**UI Design**

UI design, also known as user interface design, refers to the aesthetic design of all visual elements of a digital product’s user interface; namely the product’s presentation and interactivity.

UI design is often confused with UX design, also known as user experience design.

While UI and UX designers work closely together, the two fields refer to separate aspects of the design process.

**UX design** is the process of enhancing user satisfaction by improving the usability and accessibility of a product, webpage, or app. On the other hand, UI design is the design of the product’s interface—in other words, what the user actually sees when they interact with the product. From color schemes to typography, UI designers are responsible for the product’s look and feel. UI design involves anticipating the user’s preferences and creating an interface that both understands and fulfills them. UI design not only focuses on aesthetics, but also maximizes the responsiveness, efficiency, and accessibility of a website.

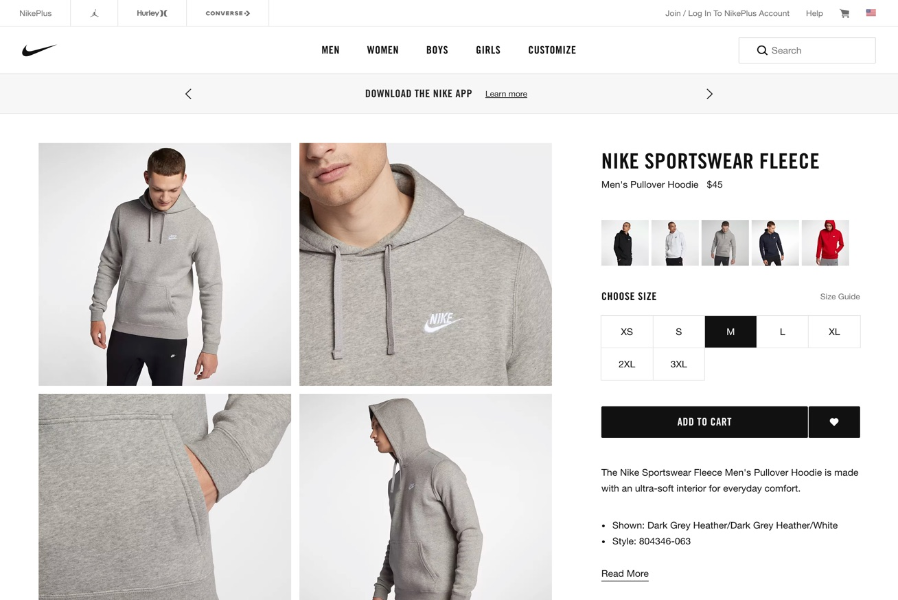
If you imagine a product as the human body, the bones represent the code which give it structure. The organs represent the UX design: measuring and optimizing against input for supporting life functions. And UI design represents the cosmetics of the body; its presentation, its senses and reactions.

“User Experience (UX) and User Interface (UI) are some of the most confused and misused terms in our field. A UI without UX is like a painter slapping paint onto a canvas without thought; while UX without UI is like the frame of a sculpture with no paper mache on it. A great product experience starts with UX followed by UI. Both are essential for the product’s success.”

User Interface Design is the craft and process of designing what a user interacts with when communicating with software.

UI design is a multidisciplinary field that requires UI designers to wear multiple hats as part of one role. While UI designers need a keen visual eye, there’s also a psychological aspect that many don’t consider to be a part of visual design. To design user-friendly interfaces, UI designers need to understand how people work—and how each visual, interactive element shapes their experience.

Here is a typical user interface on a web site. There are slightly different UI considerations for web, mobile, desktop and other types of software, but, generally speaking, they are all software, and with few concepts applied on them.



## **What is a user interface?**

Put simply, a user interface is the point of human-computer interaction and communication on a device, webpage, or app. This can include display screens, keyboards, a mouse, and the appearance of a desktop. User interfaces enable users to effectively control the computer or device they are interacting with. A successful user interface should be intuitive, efficient, and user-friendly

## **What are some of the most important elements of a user interface?**

User interface elements are the parts we use to build interactive websites or apps. They provide touchpoints for the user as they navigate their way around; from buttons to scrollbars, to menu items and checkboxes.

User interface elements usually fall into one of the following four categories:

### **Input Controls**

Input controls allow users to input information into the system. If you need your users to tell you what country they are in, for example, you’ll use an input control to let them do so.

### **Navigation Components**

Navigational components help users move around a product or website. Common navigational components include tab bars on an iOS device and a hamburger menu on an Android.

### **Informational Components**

Informational components share information with users. This includes notifications, progress bars, message boxes, and pop-up windows.

### **Containers**

Containers hold related content together, such as accordions. An accordion is a vertically stacked list of items that utilizes show/hide functionality.

