▶ **Suited for:** TV-centric tasks, limited interaction, tasks that need privacy (e.g., wild-gesture-based games in the living room)
- ▶ **Kiosks/installations (stationary):** LCD screen (typical size: 10–13 in., resolution: low to medium), buttons, speaker, touchscreen, special sensors, peripherals (microphone, camera, RFID/credit-card reader, heavy-duty keyboard)
- ▶ **Suited for:** Public users and installations, limited interaction, short series of selection tasks, monitoring tasks

## Software Interface Components

- ▶ **Virtual reality (stationary):** Large-surround and high-resolution projection screen/head-mounted display/stereoscopic display, 3-D tracking sensors, 3-D sound system, haptic/tactile display, special sensors, peripherals (microphone, camera, depth sensors, glove)
- ▶ **Suited for:** Spatial training, tele-experience and tele-presence, immersive entertainment
- ▶ **Free form (stationary and mobile):** Special-purpose hardware platforms consisting of a customized configuration of individual devices best suited for a given task (when cost is not the biggest factor).

- ▶ **Windows/layers**
- ▶ **Icons**
- ▶ **Menus**
  - Selection of a menu item involves three subtasks:
    - (a) activating the menu and laying out the items (if not already activated by default),
    - (b) visually scanning and moving through the items (and scrolling if the display space is not sufficient to contain and show the whole menu of items at once), and
    - (c) choosing the wanted item. All of these subtasks are realized by making discrete inputs, e.g., by mouse click, screen touch, button push, voice command, etc.

## ► *Direct interaction*

## ► *GUI components*

- graphical user interface (GUI), also sometimes referred to as the WIMP (window, icon, mouse, and pointer)
  - *Text box*: Used for making short/medium alphanumeric input
  - *Toolbar*: A small group of frequently used icons/functions organized horizontally or vertically for a quick direct access
  - *Forms*: Mixture of menus, buttons, and text boxes for long thematic input
  - *Dialog/combo boxes*: Mixture of menus, buttons, and text boxes for short mixed-mode input

# Wireframes

## ► *3-D interface (in 2-D interaction input space)*

## ► *Other (non-WIMP) interfaces:*

- The WIMP interface is synonymous with the GUI.
- advances in interface technologies (e.g., voice recognition, language understanding, gesture recognition, 3-D tracking) and changes in the computing environment (e.g., personal to ubiquitous, sensors everywhere)—new interfaces are starting to make their way into our everyday lives. In addition, the cloud-computing environment has enabled running computationally expensive interface algorithms, which non-WIMP interfaces often require, over less powerful (e.g., mobile) devices against large service populations.

## Based on your experience

- What is a wireframe?
- What features are important?
- low fidelity vs high fidelity wireframes
- Which wireframe tools might you want to try?

# Design Tools

- ▶ **Wireframes** are basic illustrations of the structure and components of a web page. This is generally the first step in the design process
- ▶ **Mockups** generally focus on the visual design elements of the site. These are often very close or identical to the actual final site design and include all the graphics, typography, and other page elements. Mockups are generally just image files.
- ▶ **Prototypes** are semi-functional webpage layouts of a mockup/comp that serves to give a higher-fidelity preview of the actual site being built. Prototypes will have the user interface and is usually constructed using HTML/CSS . This stage precedes programming the business logic of the site.

From: <http://sixrevisions.com/user-interface/website-wireframe/>

## low fidelity vs high fidelity wireframes

### Low-Fidelity Wireframes:

- ▶ **Focus:** Basic layout, structure, and functionality.
- ▶ **Appearance:** Simple sketches, often in grayscale, with placeholders for content and imagery.
- ▶ **Purpose:** Exploring design concepts, testing user flow, and gathering early feedback.
- ▶ **Examples:** Hand-drawn sketches, simple digital mockups with basic shape

## What is a wireframe?

- ▶ Low-fidelity visual representation, may be called a skeleton or blueprint
  - Some debate how polished wireframe should be
- ▶ Includes basic page layout
- ▶ Includes navigation
- ▶ May include:
  - headers, footers, content areas, sidebars
  - dynamic widgets, search box, graphics, links



## High-Fidelity Wireframes:

- ▶ **Focus:**  
Detailed visual representation, interactive elements, and specific UI components.
- ▶ **Appearance:**  
Polished designs with real content, colors, fonts, and interactive elements.
- ▶ **Purpose:**  
Communicating detailed design decisions, aligning team expectations, and testing user interaction.
- ▶ **Examples:**  
Detailed digital mockups with specific UI components, interactive elements, and realistic content.

