3. UI Design Patterns

What is a Design Pattern?

A design pattern is a reusable solution to a commonly occurring problem. As a recipe in cooking provides the ingredients and structure that make up a recognizable dish, so too does a design pattern provide an identifiable and predictable solution to an interface design problem.

The idea of design patterns initially came from architecture and programming, where the idea was to optimize solutions that are known to work well within given contexts.[1 (#1)] Solutions that emerged frequently enough became recognized as a formula that can be reused.

Structural and behavioral features of a pattern are familiar to users. Your team can leverage this knowledge, rather than re-inventing the wheel, to provide greater ease and use of their product. It's good to point out, however, that while design patterns are useful for informing design decisions around your particular problem, you may likely need to modify them around your users' and business' needs.

We're going to start by taking a look at a design pattern, look at a few examples of the pattern in use, and deconstruct their implementation. Then we'll list some common patterns for interface design and you may explore them in depth.

The Cooking Analogy:

In cooking we combine ingredients to prepare a dish. Let's say for instance, we're planning to make a *fish taco meal (https://www.youtube.com/watch?v=Ik94Fe6auDU&index=68)*. If you're familiar with this dish, you know that you'll usually prepare it with a flaky fish like Cod perhaps, tortillas, different seasonings, oil, salsa and maybe sliced lime for garnish. There are different ways to add to this dish to make it yours, but the basic combination of ingredients make a fish taco pretty unmistakeable.

This is very much like using design patterns. We have a general model for how to create this dish to make it recognizable. We can add or subtract from those ingredients and how they're put together to make it unique.

Elements of a Design Pattern

Design Patterns are typically written with a common set of attributes that looks something like this:

Design Pattern Model

- ***Pattern Name (and description)***
  - A label that provides a clear way to communicate and reference this pattern, particularly when discussing it with colleagues.
- ***Problem***
  - Describe what problem this solves and why this pattern exists.
- ***Context***
  - Describe when to use this solution.
- ***Solution***
  - How it works.
  - Describe the solution in detail.
- ***Recomendations***
  - Provide further recommendations.
- ***Examples***

Design Pattern Example

Let's explore writing a design pattern for a website Shopping Cart component into this model. This seems like an obvious description of a very familiar component.

While you're reading, think about how this compares to other purchasing experiences like a one-click purchase, or how it compares to a similar purchase of a service like a reservation or booking experience. Think about how this might be different on a mobile phone for example.

Pattern Name: Shopping Cart for Ecommerce Site

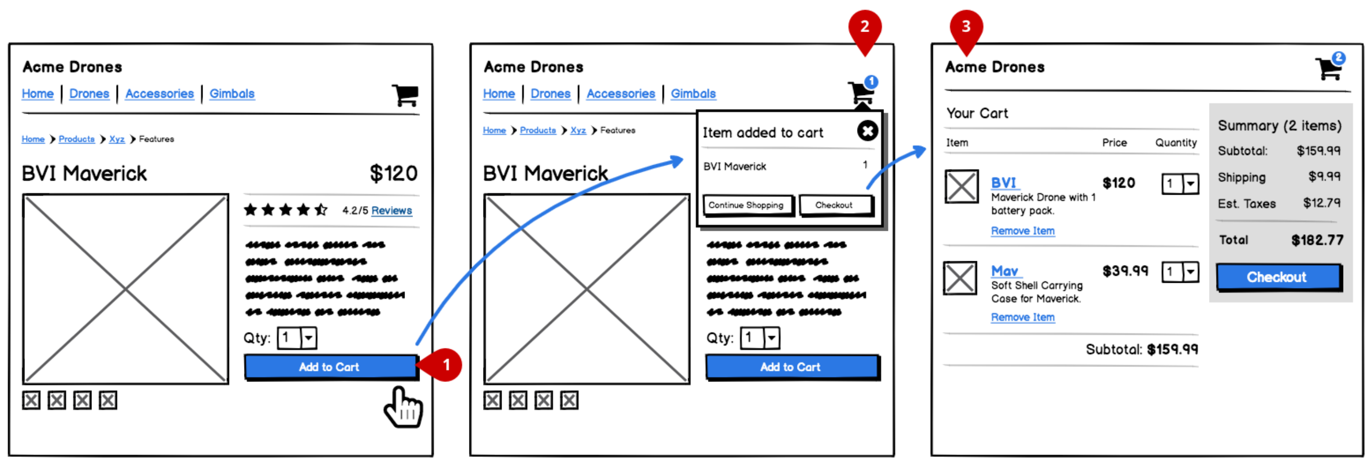
Description: The Shopping Cart component consists of 1) An "Add to Cart" button to purchase an item and 2) an accompanying cart icon used to indicate that the item is held for purchase and provides a link to view the items and begin checkout.