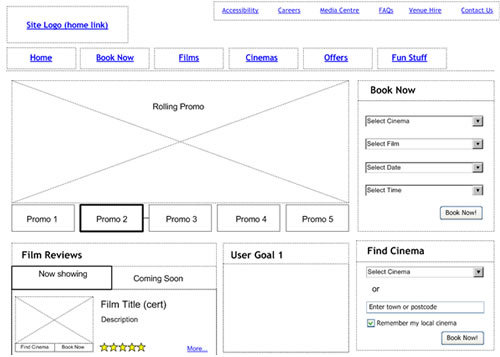
So, let's begin by stepping into the shoes of a UI designer to see how they might approach this website UI.

## **UI design layer 1: Controls**

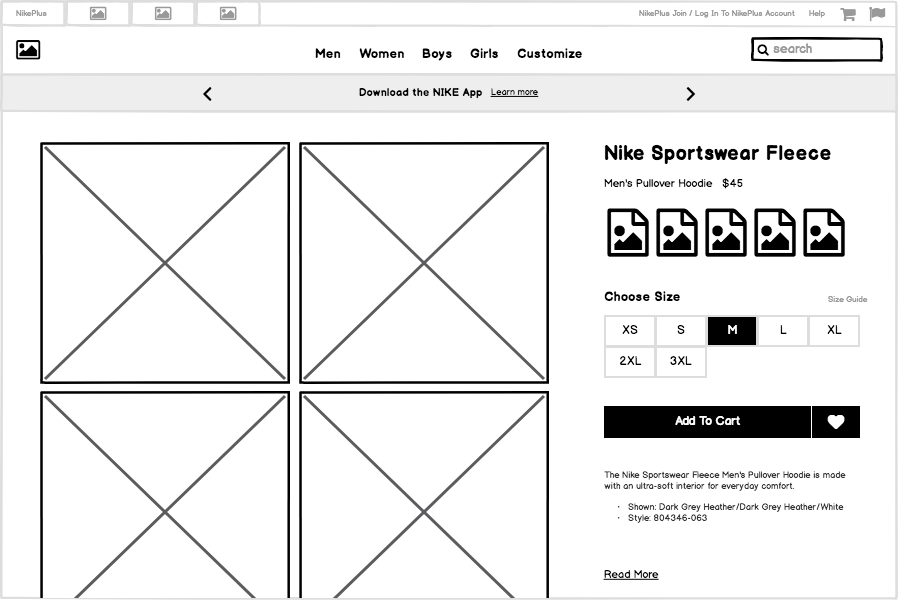
Before diving into the Controls layer, let's simplify the page above by viewing it as a wireframe.

A wireframe is a layout of a web page that demonstrates what interface elements will exist on key pages. It is a critical part of the interaction design process.

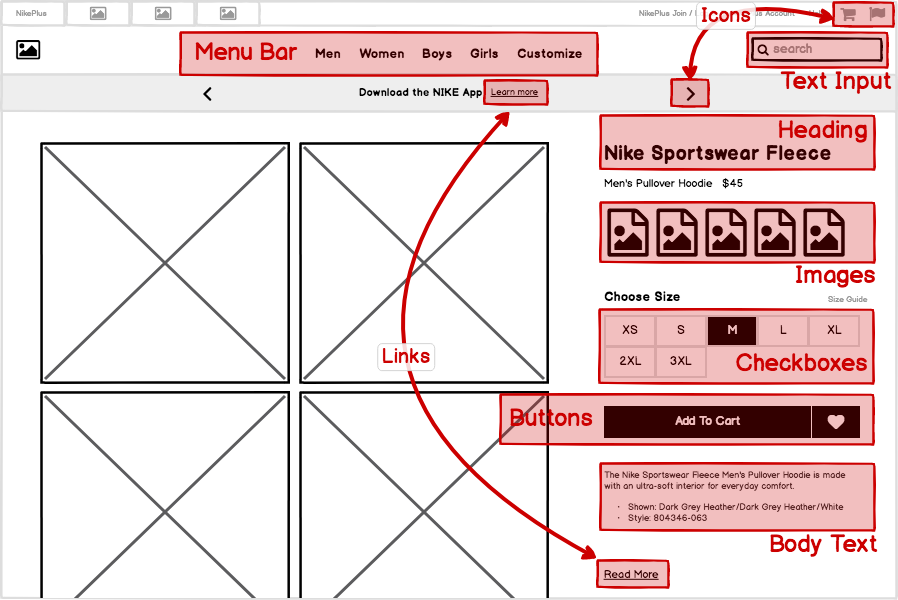
The aim of a wireframe is to provide a visual understanding of a page early in a project to get stakeholder and project team approval before the creative phase gets under way. Wireframes can also be used to create the global and secondary navigation to ensure the terminology and structure used for the site meets user expectations.



