CHALMERS |  UNIVERSITY OF GOTHENBURG

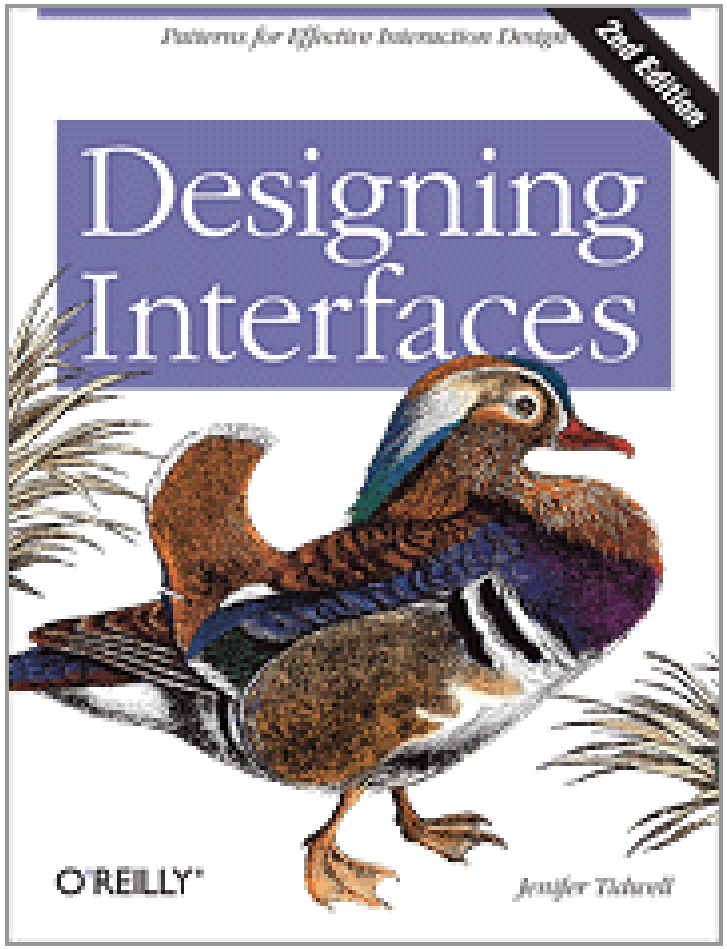
DIT045 H17 Requirements and User Experience

UX Design Patterns part 1

Jennifer Horkoff
Email: jenho@chalmers.se

UX Design Patterns

- Toolbox for interface designers



UI Resource Evolution

- Power of interfaces has increased dramatically:
  - the Java toolkits, HTML/CSS, JavaScript, Flash, and numerous open source options
  - Apple's and Microsoft's native UI toolkits are richer and nicer-looking than they used to be
  - Display technology is better
  - Web applications often look as professionally designed as the websites they're embedded in
    - some of those web sensibilities have migrated back into desktop applications in the form of blue underlined links, Back/Next buttons, beautiful fonts and background images, and non-gray color schemes
- But... just because the technical power is there, does not make it easy to design a good interface!
- Users have a much higher expectation than in the past

Interface Idioms

- Combined in new and complex ways

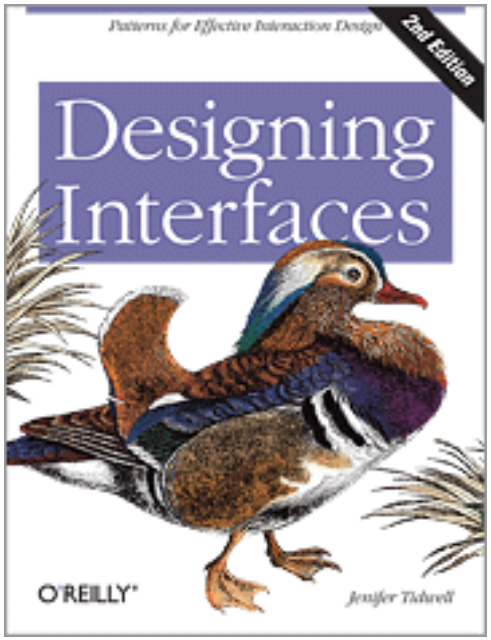
Figure P-1. *A sampler of interface idioms*

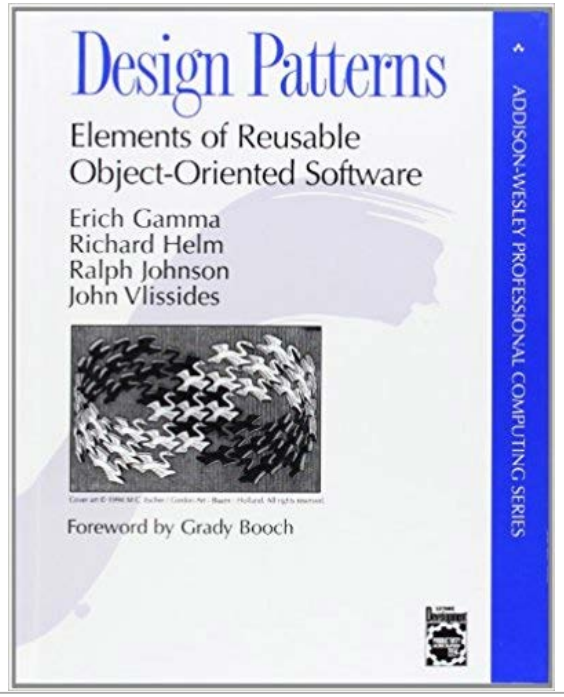
Patterns

- “patterns are structural and behavioral features that improve the “habitability” of something—a user interface, a website, an object-oriented program, or a building.”
- Patterns can be a description of best practices within a given design domain
- They aren’t off-the-shelf components
- Each implementation of a pattern differs a little from every other
- They aren’t simple rules or heuristics
- They won’t walk you through an entire set of design decision

UX Patterns vs. Design Patterns (code)

- Design patterns are established ways to structure your code design (usually Object-Oriented)
  - Iterator, abstract factory, object pool, decorator, ...
- UX patterns are somewhat similar but no code
  - Established conventions for UX appearance and behavior





Characteristics of UX Design Patterns

- *Concrete, not general*
  - Not good design principles, like “Prevent errors,” “Create a strong visual hierarchy,” and “Don’t make the user think.”
  - Patterns are concrete enough to help fill the space between high-level general principles and the low-level “grammar” of user interface design
- *Valid across different platforms and systems*
  - The best patterns aren’t specific to a single platform or idiom
- *Products, not processes*
  - Unlike heuristics or user-centered design techniques, which usually advise on how to go about *finding* a solution to an engineering or design problem, patterns *are* possible solutions.
- *Suggestions, not requirements*
  - patterns are intended to be only suggestions; you can follow them or reject them, depending on your design context and user needs

Ways to use UX Patterns

- *Learning*
- *Examples*
- *Terminology*
- *Comparison of design alternatives*
- *Inspiration*

Usage Scenarios

- Desktop applications
- Websites
- Web applications or “rich internet applications” (RIAs)
- Software for mobile devices or other consumer electronics
- Turnkey systems like kiosks
- Operating systems

Book Outline

- Organizing the Content: Information Architecture and Application Structure
- Getting Around: Navigation, Signposts, and Wayfinding
- Organizing the Page: Layout of Page Elements
- Lists of Things
- Doing Things: Actions and Commands
- Showing Complex Data: Trees, Charts, and Other Information Graphics
- Getting Input from Users: Forms and Controls
- Using Social Media
- Going Mobile
- Making It Look Good: Visual Style and Aesthetics

Won't go through all patterns!

