3.1: IA Principles & Frameworks

Learning Goals

Discuss the principles, frameworks, and historical context of information architecture

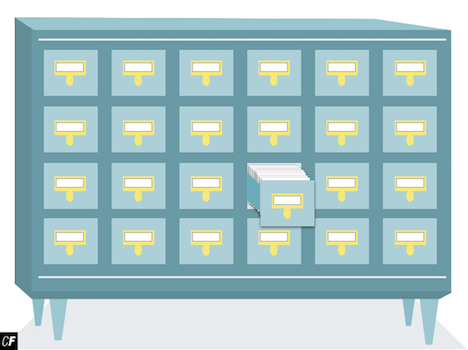
 Estimated Read Time: 30 Minutes.

Introduction

Welcome back! Great job on getting through Achievement 2! Today, we’re moving on to Achievement 3, which will be focused on the foundations of technical design and the organization of information in your application.

To kick things off, we’ll dive into the world of information architecture (IA), also known as information design. IA is the art of organizing information to make it easier for users to consume. Yes, you heard correctly—organizing information is more art than science! This is because IA relies heavily on many sets of guidelines and best practices along with personal experience. In other words, it’s a craft that requires creative thinking and problem solving.

A Brief History of IA



Humans have always been interested in creating systems to better organize their information. In 330 B.C., Ancient Egypt's Library of Alexandria listed its contents in a 120-scroll bibliography. In the 1870s, the Dewey Decimal System was created to better organize large library collections. Nowadays, however, IA tends to be primarily associated with the design of information on the web.

In 1964, IBM defined "architecture" as:

"The conceptual structure and functional behavior distinguishing the organization of data flows and controls, logical design, and physical implementation."

This definition was quite revolutionary. Before this, "architecture" was concerned solely with the layout and flow of buildings. To IBM, however, it seemed natural to adopt this term for creating the layout and flow of computer systems.

When building a structure, it doesn’t matter how well it’s painted and decorated if the foundation itself is poorly built. The same is true for design. One could build the most beautiful, user-friendly interface, but if the underlying data structure were poor, the information would still be unorganized and difficult to access.

It wasn’t until 1976 that the term "Information Architect" came about. Richard Saul Wurman, before he became the founder of TED (Technology Entertainment and Design), coined the term at an American Institute of Architects (AIA) conference. Wurman himself was a graphic designer and architect, and he believed that information should be structured in the same way as a building (i.e., with a solid foundation to build upon).

While these terms are more specific to areas *within* IA than with IA itself, here are some other terms often used as synonyms for IA:

* Usability Engineering
* Content Management
* Content Strategy
* UX Design
* Interaction Design (IxD)

Now that we've delved a bit into the history of IA, let’s take a look at the various methodologies associated with it.

IA Methodologies

The modern field of IA began far before the proliferation of web and mobile apps. Thus, it makes sense that many methodologies have roots in the physical world of IA, and why many modern UX designers still draw from it.

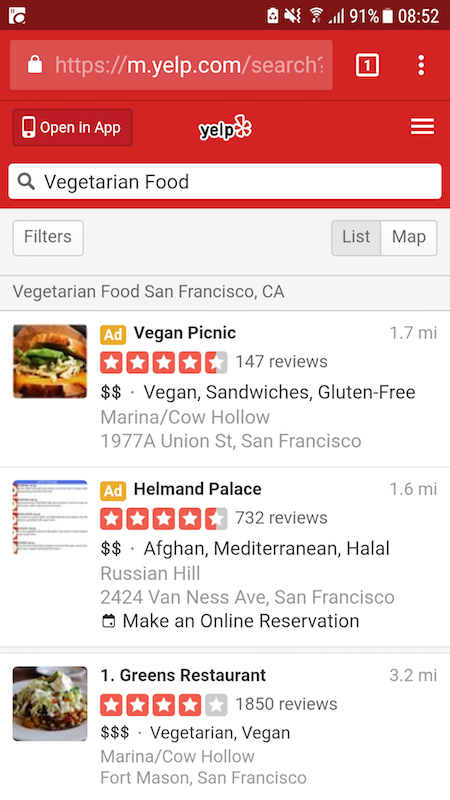
It’s important to realize that your website or app is nothing more than an archive of information. When a user logs in and is presented with a home screen or dashboard, how do they know where to go? Which buttons should they press to find the information and perform their desired task?

Below, we’ll take a look at two IA methodologies often incorporated in UX design.

Library Science

**Library science** is the study of how to categorize, catalog, and locate information. It’s all about creating metadata, which can be used to locate a resource later. Contrary to what its name may imply, library science isn’t strictly limited to libraries. It’s particularly useful to the IA side of UX design, which is concerned with creating organized applications.