### When Should you use Low-Fidelity Wireframes?



### What are High Fidelity Wireframes?



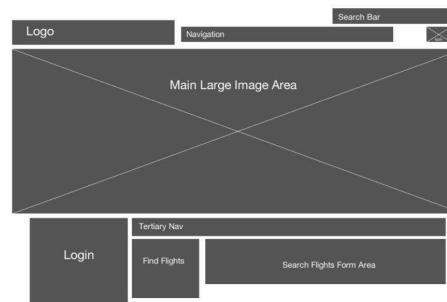
### When Should You Use High-Fidelity Wireframes?



# low fidelity vs high fidelity wireframes

Feature	High Fidelity Wireframe	Low Fidelity Wireframe
Detail Level	Rich in detail and realism	Basic, minimal details
Visuals	Detailed, often digital	Simple, often hand-drawn
Functionality	Possible interactive elements	Limited or no interactivity
Purpose	Testing and presentation	Conceptualizing and ideation
Feedback	Detailed and realistic feedback	Quick and low-cost feedback
Time Required	Time-intensive	Quick to create
Cost	Higher cost	Lower cost

#1 - They force you and your client to think about the basic interactions your user will have with your product



## Wireframes: Functionality, not visual design

Wireframes are blueprints for a site or app

Wireframes show:

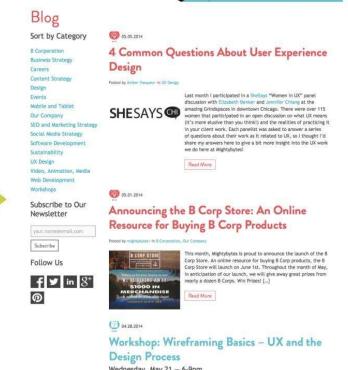
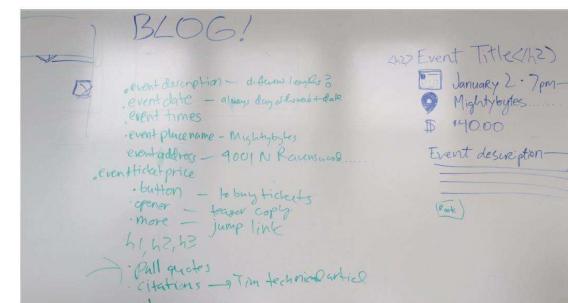
- Page layout
- Information priority
- A bridge between ideas and prototype

!

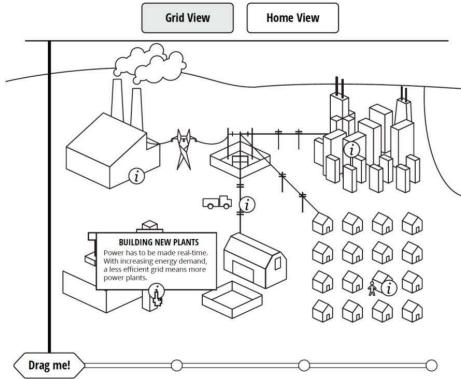
Wireframes do not show:

- Look and Feel
- Treatment of buttons and/or graphic elements
- Exact copy or verbiage

#2 - Easier to design with a wireframe in hand

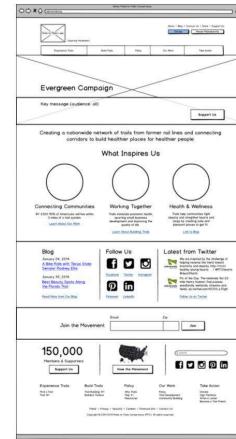


## #3 - You will save time by getting client sign off on wireframes before the design stage.



VS.

