Stephen Freiberg

Summary assessment from user's perspective

The upfront usability is much improved from that of the last project. The user is able to determine how to use the application without any README or other instructions. There is also a persistent navigation bar that indicates login status as well as provides links to the rest of the application. The application is geared toward ease of use with an emphasis on robustness through the different paths that a user may take. For example, carts persist based on both IP address and login (so that a user may have a persistent cart without giving any login information).

Summary assessment from developer's perspective

From the developer's perspective, the code here is modular and well factored. The routing file is probably not utilizing the power of Rails, but it's effective. For a system administrator that is using this application to stand up a webstore for his brick-and-mortar store, the setup is very easy and intuitive.

Most and least successful decisions

The most successful design decision was to attach order requests to Cart objects rather than Users or Sessions. This meant that a Cart could be attached to a Session or to a User, and order requests could be transferred between them. It also allows the future extension of allowing multiple carts per user. The least successful design decision was lack of validation on email addresses, since the system assumes that the user-email will be a valid field capable of receiving messages.

Analysis of design faults in terms of design principles

Because there was no clear "best way" of dealing with inventory amounts, this system ignores the problem as much as possible. When a user puts an item in a cart, it is not "reserved" in any way. Likewise the inventories are not changed when a user checks out (places an order). It is assumed that the end customer would specify the desired behavior for inventories. There is also no action with payments when the user checks out simply because it is outside the scope of the class.

Priorities for improvement

First priority for improvement is the user interface, because there is no use to a website if there are no users for it. The TwitterBootstrap styling makes this site more accessible, but there are definite improvements to be realized in the item library and shopping cart sections. Email validation would also be an improvement, and eventually tracking of user preferences based on prior orders. When the item library is big enough, a search bar would be good to add as well.

Michael Phox:

Summary assessment from user's perspective

At the time of this writing, the project has many links that are not pointing anywhere on the Heroku deployment. The root directory points to a "We're sorry" error page, and many of the links also point to error pages. I was able to view some of the forms by searching through the routes.rb file for what **should** be live links. The forms are all simple text boxes that link to other locations, so they are effective, though not visually appealing. It appears that the session information is not being handled properly (login does not persist between pages), and the inter-controller interaction appears to be buggy. Creating and destroying items does not appear to work, so it is not possible at this time to test the cart semantics. As a user, I would not have the patience to use this site simply because the links between actions are broken.

Summary assessment from developer's perspective

As a developer, the design documentation online indicates a well thought-out and careful object and event model. The wireframes also indicate the control flow that I would expect from the shopping cart, with the exception of requiring a user to sign in to shop, and requiring input of a username / password pair to log out. The only problem that I see (from a security standpoint) is the "create admin user" link. I believe that with small changes to the routing file, this would be a usable shopping cart application.

Most and least successful decisions

The most successful design decision here was to keep the Cart object separate from the User object. This allows future extensions to include having multiple Carts per User, as well as Wishlists, etc. The least successful decision was the lack of attachment of Item to Order. Without this connection, it seems difficult to determine the effects of an order (which item was requested? What happens if the item is out of stock?).

Analysis of design faults in terms of design principles

The design appears modular and robust, but I am concerned that there is no "finalized order" object that is displayed to the administrator. How is the administrator notified of past orders?

Priorities for improvement

First priority would be to fix the control flow so that the application is smoothly usable by customers. Next would be to spruce up the user interface so that the application is more appealing and therefore more likely to be used, and remove the security hole of "create a new administrator account".