This would take a long time.

Will pick a few examples for each section for illustration.

The rest you should look at on your own and consider using in your assignments.

Organizing the Content: Information Architecture and Application Structure

- At this point you know your requirements
- Have picked a platform, or combination (web, desktop, mobile, etc.)
- Can sketch... but be careful not to lock your thinking into a particular design
- It can be helpful to think about an application in terms of its underlying data and tasks.
  - What objects are being shown to the users? How are they categorized and ordered?
  - What do users need to do with them?
  - How many ways can you design a presentation of those things and tasks?
- *Information architecture* (IA) is the art of organizing an information space.
  - presenting, searching, browsing, labeling, categorizing, sorting, manipulating, and strategically hiding information
  - Start here for new products

The Big Picture

- Home page: via text and imagery, users should be directed to the part of your site or app that accomplishes *their* purposes
- At this level, you'll make decisions about the whole package. What interaction model will it use?
  - The desktop metaphor?
  - The simpler model of a traditional website?
  - Or a richly interactive site that splits the difference?
  - Is it a self-contained device such as a mobile phone or digital video recorder, for which you must design the interactions from scratch?
- Any such page within this design will primarily do one of these things:
  - 1. Show one single thing, such as a map, book, video, or game
  - 2. Show a list or set of things
  - 3. Provide tools to create a thing
  - 4. Facilitate a task
  - Or some combination of the above

Information Patterns

1. Feature, Search, and Browse
2. News Stream
3. Picture Manager
4. Dashboard
5. Canvas Plus Palette
6. Wizard
7. Settings Editor
8. Alternative Views
9. Many Workspaces
10. Multi-Level Help

Feature, Search, and Browse

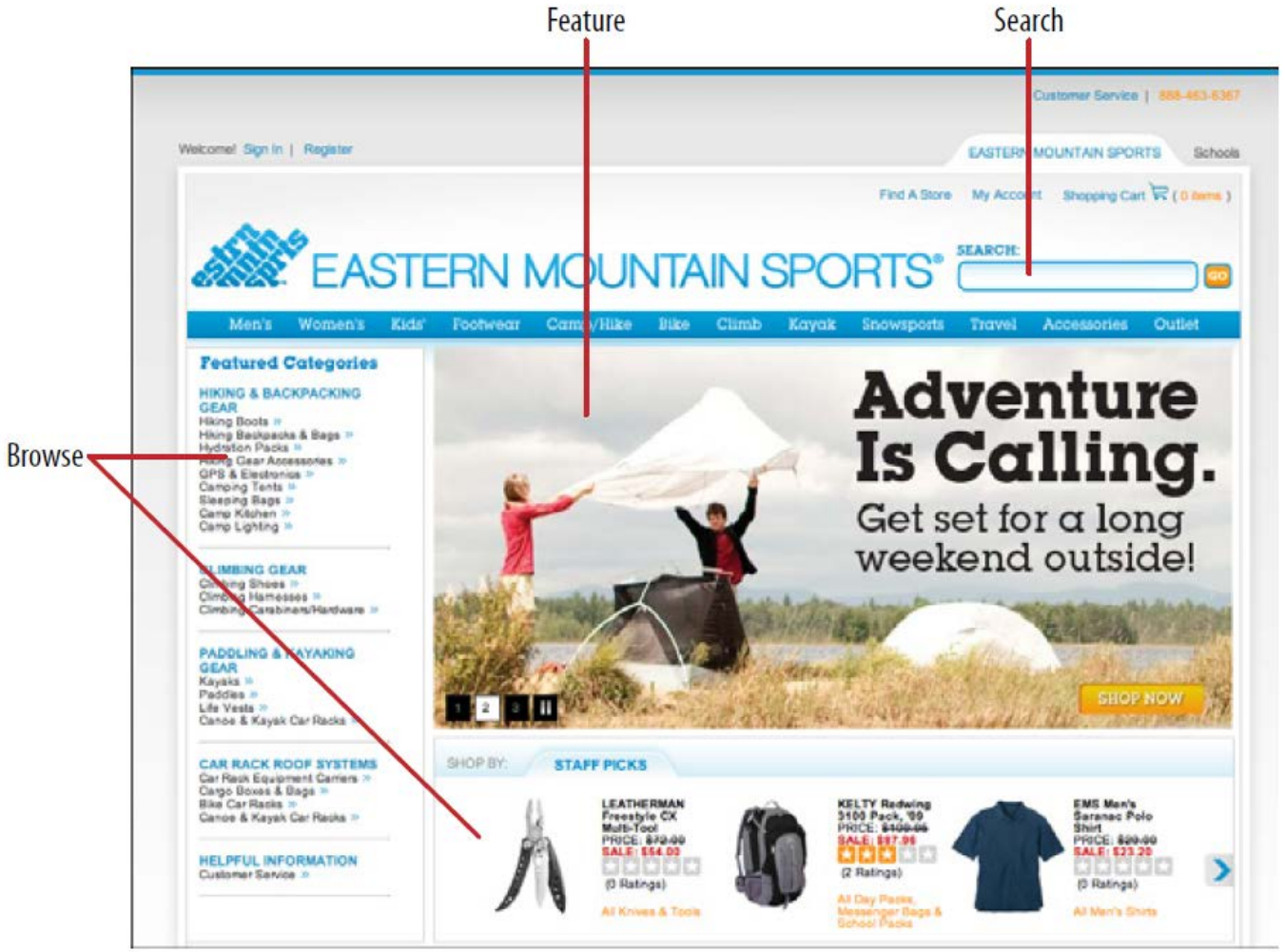


Figure 2-1. *EMS*

Feature, Search, and Browse

- **What:** 3 element on the main page, featured article or product, a search box and a list of items or categories
- **Use When:** When you have a long list of items, products, etc. Want to grab attention with feature.
- **Why:** Super familiar, recognizable, hook the user with features
- **How:**
  - Search box in prominent location, demark it
  - Should be most obvious text field
  - Place browsing very near feature
  - Could have categories with subcategories and mouse-over