One example we can look at is Yelp, a website, web app, and mobile app that lets you find nearby businesses and restaurants. While it may be based on location and proximity, it’s really no different than a library. Imagine that a location on Yelp is a book in a library. It’s the UX designer’s job to make sure the user can easily weed through the millions of restaurants in order to find the restaurant they seek.



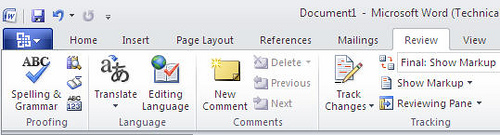
Web app screen of Yelp

Cognitive Psychology

**Cognitive psychology** is the study of how the mind works and the underlying processes that take place in a person’s mind. A few important aspects of cognitive psychology that directly relate to UX design are cognitive load, mental models, and decision-making. As you read through the following terms and review the corresponding images, think about how each plays into your user journey.

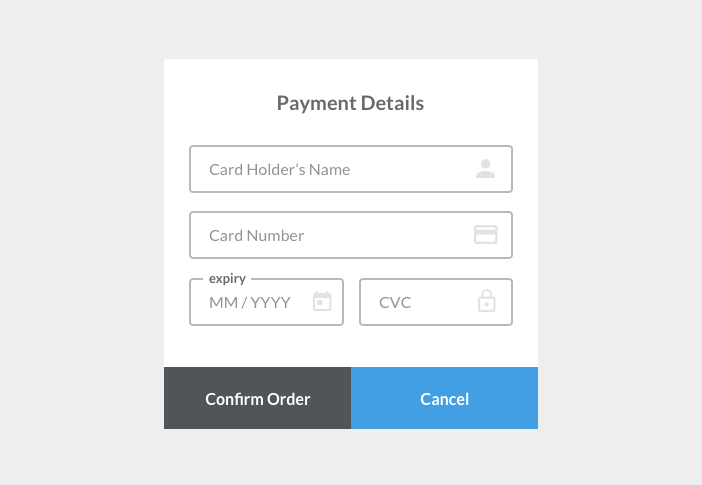
**Cognitive Load**

Cognitive load refers to how much information a person can process at any given time. Keeping this load low is important to prevent overloading a user with too much data at once. Take a look at the image below. Although the text and the icons are grouped in a relatively logical order, it still takes some time to look through each function and spot the one that’s needed. Imagine if you had to do this for every function you wanted to perform!



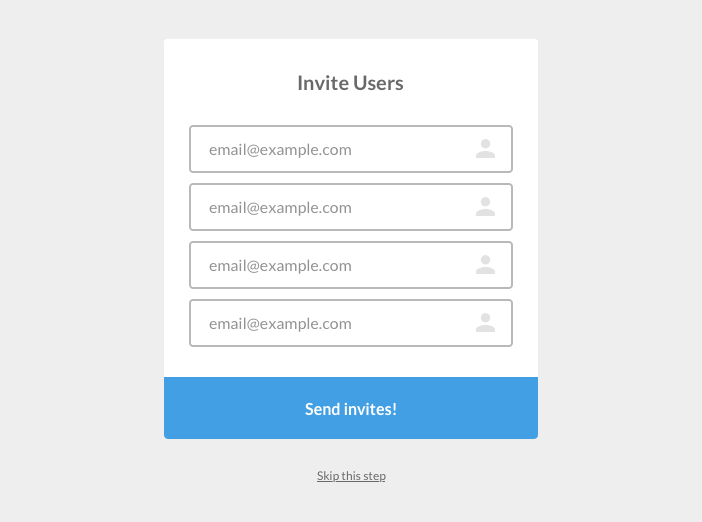
**Mental Models**

As mentioned in Exercise 2.5: Mental Models & User Journeys, a mental model is an assumption that a user may have before using a given system. In the image below, a user fills out a form and expects the brightest button below the form to be the submission button, allowing them to move on to the next step. What if, however, this button actually *cancelled* the process? Doing so would violate users' previous assumptions and create intense friction.



**Decision-Making**

People don’t want to make hard decisions. As the UX designer, it’s your job to make sure the decisions users are faced with are simple and clear. In the image below, the user only has to make one decision that relates directly to the task at hand: Do I fill out the form and press send? Or do I skip this step? There are no other links or buttons on the screen, making it distraction-free and, thus, easier to accomplish.