From a practical perspective, the wireframes ensure the page content and functionality are positioned correctly based on user and business needs. And as the project moves forward they can be used as a good dialogue between members of the project team to agree on the project vision and scope.



Now let's look at some of the UI Controls that were used to build this page

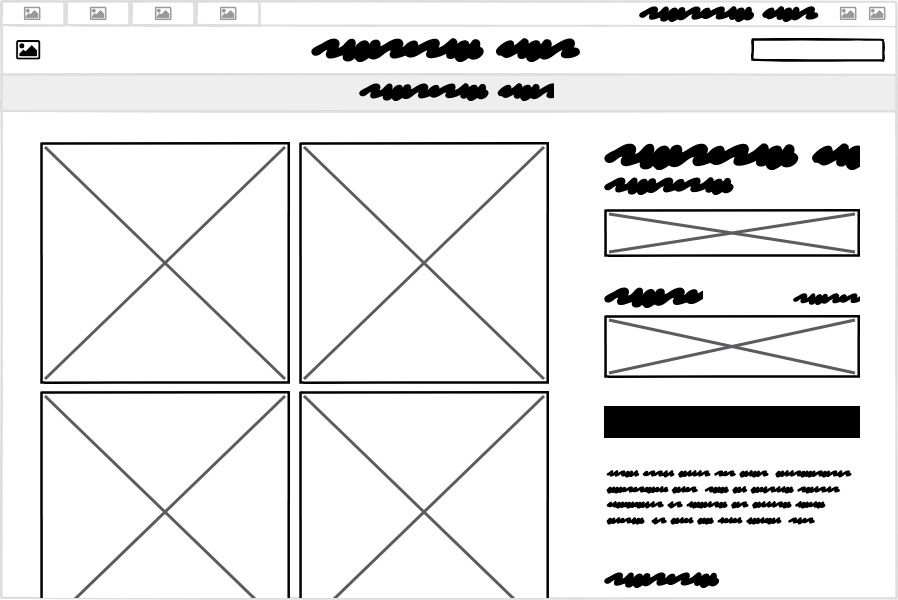


User interface controls (also known as elements, components, and "widgets") are individual pieces of a user interface that perform a single function. Some examples are [links](https://balsamiq.com/learn/courses/intro-to-ui-design/links/), [buttons](https://balsamiq.com/learn/courses/intro-to-ui-design/buttons/), and [icons](https://balsamiq.com/learn/courses/intro-to-ui-design/ui-controls/). Even plain text can be considered a control, since its function is to describe or label something within the user interface.

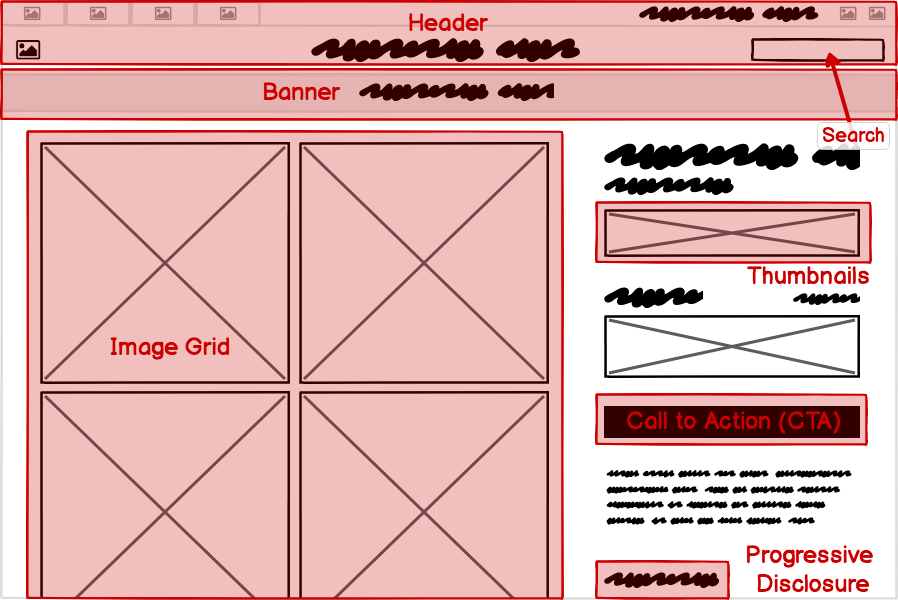
Each one of these controls was selected for a specific reason. **UI design is concerned with the process and rationale of choosing controls**

## **UI design layer 2: Patterns**

We can further simplify this page by decreasing the fidelity of our wireframe to abstract away the individual controls, like this:



Now let's think about the groups of controls and what purpose they serve as units within the page. A UI Pattern is a group of controls that function to solve a particular problem. Let's look at some of the patterns on this page:



It can be useful to consider this layer of UI design even before moving on to the level of controls, as each pattern can meet its goals in different ways and using different controls.