## #4 - Once there is sign off on wireframes the developers can start - without waiting for your design



```
<!DOCTYPE html>
<!-- If IE --> <html class="no-js lt-ie9 lt-ie8 lt-ie7"> </!endif-->
<!-- If IE 7 --> <html class="no-js lt-ie9 lt-ie8 lt-ie7"> </!endif-->
<!-- If IE 8 --> <html class="no-js lt-ie9 lt-ie8"> </!endif-->
<!-- If gt IE 8 --> <html class="no-js"> </!endif-->
<head charset="utf-8">
  <title>Rails to Trails Conservancy</title>
  <meta name="description" content="Rails to Trails Conservancy is a nationwide network of trails from former rail lines and connecting communities to the outdoors through rail-trail projects."/>
  <meta name="viewport" content="width=device-width,initial-scale=1.0">
  <!-- Place favicon.ico and apple-touch-icon.png in the root directory -->
  <link rel="stylesheet" href="css/style.css"/>
  <link href="http://fonts.googleapis.com/css?family=Open+Sans:400,700" rel="stylesheet" type="text/css"/>
  <script src="js/modernizr.custom.32782.js"></script>
  <script src="js/respond.min.js"></script>
  <script src="js/ios-orientationchange-fix.js"></script>
</head>
<body>
  <div id="site-main" class="skip"><a href="#main">Skip to main content</a>
```

## #5 - You can workout new ideas about user interaction you might have before you spend time in the design phase — without being distracted by design components

**Resource Library - Benefits of Trails**

Search Collection

Keywords  Search

Acquiring Rail Corridors: A How To Manual  
Published on 04/28/01 by RTC  
Collection: Trail Planning | Tag: Accessibility  
[View Details](#) [Download](#)

Trail User Survey Workbook: Sample Surveys and Methods, 2005  
Published on 02/01/04 by RTC  
Collection: Trail Planning Projects | Tag: Corridor research, Design  
[View Details](#) [Download](#)

Secrets of Successful Rail-Trails: An Acquisition and Organizing Guide  
Published on 04/18/01 by RTC  
Collection: Trail Planning | Tag: Environment  
[View Details](#) [Download](#)

Reconnecting Communities: Atlanta Rail Corridors Assessment  
Published on 04/28/04 by RTC  
Collection: Trail Planning Projects | Tag: Liability/Insurance, Outreach  
[View Details](#) [Download](#)

Page 1 Page 2 Page 3 Page 4

**Resource Library - Benefits of Trails**

Search Collection

Keywords  Search

Filter Results

Resource Category: All Resources Published by RTC Published by all others

Resource Type: All Types Case Study Fact Sheet Manual Plan Report Webinar

State: All States Tag: All Tags

Acquiring Rail Corridors: A How To Manual  
Published on 04/28/01 by RTC  
Collection: Trail Planning | Tag: Manual  
[View Details](#) [Download](#)

Trail User Survey Workbook: Sample Surveys and Methods, 2005  
Published on 02/01/04 by RTC  
Collection: Trail Planning Projects | Tag: Manual, Report  
[View Details](#) [Download](#)

Secrets of Successful Rail-Trails: An Acquisition and Organizing Guide  
Published on 04/18/01 by RTC  
Collection: Trail Planning | Tag: Manual  
[View Details](#) [Download](#)

Reconnecting Communities: Atlanta Rail Corridors Assessment  
Published on 04/28/04 by RTC  
Collection: Trail Planning Projects | Tag: Manual, Report  
[View Details](#) [Download](#)

Page 1 Page 2 Page 3 Page 4

Version #1

Version #2

## An amazing analogy!

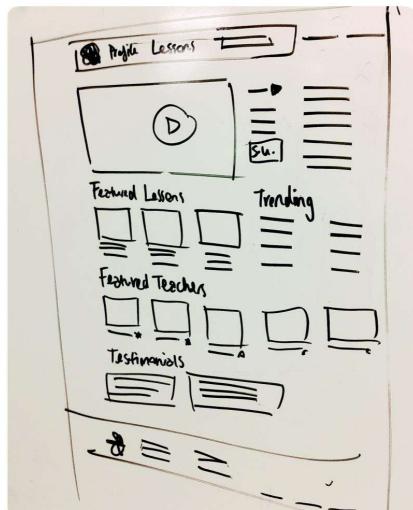
Think of your website or app like a house:

- Wireframes are the architectural blueprint of the house
- HTML/CSS are the bones of the house - the frame, the doors, the roof
- Backend Development (i.e. a Drupal install, Wordpress setup, ect.) is the plumbing and heating - you don't see it, but it's an integral part of how the house operates
- Visual Design is how you decorate the house - it's curtains, carpet, paint or wallpaper & furniture

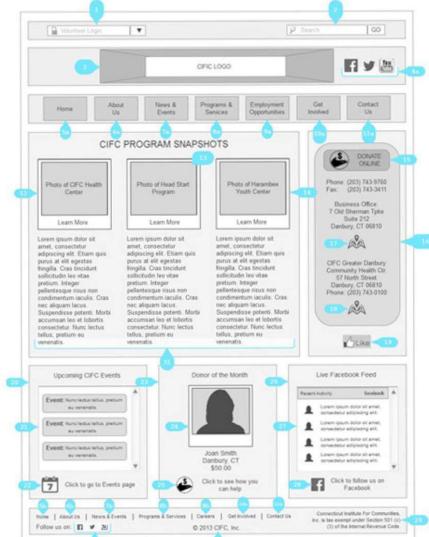
# Your first wireframe - Let's get started!

a presentation  
by

Diferent strokes for diferent folks.



Low Fidelity



High Fidelity

## What kind of wireframe do YOU need?

Some questions to ask before you begin:

- What is the budget?
- What is the timeframe?
- What do you need to get sign off to move forward in the process with your client?

## What program should you use?

Pick your poison:

- Paper sketches
- Illustrator
- Photoshop
- Keynote or Powerpoint
- InDesign
- Fireworks
- Balsamiq
- Omnigraf e

UX SKETCHBOOK



## Diferent projects require diferent workflow

Sketch > Code

Sketch > Wireframe > Visual Design > Code

Wireframe > Hi-Def Wireframe > Visual Design > Code

Wireframe > Interactive Prototype > Visual Design > Code

Sketch > Wireframe > Hi-Def Wireframe > Visual Design > Code

# Let's build a wireframe!

## Different projects require different workflow:

Sketch > Code

Sketch > Wireframe > Visual Design > Code

Wireframe > Hi-Def Wireframe > Visual Design > Code

Wireframe > Interactive Prototype > Visual Design > Code

Sketch > Wireframe > Hi-Def Wireframe > Visual Design > Code

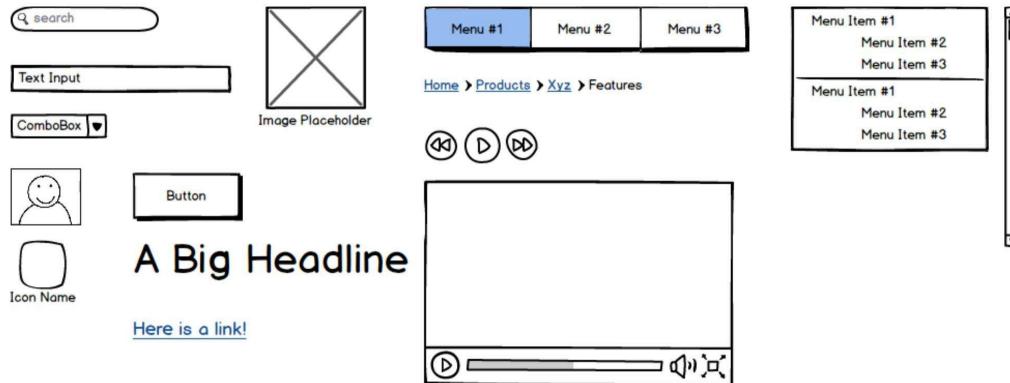
## The basics - the frame of the house

HEADER  
(first thing you want users to see)

BODY  
(followed by the content)

FOOTER  
(typically the less important stuff)