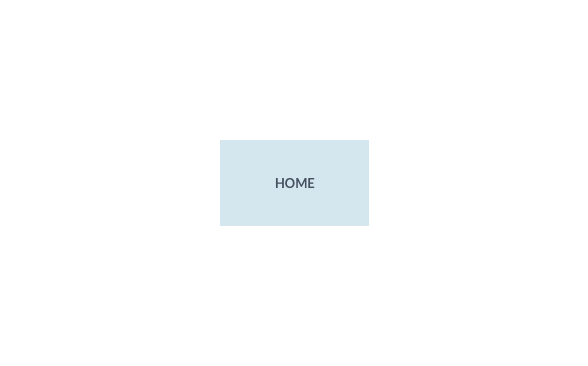
Common IA Structures

Now that you have a solid handle on the history of IA and how its methodologies can be readily applied to UX, let’s talk about specific IA structures that are commonly used in web and mobile app design. **IA structure** refers to how pages are laid out, connected, and given different levels of hierarchy within a website or application. These structures are illustrated using something called a **sitemap**, which is an organizational map that outlines the hierarchy of a website or application. Sitemaps are inherently different from user flows as they focus only on the page hierarchy rather than the specific actions the user takes to traverse the application. As you’ll notice in [this example](https://s3.amazonaws.com/coach-courses-us/public/courses/ux-immersion/A3/E1/A3E1_travelappsitemap.png), sitemaps are typically numbered to indicate hierarchy. It’s especially useful to use numbers when you have a big site with many pages as you'll be able to quickly refer to a specific page in conversation.

The nature of your product, how complex the content is, and what types of content your product offers will all determine which structure you choose when designing the information architecture of your product. There are several types of structures that can be utilized in IA. Below are examples of some of the most common structures with their corresponding sitemaps:

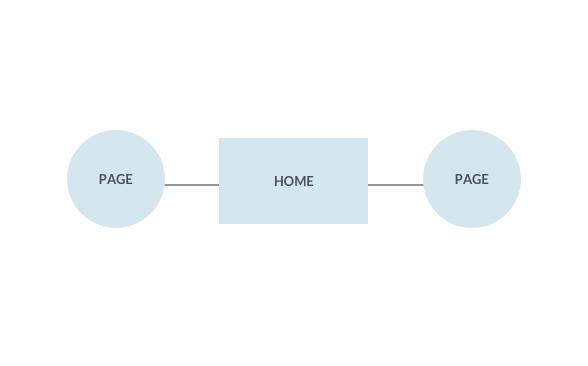
Single Page

This is exactly what is sounds like—an application or website that contains all functionality and information within a single page. This is great for very simple websites and applications (e.g., the website for a local restaurant where a visitor can view the menu, opening hours, and contact information all on one page).



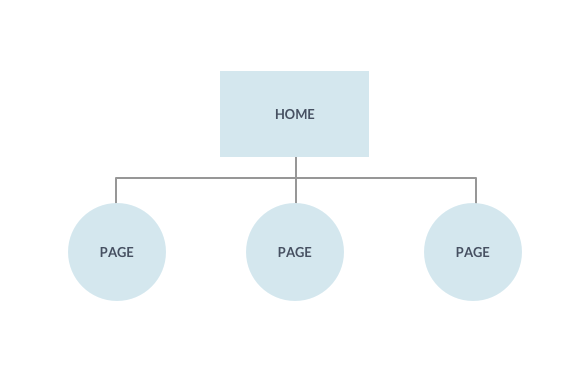
Flat Structure

In this structure, each page is at the same level in the hierarchy. No page is more important than the other, and you can’t go any deeper within the hierarchy. This structure works well with informational, brochure-style websites where all pages are equal in importance (e.g., a site that only contains a “Home” page, an “About” page, and a “Contact” page). It can work well on informational websites and apps for small businesses such as flower shops or restaurants.



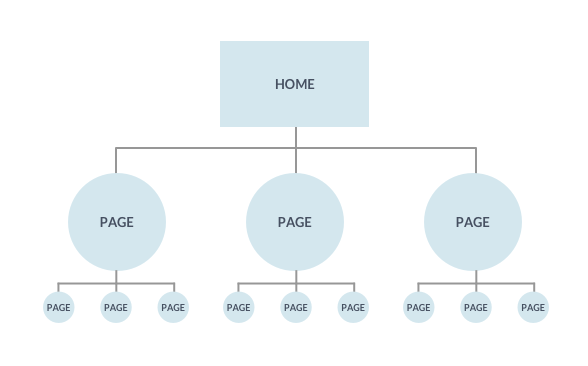
Index Page

This is similar to the flat structure discussed above. The one difference, however, is that visitors will first land on a home or index page that’s higher than all other pages. From here, they can navigate to different areas of the site. Each of these areas are of equal importance in the site hierarchy. If this site was for a restaurant, for example, it may include a “Home” or “Index” page from which you can navigate to a “Menu” page, a “Contact” page, and an “Opening Hours/Calendar” page.