- **Related Patterns:** breadcrumbs

Dashboard

Figure 2-19. *Google Analytics*

Dashboard

- **What:** arrange data displays into single information-dense page, updated regularly
  - Show relevant, actionable information
  - Let the customize it
- **Use When:** your system has a lot of incoming data
- **Why**: familiar and recognizable
  - Familiar patterns and components:
    - Titled sections, collapsible panels, movable panels, etc.
- **How:** determine what information your users want to see
  - Group related info, try to avoid scrolling
- **Related Patterns:** many

Wizard

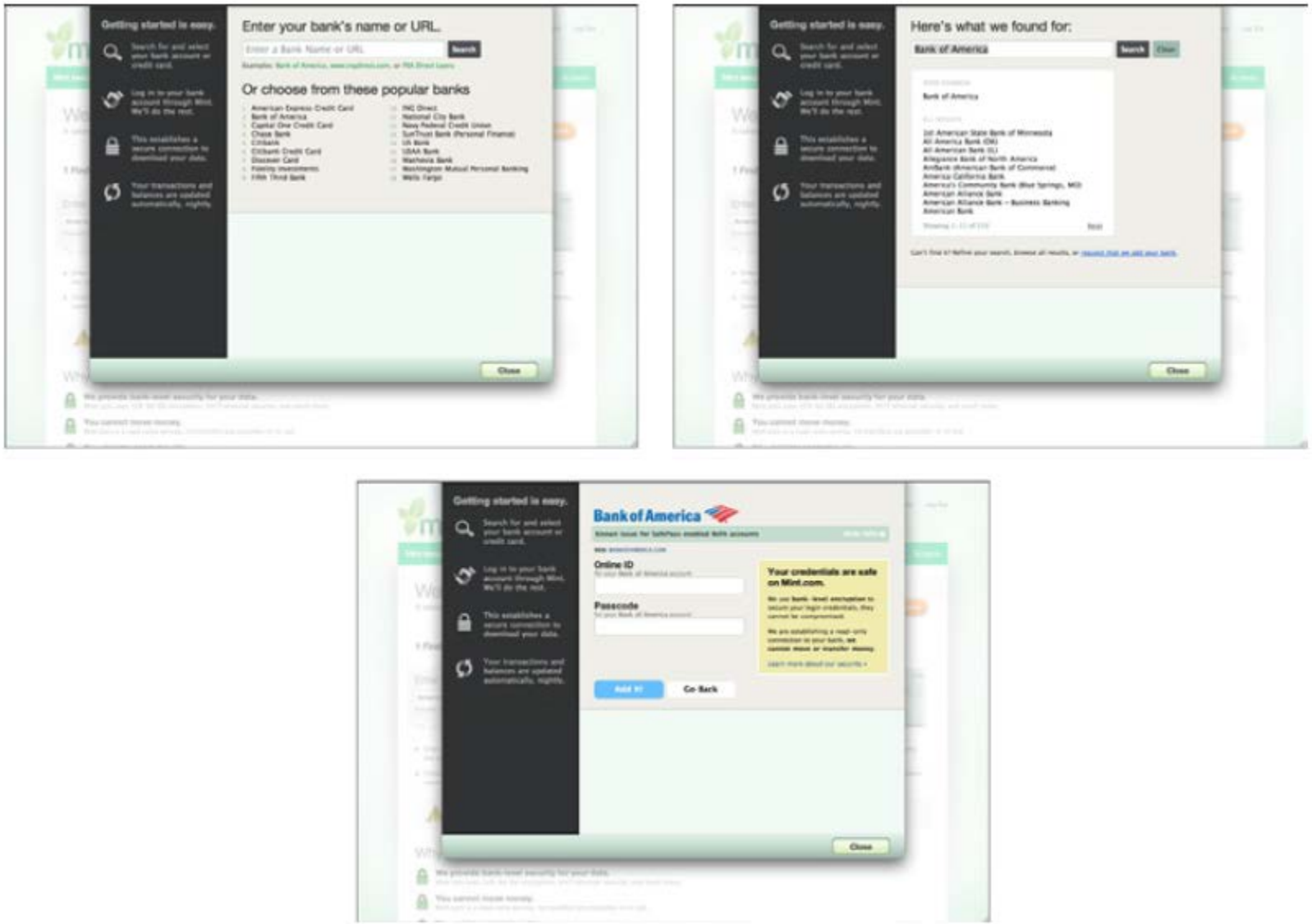


Figure 2-26. *Mint's add-a-bank Wizard*

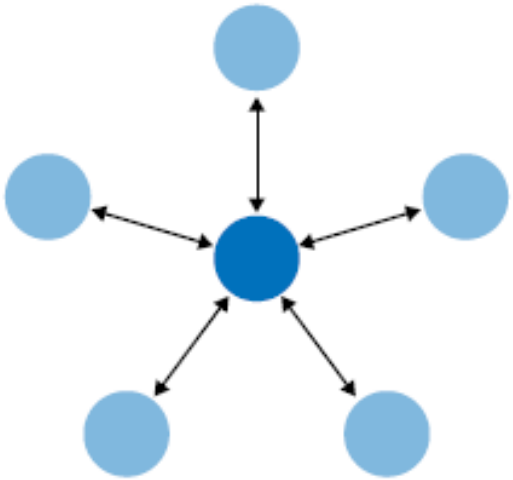
Wizard

- **What:** lead the user through steps in a prescribed order
- **Use When:** you have a long and complicated task that is new for users, not something the do often or want to customize
- **Why:** divide and conquer, split complex tasks into chunks
  - If the task is really complicated, consider splitting it up
- **How:**
  - Chunk the task
  - Allow user to move back and forth
  - Overview of steps
- **Related Patterns:**
  - Titled sections, responsive enabling, good defaults, etc.

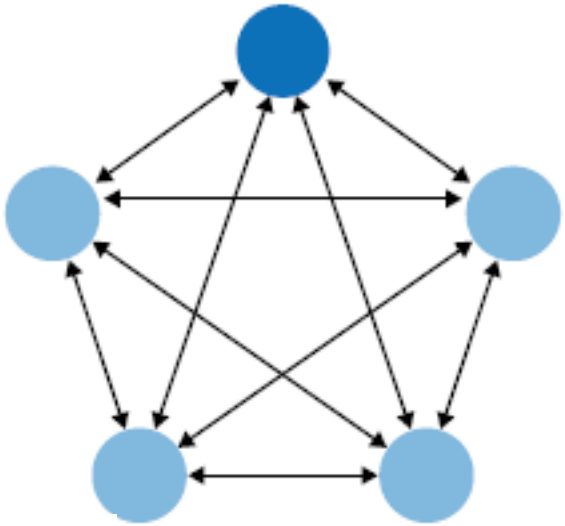
Getting Around: Navigation, Signposts, and Wayfinding