Problem: Users want to purchase an item in an ecommerce site.

Context of Use: Use this pattern when an online store allows browsing items, has more than a single item to purchase, or requires review of order before completing purchase.

While shopping on an online store the user may select items to purchase, but want to continue browsing, and may want to review and edit what they've selected before beginning checkout. This is similar to holding items in a shopping cart in the physical world.

Solution:



1. User selects *Add To Cart* button   2. Cart Icon updates and displays items in cart   3. Full Cart is displayed to begin Checkout

(//media.balsamiq.com/img/support/ui101/patterns/Shopping-Cart-Pattern.png)

1. ***Viewing Product and Adding an Item to Cart***
   1. Present a button with the product to add the item to a shopping cart.
2. ***Updating and Previewing Cart***
   1. When the user has selected the Add to Cart action, provide feedback that the item has been added. Show the item count increase in a numeric indicator next to the cart icon.
   2. Optionally show a preview of the cart with the item and the selected options displayed
   3. Provide feedback about the next steps, for example editing the cart, viewing the full cart, continuing shopping, or beginning checkout to complete the purchase.
3. ***Displaying the Shopping Cart***
   1. Provide a separate Shopping Cart View of the items added to cart so that the user may modify or complete their order.

2. Provide actions for editing quantities, removing items.
3. Provide an action for "checking out" or completing their purchase.

Recommendations:

- Add to Cart may be presented with options, for example a selector for quantity, selectors for style, etc. Provide conditional logic if the items requires options to be selected, for disable the Add to Cart button if style or size has not been selected.
- Consider a one-click option for signed in users, or navigate directly to the checkout process if the store offers a single item for purchase.

Examples:

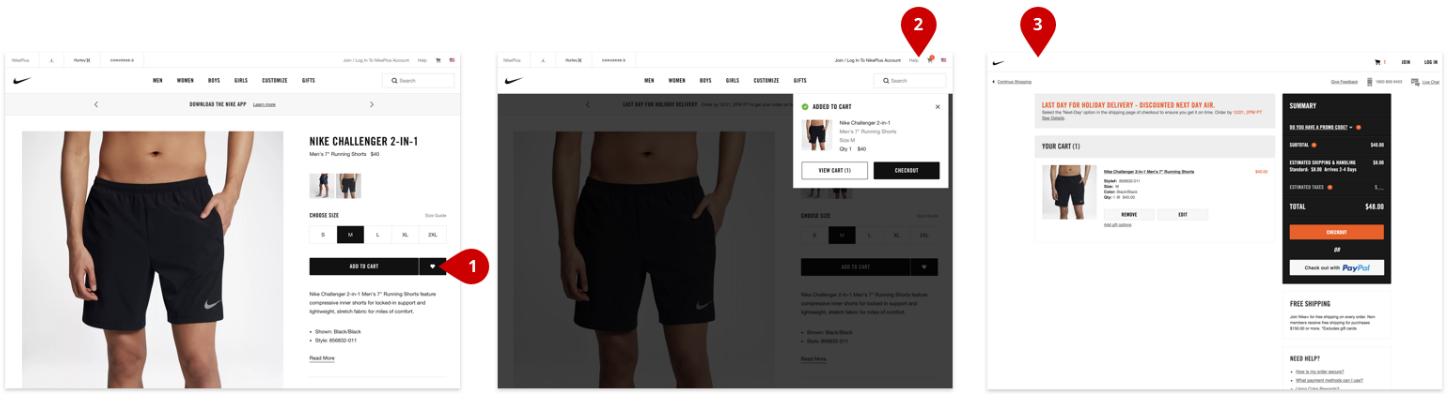
- Nike
- Shopify
- Crate and Barrel

Examples

Let's look at two of the Shopping Cart examples listed above and go a little further to deconstruct how they solved this specific need and how this reflects the pattern.

We'll look at the point where a customer decides to purchase a product using the Shpping Cart metaphor in two sites: the *Nike (https://store.nike.com/)* Store and *Tatt.ly (https://tattly.com/)* a site that uses Shopify.