## **UI design layer 3: Design principles**

The most commonly understood definition of UI design is the visual design layer. But even this is more purposeful than most people understand. Visual design isn't merely "making it look pretty." A better way to think of it is as the application of established **visual design principles**, many of which are rooted in scientific psychological, neurological, or physiological understanding.

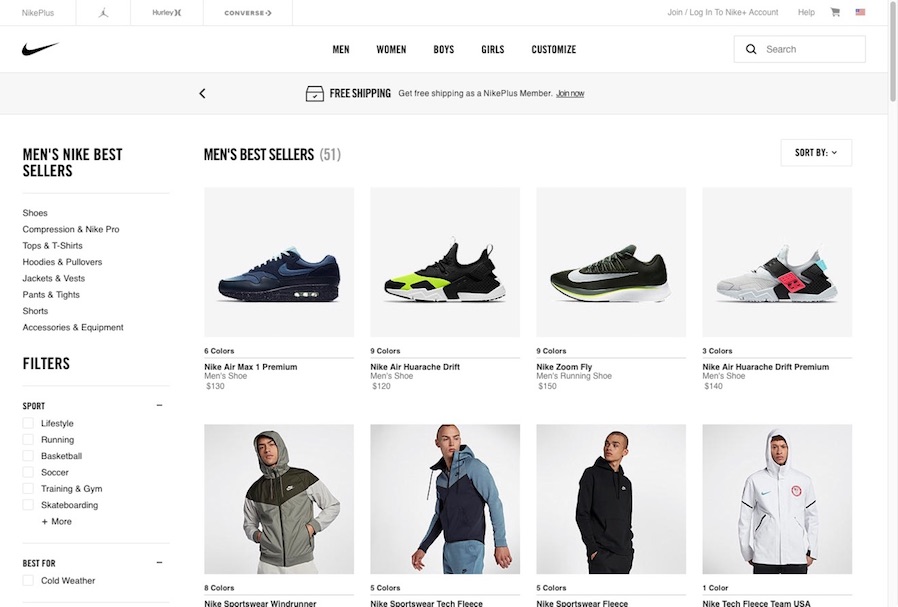
The specific principles we'll cover here are **Contrast**, **Hierarchy**, **Proximity**, and **Alignment**.

## **UI design layer 4: Templates**

Finally, looking at this site as a whole, we can view this page as an instance of a **template** that can be reused across the site, rather than a single page that was designed for this particular article of clothing. For a site or product that can have dozens, or even thousands, of screens, it is useful both from the designer/developer and the end-user perspectives to have screens that **behave predictably and look similar across the entire application**.

The example we've been looking at so far could be described as a "product detail view" template that would look very similar when any other product is viewed.

Another UI template is the category template, shown here:



**HTML**

HTML is where the magic of web page design begins.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

**CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

**Frameworks**

Frameworks are, as the name suggests, a frame that holds together code written by the user. In UI development and design, they create a base around which the rest of the project can be built.

Frameworks are found in a number of different development areas, and can take many forms:

* Programs that support the main design
* Code libraries
* Tool sets
* Application Programming Interfaces (API)
* Compilers

And a number of others.

What does all this mean? Simply that a framework can save designers and developers a lot of time!

**How does this relate to UI?**

In the world of UI design, frameworks usually come in the form of design programs. That is, software that assists you in building your own software projects.

(Some software, such as the Cocoa Touch, will aid you in designing specifically for one operating system – in their case iOS.)

The User Interface framework helps to speed the development of front end application.

This enables the developers to build beautiful, responsive multi-device application without deep technical and design skills.

Let us take an example of Bootstrap Framework (worked on it) created by Twitter developer and see the functionality it provides:

) It is open source and free to use.

2) Built on responsive 12-column grids Layout.

3) Styled and enhanced all the fundamental HTML elements.

4) compatibility for almost all the browsers.

5) customizable as per the designs of your project.

6) Holds a big support community.

Browsers:

If your software is browser based, you’ll need to ensure it works for every one on the market: Chrome, Firefox, Safari and so on.

While this CAN be done bespoke by every programmer, it’s one of those tasks that’s hugely time-consuming and fiddly, and can become a massive problem if you let it.

Modern frameworks will assist you in ensuring that your software works well across the board. In most cases, a huge amount of the testing has already been carried out.

Cross-testing can be brutally expensive in terms of man hours: don’t go manual or bespoke here unless you absolutely have to.