Chapter 6

Case Study: An E-Commerce Site

In the previous chapter we looked at workflows for various kinds of applications. The examples were purposely rather small, so we could cover them in a reasonable amount of time. In this chapter, we’ll walk through a bigger, more complex application in which we’ll encounter different kinds of problems. Here you’ll come across real-world situations and learn how to solve them in Flash Catalyst.

Understanding the Project

You’ll be creating a commercial web site where visitors can buy photos. Development tasks include making items display in a list that lets viewers take a closer look at specific photographs. But the site should be more than just a web shop, so you’ll also need to create a page with background information about the photographer, another with contact information in case potential buyers want to ask questions, and of course, a general home page. On the design side, because the pictures will be of nature scenes, the entire site should have elements that create a nature theme.

This application may seem no bigger or more complex than the previous examples, but don’t be fooled—some elements will become headaches if you’re not careful. And most often, developers make their biggest mistake before they’ve even started the project—they start thinking in code straightaway. Take special care to avoid doing so. You need to maintain a “helicopter view” of the project. You have to detach yourself from what think you’ll need to do to make it work and just look at the functionality required.

If you were the site designer, early on you might be thinking about design elements—nice screen transitions, a cool new logo, a certain font to use, or a blend mode to employ. Likewise, as a developer, right off the bat you’re probably thinking about the implementation—data grids, combo boxes, and input fields. But the problem isn’t entirely clear yet, so it may turn out that your ideas aren’t the best solution.
I know this because I've been there myself and I've worked with lots of other developers who fall into the same trap. There’s no shame in that, because you tend to solve problems from what you’ve learned in the past. But sometimes you need to change the way you work to find a better solution. This will also lead to you being a better developer and learning new things you can use in later projects.

Analyzing the Project

When you start examining the project to determine its needs, most likely you immediately try to get a high-level technical overview. But your analysis should include another part that should impacts the design and implementation—the target audience. An application that will be used day in day out by professionals is certainly going to look completely different from one that has a wide range of visitor profiles. Our application fits the latter category, so it must be accessible to people that fit a variety of profiles. So, it needs to be easy to navigate, visitors must immediately understand what the application is for, and the functionality should be limited to just what's necessary.

Once the concept of the application is clear, you can start thinking in somewhat more detail. We’re still talking about implementation at a high level, though, not how we’re going to realize our goals in code. And even with design elements, discussions should stay at a high overview level—the actual design will emerge from what we determine at this stage.

By now you may be asking, “How do I force myself to take a helicopter view of the project?” The simplest approach is to employ pen and paper—just write down what functionality you think you need to include. If what you’re writing concerns technical issues, eliminate it from the functional analysis and move it to the technical analysis. Another helpful approach is to place yourself in the position of the site user and ask, “What do I, the user, want?” Customers don’t usually care about how features are implemented under the hood; they just want the functionality.

A mind map can also help. This is basically a graphic that shows the links among concepts. Again, write down everything concerning functionality that comes to mind, but do so in a tree-like structure that shows how thoughts link to one another. The result may contain technical specifications and implications, but that’s okay. You just extract them when making your functional analysis. Mind mapping is becoming quite popular, though the maps can become complex and hard to interpret later. There are other techniques as well, but delving into all of them would just take us too far off topic.

Functional Analysis

On a general level, you need to create a site that consists of two columns. The left one will hold the navigational and optional components relevant to the page being shown. The right side will hold the actual content. Only one page will be visible at a time.

Since the home page is the first page visitors see, it should give a clear overview of what the site’s about. There needs to be a section on what visitors can find and another on where to go next. The page should also be visually attractive, so it requires more that text alone, which would be boring. Visitors should also see a direct link to the page containing the photos for sale. The second page will have a short bio of the photographer, including his status as an instructor, but no picture is necessary.

The third page will contain the crux of the site—the photo gallery. Pictures must come from a database to make for easy extensibility when more are added. Each picture should be accompanied by a price tag that’s always visible, as well as a details button. When clicked, it will produce a bigger version of the photo plus some information about where the shot was taken. As a design constraint, this has to be done in a pop-up window that overlays the gallery. After visitors log in, they’ll be able to directly add a photograph to their shopping carts.