# Symbol libraries are your friend



# Wireframing in the Wild

a presentation  
by

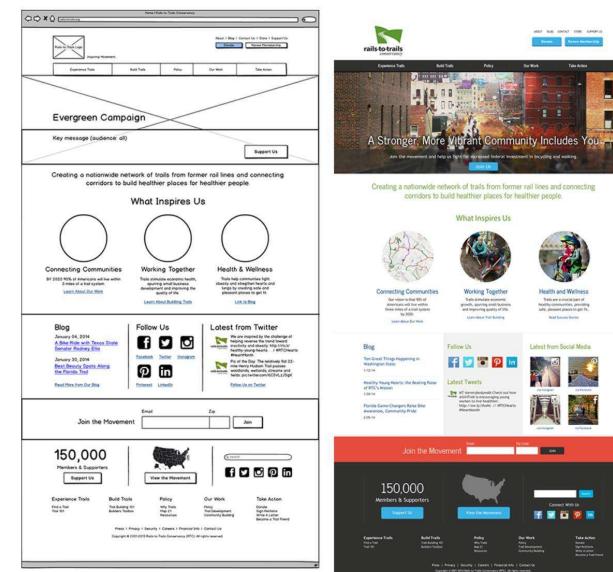
# Wireframe a site for Narwhal Adoption

It needs to accomplish 3 things:

- Search for local Narwhals
- Buy Narwhal merchandise
- Adopt a Narwhal



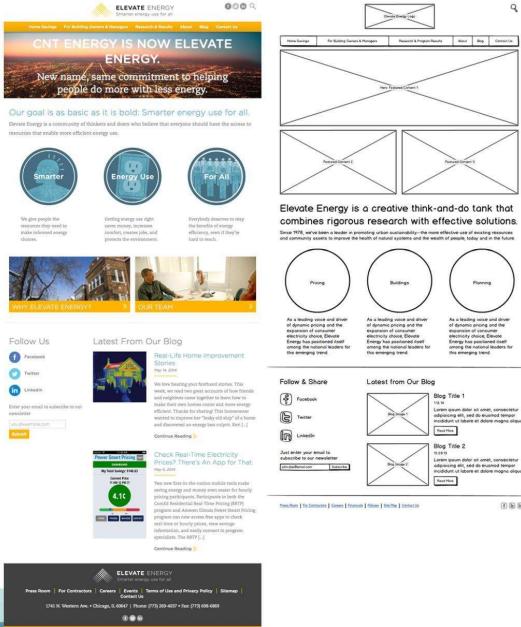
## Wireframes in the wild: Rails-to-Trails



This site is a great example of very little change from the wireframes to the design phase.  
!

We spent 70% of our time wireframing, 30% designing.

# Wireframes in the wild: Elevate Energy



This site is a great example of some changes from the wireframes to the design phase.

!

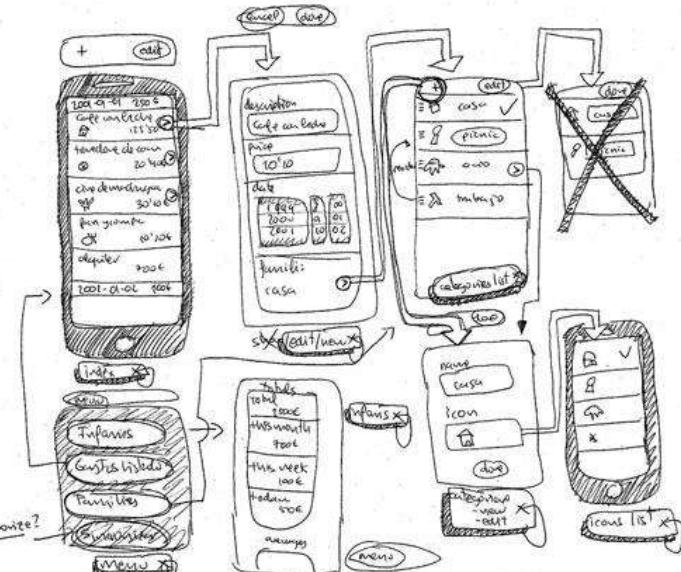
Entering in more specific content in the design & coding phase made the client realize they wanted some content to shift.