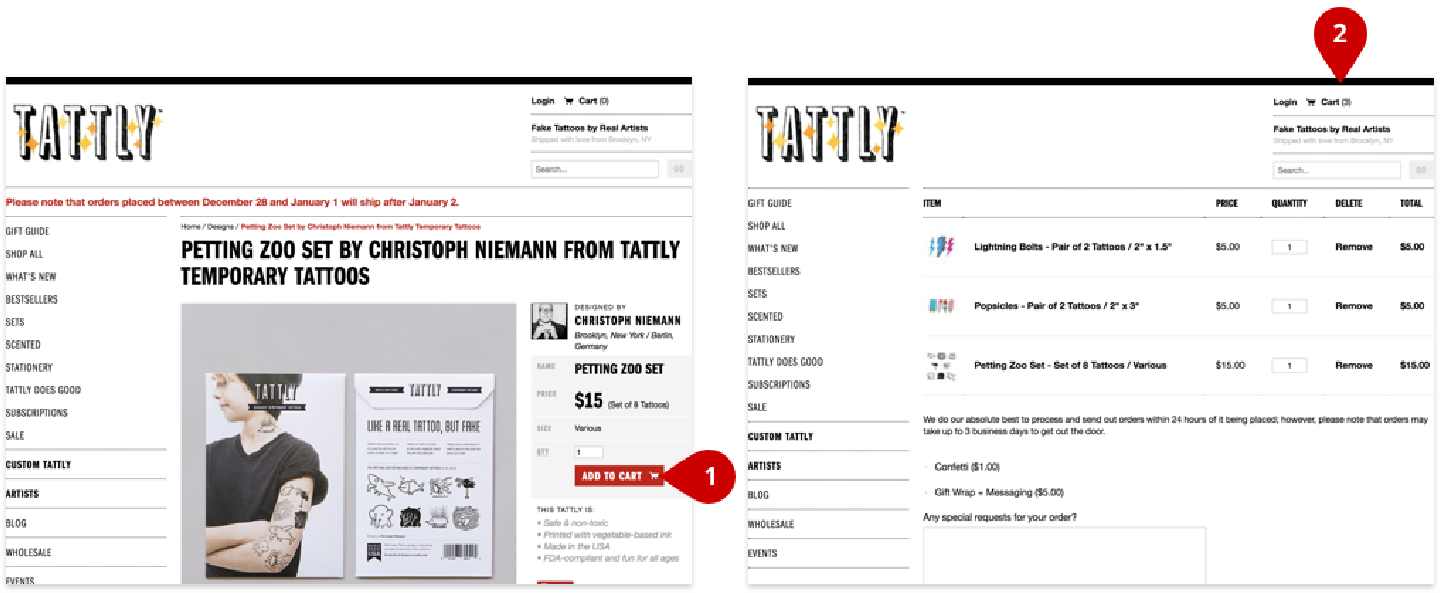
Nike.com cart.



1. User chooses size and selects *Add To Cart* button   2. Cart Icon updates and previews items in Cart   3. Full Cart is displayed to begin Checkout

(//media.balsamiq.com/img/support/ui101/patterns/Shopping-Cart-Nike.png)

Tattly cart, which use Shopify for ecommerce.


TATTLY
Login
Cart (0)
Fake Tattoos by Real Artists
Shipped with love from Brooklyn, NY
Search...
GO
Please note that orders placed between December 28 and January 1 will ship after January 2.
GIFT GUIDE
SHOP ALL
WHAT'S NEW
BESTSELLERS
SETS
SCENTED
STATIONERY
TATTLY DOES GOOD
SUBSCRIPTIONS
SALE
CUSTOM TATTLY
ARTISTS
BLOG
WHOLESALE
EVENTS
Home / Designs / Petting Zoo Set by Christoph Niemann from Tattly Temporary Tattoos
PETTING ZOO SET BY CHRISTOPH NIEMANN FROM TATTLY TEMPORARY TATTOOS
DESIGNED BY
CHRISTOPH NIEMANN
Brooklyn, New York / Berlin, Germany
NAME PETTING ZOO SET
PRICE $15 (Set of 8 Tattoos)
SIZE Various
QTY 1
ADD TO CART
THIS TATTLY IS:
• Safe & non-toxic
• Printed with vegetable-based ink
• Made in the USA
• FDA-compliant and fun for all ages
1
2
Cart (3)
We do our absolute best to process and send out orders within 24 hours of it being placed; however, please note that orders may take up to 3 business days to get out the door.
Confetti ($1.00)
Gift Wrap + Messaging ($5.00)
Any special requests for your order?