- Navigating can be dull and frustrating
- Less navigation is better (but not always easy?)
- Staying found:
  - Signposts: page and window titles, logos, tabs, selection indications
  - Wayfinding: find way towards goal
    - Good signage
    - Environmental clues (e.g., close button on top right)
    - Maps

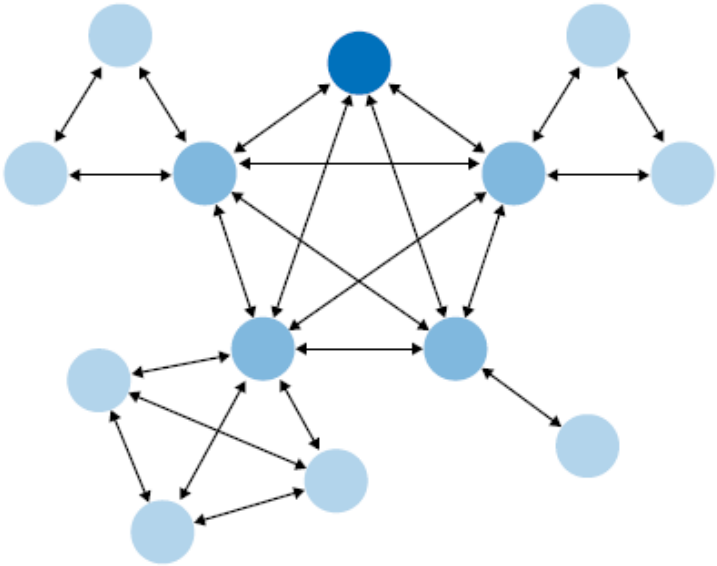
Navigation Models (some)



Hub and spoke



Fully Connected



Multi-level

Navigation Patterns

1. Clear Entry Points
2. Menu Page
3. Pyramid
4. Modal Panel
5. Deep-linked State
6. Escape Hatch
7. Fat Menus
8. Sitemap Footer
9. Sign-in Tools
10. Sequence Map
11. Breadcrumbs
12. Annotated Scrollbar
13. Animated Transition

Clear Entry Points

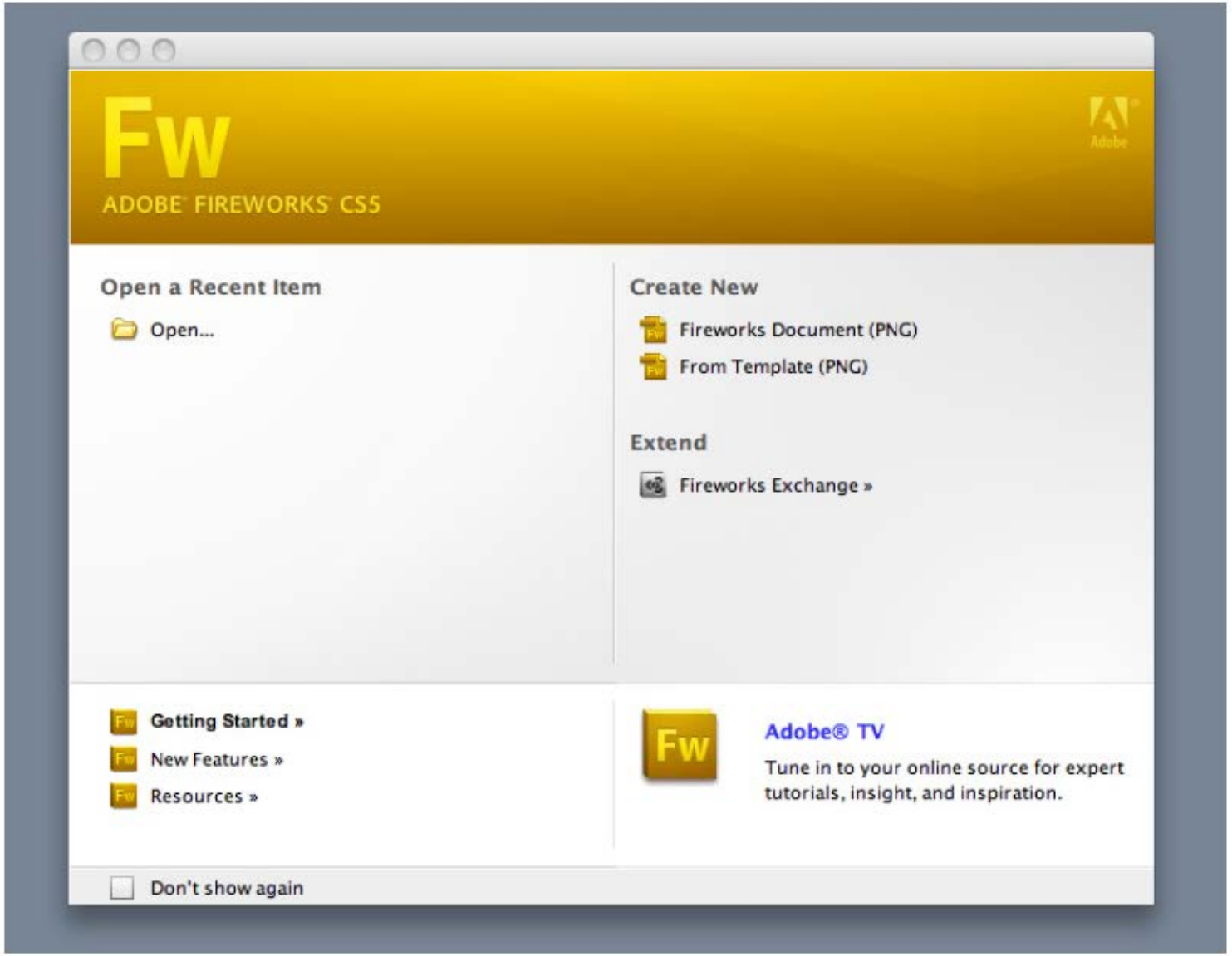


Figure 3-13. *Fireworks startup dialog*

Clear Entry Points

- **What:** present only a few main entry points to the interface, make them task-oriented
- **Use When:** site with many first-time or infrequent users
- **Why:** A few options for getting started, clear tasks
- **How:**
  - Entry points should cover most reasons users would want to be there (use cases)
  - Use understandable language
  - Emphasize them relative to their importance
- **Related Patterns:** none?