# Reasons to Use a Wireframe

- ▶ **Wireframes Make Design Changes More Efficient**
  - Client can view overall layout
- ▶ **Wireframes Make Site Navigation Designs Better**
  - Can test and refine without having to reprogram
- ▶ **Wireframes Can Improve Content**
  - No large blocks of undifferentiated text
- ▶ **Wireframes Can Improve User Interface Copy**
  - What label should be used for the call to action button?
- ▶ **Wireframes give web developers a clear path**

From: <http://sixrevisions.com/user-interface/wireframing>

## Rough Sketch



## PowerMockup

- ▶ As PowerMockup relies on features that only the Windows version of Microsoft Office provides, it doesn't work with Office for Mac. If you want to give PowerMockup a try, you can download a free trial version from the product website at [www.powermockup.com](http://www.powermockup.com).
- ▶ <http://usabilitygeek.com/wireframing-storyboarding-powerpoint-powermockup/>

# Some resources

- ▶ <http://sixrevisions.com/user-interface/wireframing-benefits/>
- ▶ <http://blog.teamtreehouse.com/20-steps-to-better-wireframing>
- ▶ <http://sixrevisions.com/user-interface/website-wireframing/>

## “Naïve” Design Example: No Sheets 1.0

HCI design process more concretely, we will go through the design of a simple interactive smartphone (Android) application, called *No Sheets*.

The main purpose of this application is to use the smartphone to present sheet music,\* thereby eliminating the need to handle paper sheet music



Score

E	E	B7	E7	A
E	B7	A	B7	A
E	B7	E		A
E	B7	A	B7	#

Chords



### ► *1. Requirements Analysis* Initial Requirements for No Sheets

1. Use the smartphone to present transcribed music like “sheet music.” Transcription includes only those for basic accompaniment like the chord information (key and type such as C# dom7), beat information (e.g., second beat in the measure).
2. Eliminate the need to carry and manage physical sheet music. Store music transcription files using a simple file format.
3. Help the user effectively accompany the music by timed and effective presentation of musical information (e.g., paced according to a preset tempo).
4. Help the user effectively practice the accompaniment and sing along through flexible control (e.g., forward, review, home buttons).
5. Help user sing along by showing the lyrics and beats in a timed fashion.

## ► 2. User Analysis

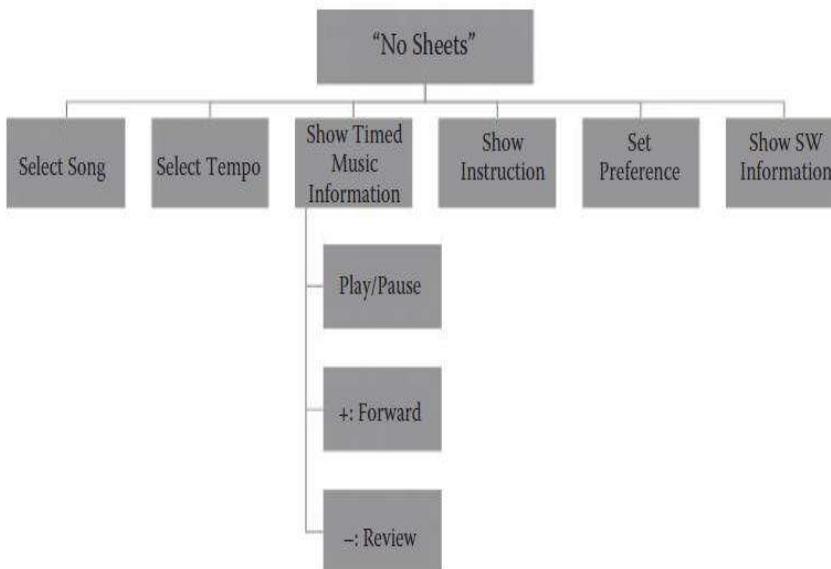
- ▶ A very minimal user analysis (that of the developer himself) resulted in (naïve first trial) interface requirements as shown