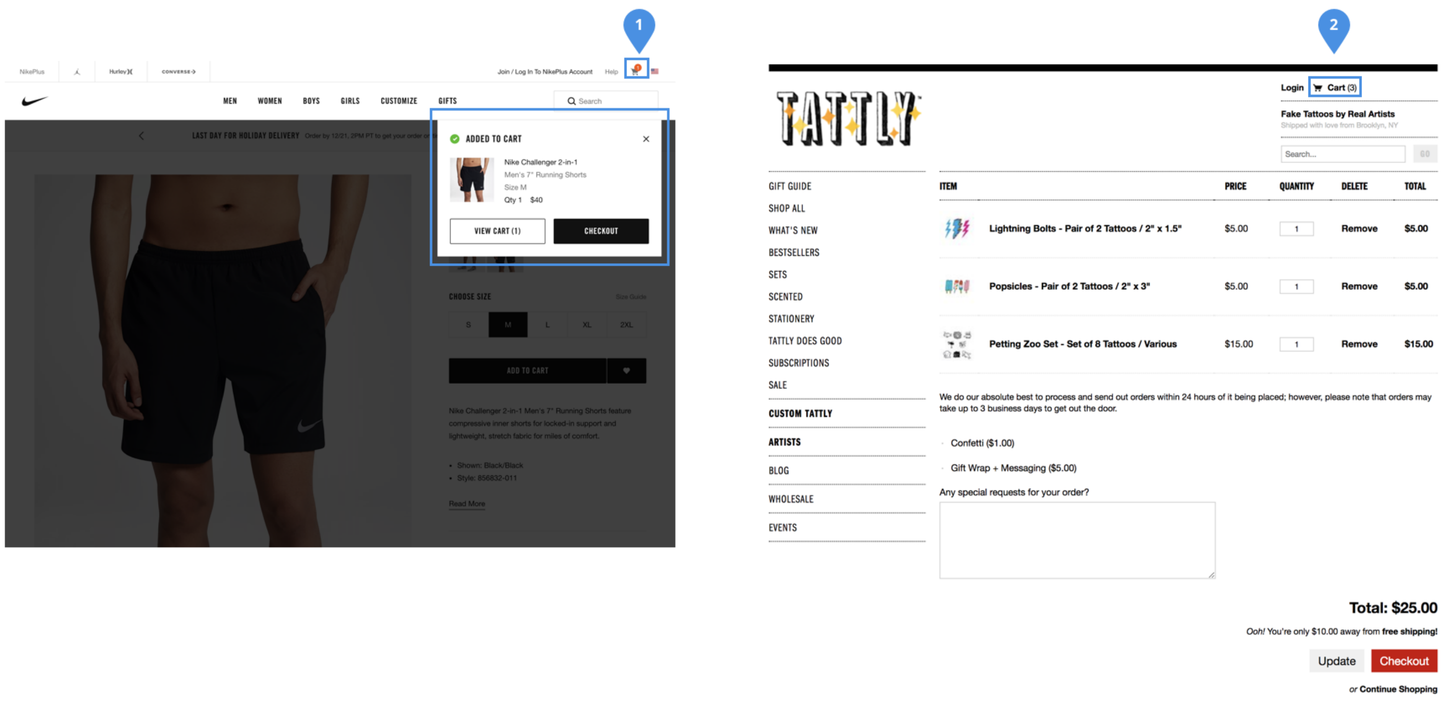
ITEM	PRICE	QUANTITY	DELETE	TOTAL
Lightning Bolts - Pair of 2 Tattoos / 2" x 1.5"	$5.00	1	Remove	$5.00
Popsicles - Pair of 2 Tattoos / 2" x 3"	$5.00	1	Remove	$5.00
Petting Zoo Set - Set of 8 Tattoos / Various	$15.00	1	Remove	$15.00

1. User selects *Add To Cart* button **2. Cart Icon updates and displays Cart to begin Checkout**

[(//media.balsamiq.com/img/support/ui101/patterns/Shopping-Cart-Tattly.png)]

Comparison

In both of these examples, the design pattern is based on using the shopping cart as a metaphor for temporarily holding things you are going to purchase. There's a common iconography that's been established, there's usually an indicator of the number of items being held, and an assumption that the next step is to "check out" as one would do in the physical world at a store.