Modal Panel

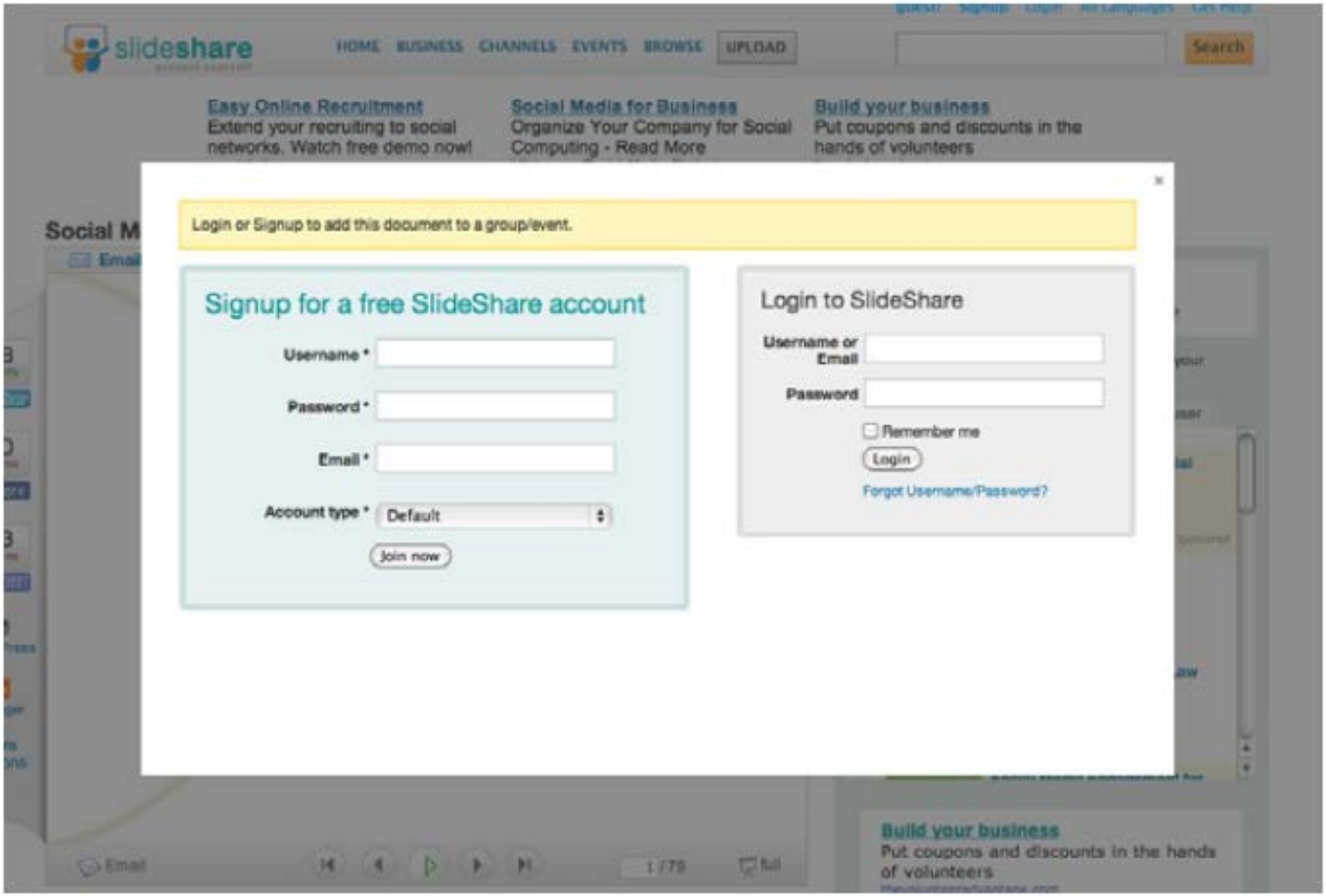
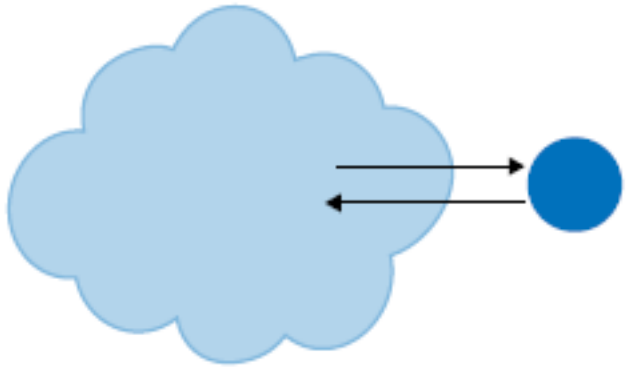


Figure 3-23. *SlideShare's login modal panel*

Modal Panel

- **What:** show only one page with no other options until the users finishes the immediate task
- **Use When**: the app or site has gotten to a stage in which it cannot proceed without input from the user
- **Why:** easy to understand, cuts off users from other functions
- **How:**
  - On the same screen, place a panel, dialog box, or page that requests the needed information
- **Related Patterns:**
  - Escape hatch



Breadcrumbs

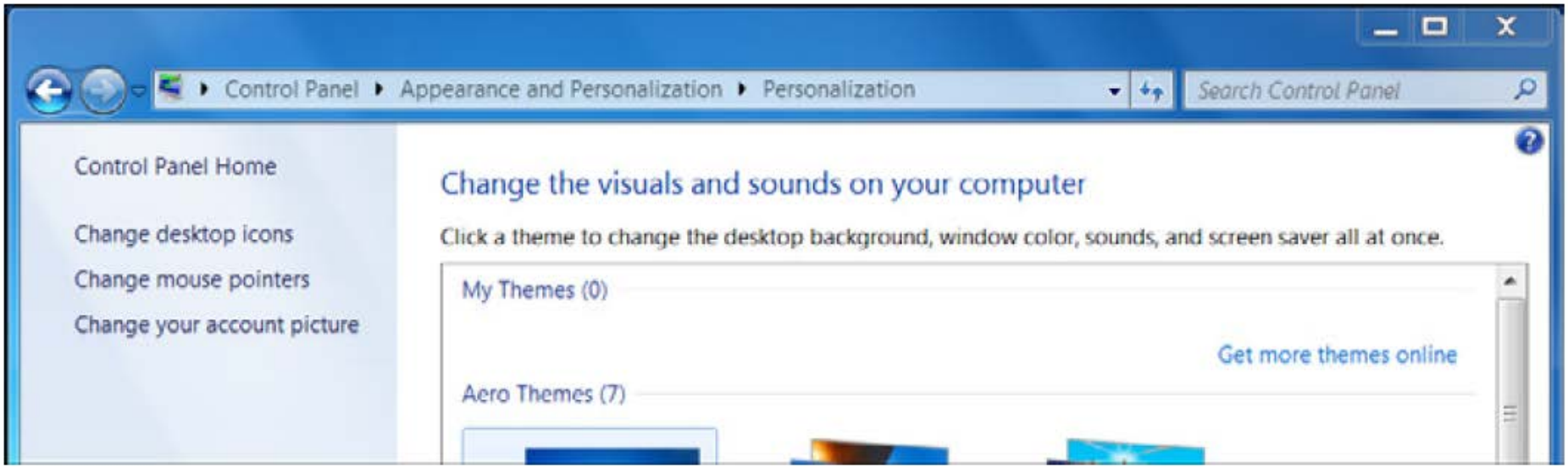


Figure 3-53. *Windows 7 control panel*

MotheringDotCommunity Forums > Parenting > Ages and Stages > The Childhood Years
Birthday Party Food Ideas--Four Year Olds