## User Interface Requirements from a Very Minimal User Analysis

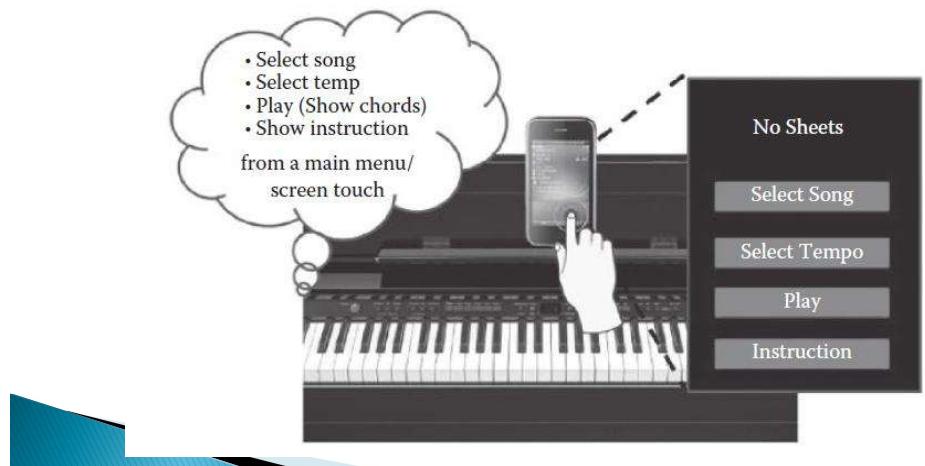
Display mode	Portrait
Layout	Top: Song title Middle: Chord – Beat – Lyrics Bottom: Control buttons
Paging	Left to right Current chord/music info in the left Next chord/music info in the right
Colors	Current chord: Yellow with blue background Next chord: Reversed Buttons: Red Background: Black



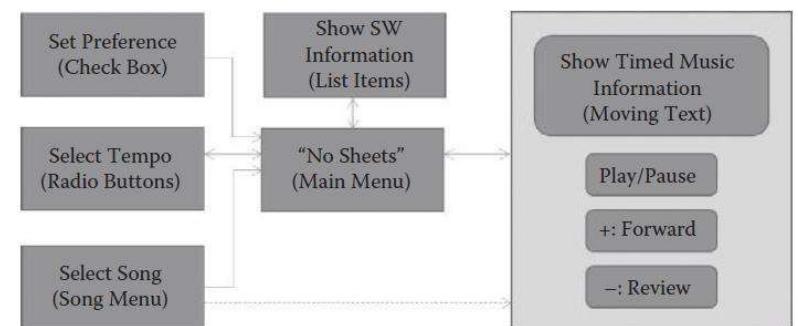
# simple task model for No Sheets



### ► 3. *Making a Scenario and Task Modeling*



# state-transition diagram for No Sheets



# *Interface Selection and Consolidation*

SUBTASK	INTERFACE DESIGN CHOICE	JUSTIFICATION
Invoking main functions	<ul style="list-style-type: none"><li>• Touch menu</li><li>• Menu items in red</li></ul>	<ul style="list-style-type: none"><li>• Familiar interface</li><li>• Catch attention</li></ul>
Selecting/changing song	<ul style="list-style-type: none"><li>• Scrolling menu</li><li>• Return to main menu upon selection</li></ul>	<ul style="list-style-type: none"><li>• There may be many songs</li></ul>
Selecting/changing tempo	<ul style="list-style-type: none"><li>• Scrolling radio buttons</li><li>• Return to main menu by OK button</li></ul>	<ul style="list-style-type: none"><li>• Only one tempo is chosen at a given time</li></ul>
Showing instruction	<ul style="list-style-type: none"><li>• Show a one page/screen image with condensed instructional content</li></ul>	<ul style="list-style-type: none"><li>• Present condensed content</li></ul>
Playing/pause (view)	<ul style="list-style-type: none"><li>• Show progress bar on top</li><li>• Control interface in the bottom</li><li>• Provide sound beeps and vibration for first and second beat</li><li>• Color-code different types of information</li></ul>	<ul style="list-style-type: none"><li>• Show status</li><li>• Familiar interface</li><li>• Use multimodal feedback for redundancy</li></ul>

Initial design wireframe for No Sheets 1.0 using a wire-framing tool