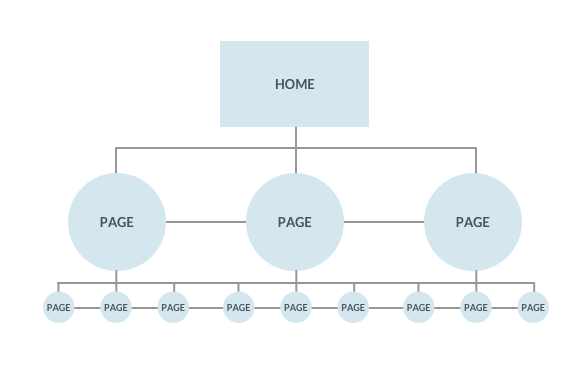
Strict Hierarchy

This structure is a little more complex than those explored previously. Here, you navigate to subpages from the “Home” page. Each subpage (parent) leads to a series of its own subpages (children), but you have to follow the correct path in order to reach the child subpages. This structure is used for more complex sites such as online newspapers, stores, and blogs where content is organized by multiple categories. Think of a local newspaper site. You may have a “Home” page with links to the daily news. From here, however, you can navigate to different subcategories (think “Sports,” “Environment,” and “Politics”). From each subcategory, you can dive even deeper. On the “Sports” page, for example, you can navigate to “Basketball,” “Football,” and “Baseball.”



Co-Existing Hierarchies

This structure is the same as the strict hierarchy discussed above with one exception—you can access each child page from different parent pages. This works well if there’s a great deal of overlapping information on your site or app. For example, if a user were browsing an article on seasonal vegetables on your good food blog, you might want them to have easy access to the “Recipes” page to make use of their new knowledge.



These structures aren’t necessarily rules, and as you’ll surely notice, it’s possible to mix and match these to suit your application or website structure. They can be very useful in creating foundations for your own sitemaps and layout flow charts. Next time you’re browsing a website or using an app, pay attention to the IA structures incorporated as well as how these structures work with the site content and the functionality presented. Make note of anything that catches your attention and start thinking about which IA structures would work best for your project.

Where IA and UX Meet

Some UX design teams have their own dedicated “Information Architect,” but not all teams are so lucky (nor their employers so profitable). This means the task of creating an information architecture falls squarely on the shoulders of the UX designers themselves. This is why it’s so important that you have a solid understanding of IA and its implications for UX design.

The most common tasks of IA include creating:

* User flows (which you’ve done in Exercise 2.6: Task Analysis & User Flows)
* Wireframes
* Sitemaps
* Navigational structures
* Labeling and data modeling

Each of these tasks is closely related to overall user experience, so you can see why this job would fall into the lap of the UX designer.

Summary

Today, we dug deep into the world of information architecture by discussing its origins and how it was adapted from the field of architecture, as well as how the field of IA is now commonly associated with web and mobile app design. We also learned about IA methodologies and some common layout structures.

Now, you’re going to get the chance to become an IA by designing your own sitemap.

Resources

* [Why IA Matters for UX — A Brief History of Information Architecture](https://uxdesign.cc/a-brief-history-of-information-architecture-d26b17205e7b)
* [The History of Information Architecture by Dyno Mapper](https://dynomapper.com/blog/19-ux/187-history-of-information-architecture)
* [Explaining Information Architecture by Dan Klyn](https://vimeo.com/8866160)
* [Sitemap for Course Demo Project: Triply](https://s3.amazonaws.com/coach-courses-us/public/courses/ux-immersion/A3/E1/A3E1_travelappsitemap.png)

Take the quiz to test your knowledge on this Exercise.

Take Quiz

Task

* [DIRECTIONS](https://careerfoundry.com/en/course/become-a-ux-designer/exercise/ia-principles-and-frameworks#directions)
* [SUBMISSION HISTORY](https://careerfoundry.com/en/course/become-a-ux-designer/exercise/ia-principles-and-frameworks#step_submission_history)

 Estimated Task Time: 3 Hours.

In this Task, you’ll create a sitemap for your project.

**Directions**

1. Review the different types of site structures above and decide which type is the best fit for your web app. Remember, it’s also okay to mix and match if necessary!
2. Review the user journeys and personas you created in Achievement 2 to make sure you have a solid handle on the primary screens of your app.
3. Start with a home screen or dashboard. From there, branch off to each page of your site and create tiles and lines to represent the information architecture of your app. As you create your sitemap, make sure to use a number hierarchy to keep track of all the pages. For inspiration, take a look at [this sitemap example](https://s3.amazonaws.com/coach-courses-us/public/courses/ux-immersion/A3/E1/A3E1_travelappsitemap.png) for the course demo app, Triply.
   * To create your sitemap, you can use any tool you're comfortable with that allows you to create boxes, text, and lines, such as Sketch, [OmniGraffle](https://www.omnigroup.com/omnigraffle), [Balsamiq](https://balsamiq.com/), Google Drawings, [Lucidchart](https://www.lucidchart.com/" \t "_blank), or even PowerPoint.
4. Upload a PDF of your complete sitemap here to discuss with your Tutor!