Figure 3-54. *Mothering.com forums*

Breadcrumbs

- **What:** pages with deep navigational hierarchy, show a list of all parent pages
- **Use When:** there is a hierarchical structure with more than two levels
- **Why:** helps the user figure out where they are, context
- **How:**
  - Near the top of the page, put a line of text or icons indicating the current hierarchy > / or >>
  - Labels should be page titles
- **Related Patterns:**
  - sequence map

Organizing the Page: Layout of Page Elements Lists of Things

- "Page layout is the art of manipulating the user's attention on a page to convey meaning, sequence, and points of interaction."
- a good visual hierarchy gives instant clues about:
  - The relative importance of page elements
  - The relationships among them

Emphasis

- How to make things look important:
  - Density, background color, rhythm, position and size

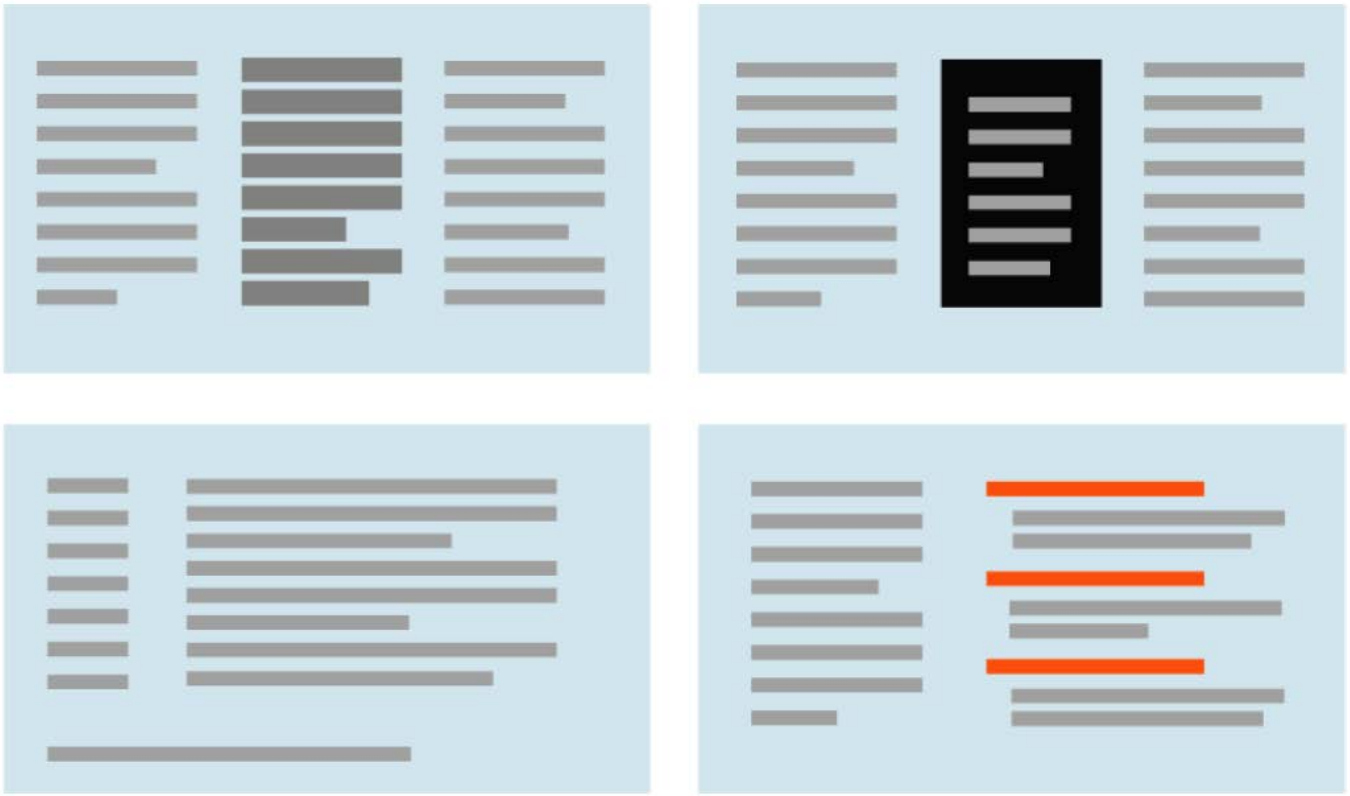


Figure 4-2. *Ways to emphasize blocks of text or small items (clockwise from upper left): density, background color, rhythm, and position and size*

Showing Relationships

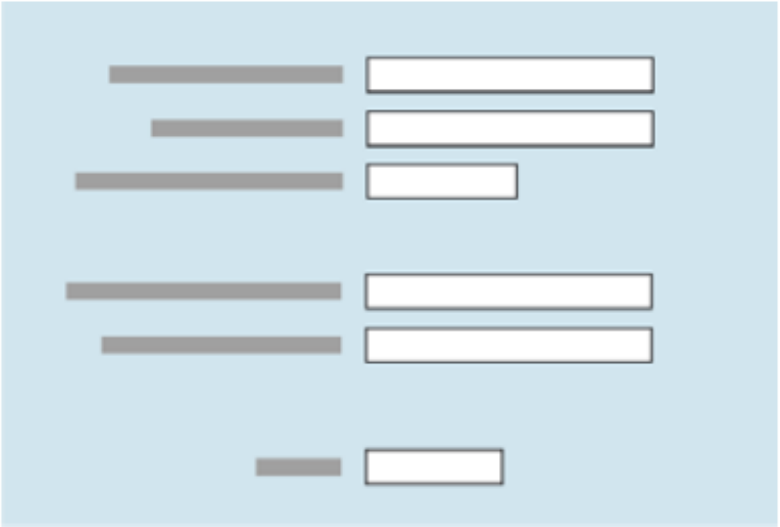


Figure 4-4. *Grouping related items*

Figure 4-5. *Peer items*

Figure 4-6. *Distinguishing one item among peers*

Figure 4-7. *Lists of items*

Gestalt Principles (from psychology)

- *Proximity*
  - Put things close together, and viewers will associate them with one another. This is the basis for strong grouping of content and controls on a UI.
- *Similarity*
  - If two things are the same shape, size, color, or orientation, for instance, viewers will also associate them with each other.
- *Continuity*
  - Our eyes want to see continuous lines and curves formed by the alignment of smaller elements.
- *Closure*
  - We also want to see simple closed forms, such as rectangles and blobs of whitespace, that aren't explicitly drawn for us. Groups of things often appear to be closed forms.