1
2
NikePlus
Hurley
CONVERSE
Join / Log In To NikePlus Account
Help
MEN
WOMEN
BOYS
GIRLS
CUSTOMIZE
GIFTS
Search
LAST DAY FOR HOLIDAY DELIVERY
ADDED TO CART
Nike Challenger 2-in-1
Men's 7" Running Shorts
Size M
Qty 1 $40
VIEW CART (1)
CHECKOUT
CHOOSE SIZE
S
M
L
XL
2XL
ADD TO CART
Nike Challenger 2-in-1 Men's 7" Running Shorts feature compressive inner shorts for locked-in support and lightweight, stretch fabric for miles of comfort.
• Shown: Black/Black
• Style: 856832-011
Read More
TATTLY
Login
Cart (3)
Fake Tattoos by Real Artists
Shipped with love from Brooklyn, NY
Search...
GO
GIFT GUIDE
SHOP ALL
WHAT'S NEW
BESTSELLERS
SETS
SCENTED
STATIONERY
TATTLY DOES GOOD
SUBSCRIPTIONS
SALE
CUSTOM TATTLY
ARTISTS
BLOG
WHOLESALE
EVENTS
We do our absolute best to process and send out orders within 24 hours of it being placed; however, please note that orders may take up to 3 business days to get out the door.
Confetti ($1.00)
Gift Wrap + Messaging ($5.00)
Any special requests for your order?
Total: $25.00
Ooh! You're only $10.00 away from free shipping!
Update
Checkout
or Continue Shopping

ITEM	PRICE	QUANTITY	DELETE	TOTAL
Lightning Bolts - Pair of 2 Tattoos / 2" x 1.5"	$5.00	1	Remove	$5.00
Popsicles - Pair of 2 Tattoos / 2" x 3"	$5.00	1	Remove	$5.00
Petting Zoo Set - Set of 8 Tattoos / Various	$15.00	1	Remove	$15.00

[(//media.balsamiq.com/img/support/ui101/patterns/Examples.png)]

There are only subtle variations between these experiences.

1. Nike provides a nice behavior to preview the cart without leaving the product page, perhaps making it easier for the user to return to shopping. In another site with a similar dialog to Nike's Crate and Barrel uses this same idea of previewing the Cart, but additionally promotes other products the customer might want to purchase based on what's in their cart. Nike promotes other products on the cart view itself.
2. Shopify directs the user to the Cart immediately after adding a product to cart.

In both Nike and *Tatt.ly (https://tattly.com/)* (Shopify), there's the same general behavior and structure of the experience. Instead of re-creating the experience, for the most part, both of these sites rely on predictable conventions, which make up the common shopping cart pattern.

Using and Creating Your Own Patterns

It's likely that many of the interfaces you see were probably designed with a common design pattern in mind. The Cart example is one that is copied often because it's based on an understanding leveraging the recognizability of this pattern. It puts users at ease, because they have expectations for how online shopping works, and it satisfies business needs to make the experience as frictionless as possible, while also looking for valuable add on sale opportunities.

As you begin to use design patterns, remember that while they are great for helping you inform your design decisions when solving a common UI problem, they're not meant to be copied without thinking about your users' and product's particular needs.

Jennifer Tidwell, who wrote an excellent interface design book titled *Designing Interfaces (https://www.amazon.com/gp/product/1449379702/),* gave this word of advice about using patterns in the section called "About Patterns."

"*They aren't off-the-shelf components;* each implementation of a pattern is a little different from every other. They aren't simple rules or heuristics either. And they won't walk you through an entire set of design decisions..."

You'll find many potential solutions for your own design problem in our *Design Pattern and User Interface Galleries (/learn/inspiration/ui-galleries/)*. You can also find pre-built interfaces for many of these patterns in *Wireframes to Go (https://wireframestogo.com/)*.

Further Reading

- ["Elements of a Design Pattern" by Jared Spool (https://articles.uie.com/elements_of_a_design_pattern/)](https://articles.uie.com/elements_of_a_design_pattern/)

1: More about the origins of design patterns: A design pattern in architecture and computer science is a formal way of documenting a solution to a design problem in a particular field of expertise. The idea was introduced by the architect Christopher Alexander in the field of architecture, and has been adapted for various other disciplines, including computer science. An organized collection of design patterns that relate to a particular field is called a pattern language.

By *Mike Angeles*

Got questions or feedback? Email *mike@balsamiq.com (mailto:mike@balsamiq.com)*.

← 2.15. Icons (/learn/courses/wireframing/controls/icons/)

4. Design Principles → (/learn/courses/wireframing/principles/)