Gestalt Principles (from psychology)

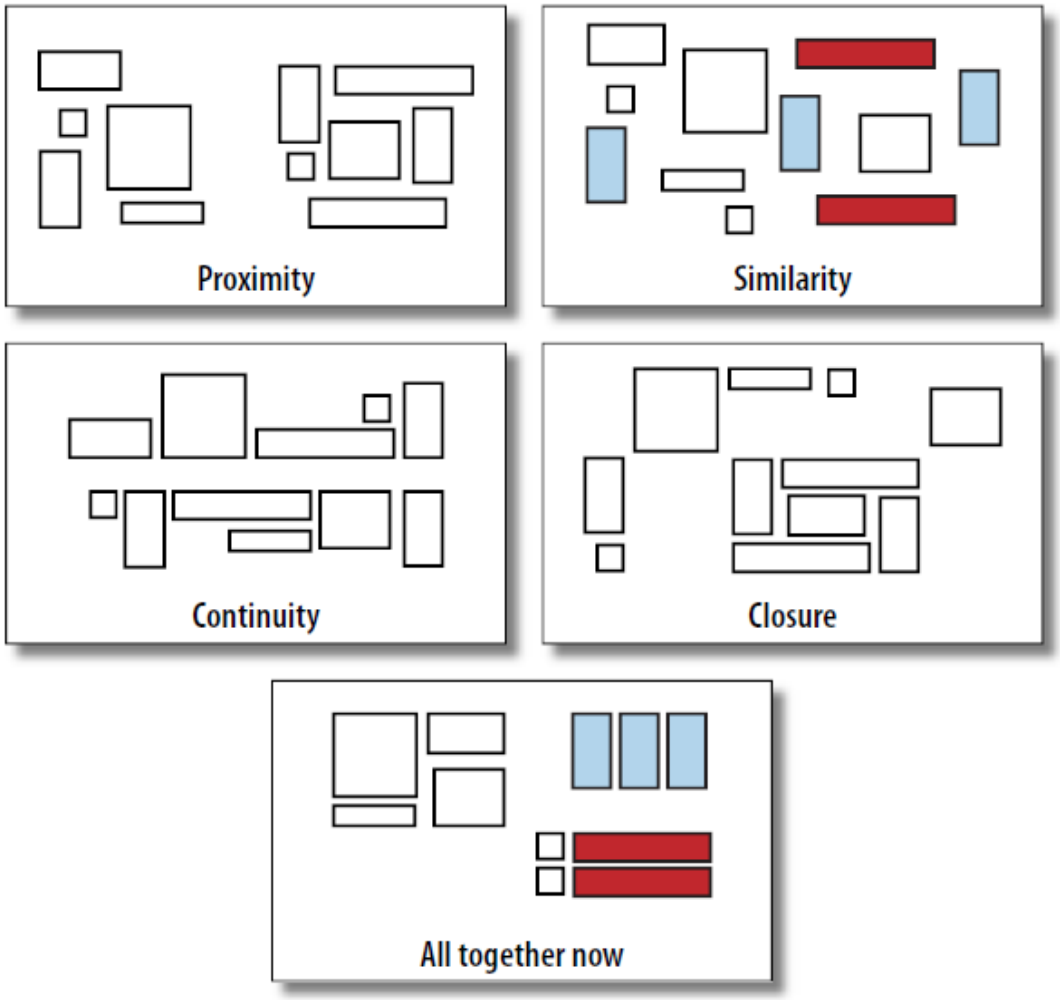


Figure 4-14. *Four Gestalt principles*

Layout Patterns

1. Visual Framework
2. Center Stage
3. Grid of Equals
4. Titled Sections
5. Module Tabs
6. Accordion
7. Collapsible Panels
8. Movable Panels
9. Right/Left Alignment
10. Diagonal Balance
11. Responsive Disclosure
12. Responsive Enabling
13. Liquid Layout

Visual Framework

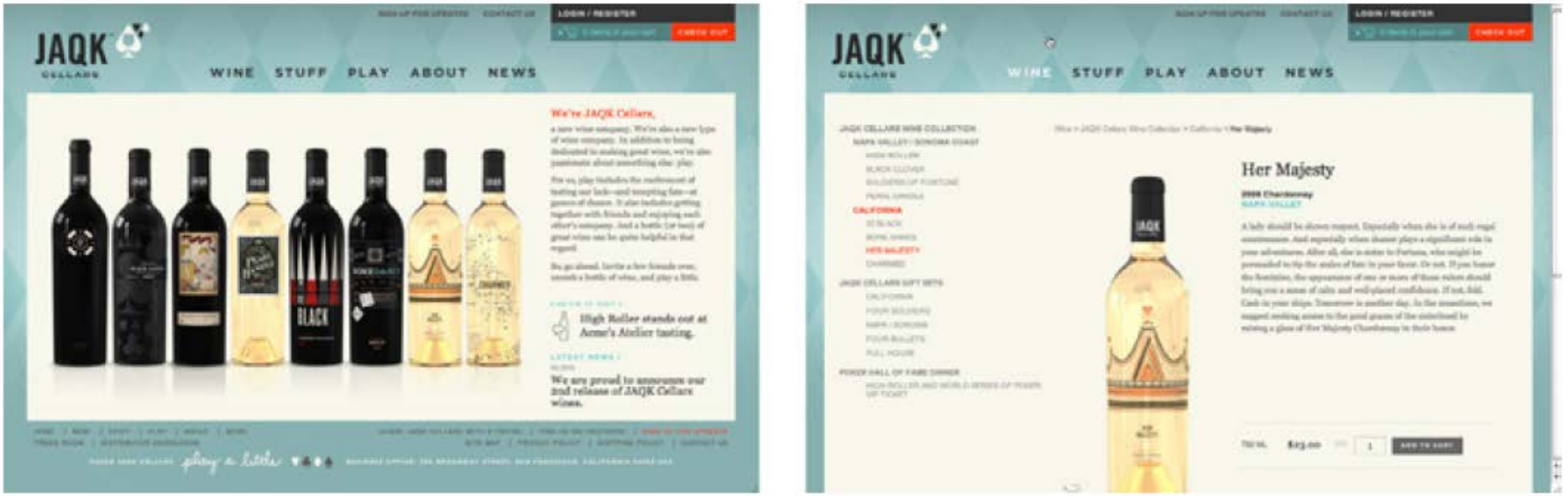


Figure 4-15. *JAQK*

Visual Framework

- **What:** Design each page to use the same basic layout
  - Colors, style, with some flexibility
- **Use When:** you're building a website with many pages
  - Want it to appear cohesive, part of same site
- **Why:** familiar layout to find things (signposts)
  - Helps the page content stand out more (as the layout and style are familiar)
- **How:**
  - Color, Fonts, Writing style and grammar
  - Layout grid
- **Related Patterns:** breadcrumbs, module tables, sequence map, title sections, etc.

Titled Sections

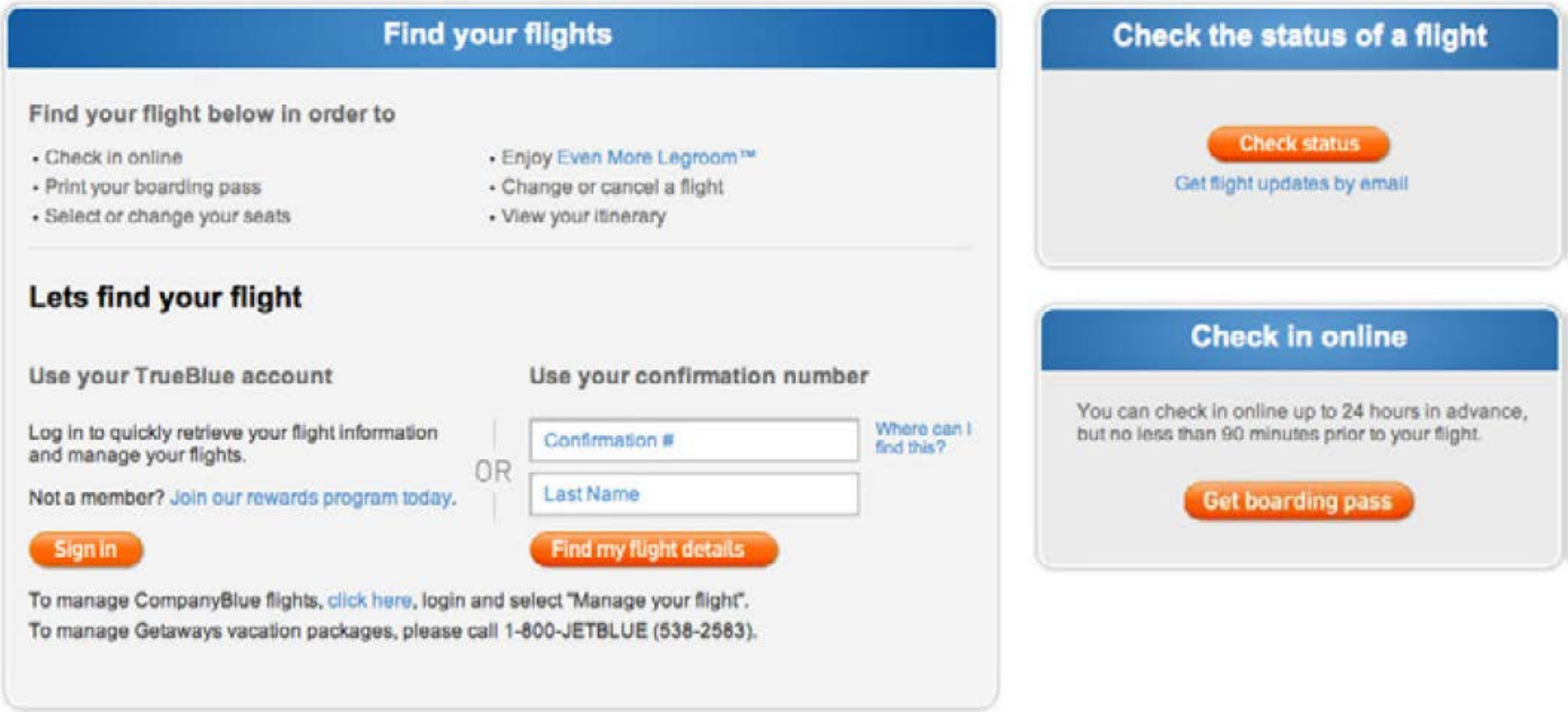


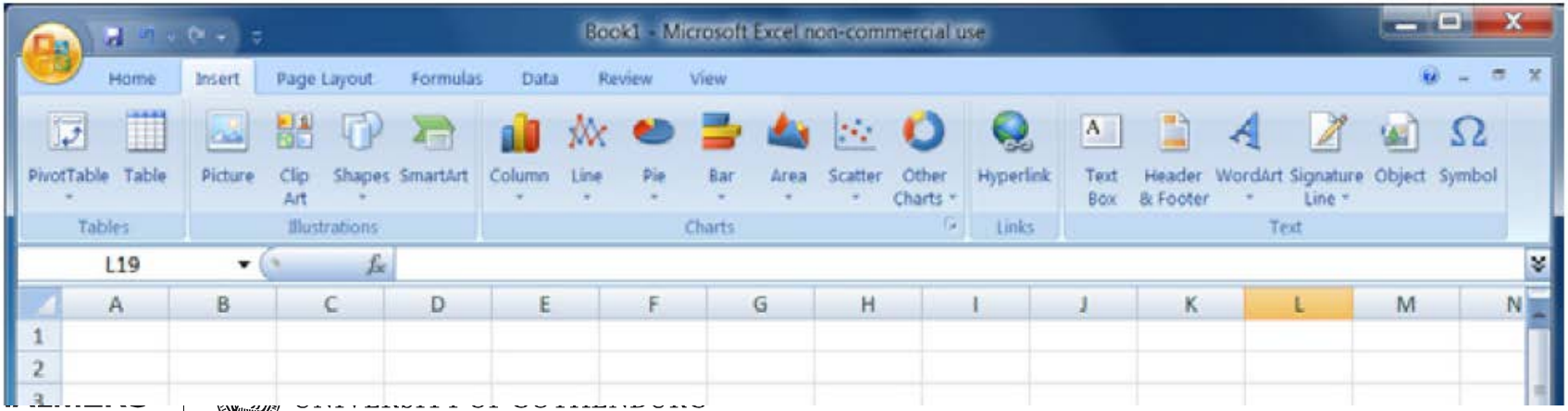
Figure 4-28. *JetBlue's titled sections*

Titled Sections

- **What:** define separate sections of content, separate visually
- **Use When:** you have a lot of content, but want to make the page easy to scan and understand
- **Why:** can understand each page in chunks, the eye is guided along the sections in the page
- **How:**
  - Information architecture: spit into coherent chunks with short memorable names
  - Make the sections stand out
- **Related Patterns:** Module Tabs, an Accordion, or Collapsible Panels

Module Tabs

Figure 4-31. *MapQuest*



Module Tabs

- **What:** put models of content into small tabbed area, one tab visual at a time
- **Use When:** you have a lot of different content to show on the same page, no room for everything
  - Page content can be grouped into modules
  - Modules are roughly the same size and independent
  - < 10 modules (even less better)
  - Modules content is related or similar
- **Why:** familiar convention
- **How:**
  - Information architecture: spit into coherent chunks with short memorable names (again)
  - Indicate which tab is selected
- **Related Patterns:** Accordions, Movable Panels, Collapsible Panels, Titled Sections, Carousel

Questions?