

Design Analysis: AZCart, a Shopping Cart Plugin angelaz

NOTE: I took 2 slack days for this project.

OVERVIEW

AZCart is a lightweight web application that simulates a shopping cart experience online. AZCart provides a similar service as Opencart, Shopify, Paypal, and Stripe, but it fulfills the shopkeepers' need for a more basic and minimal setup effort required product.

FEATURES

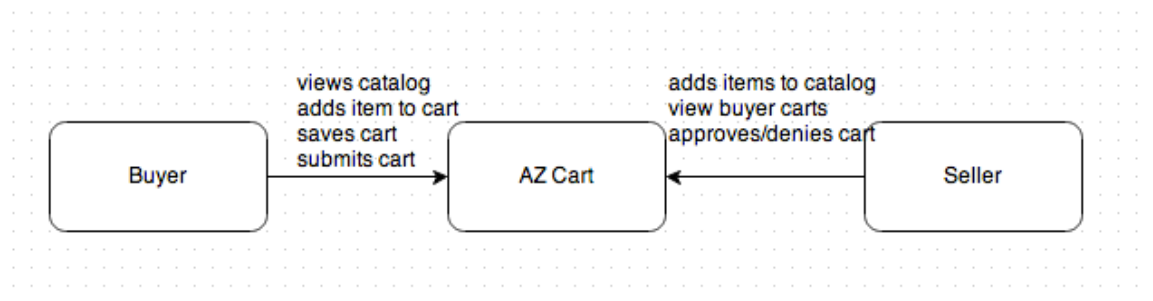
When a shopper is browsing a website, they can

- 1) create account/log in/log out
- 2) add items to the shopping cart
- 3) remove items
- 4) edit quantity of an item
- 5) checkout items in the cart
- 6) retain the shopping cart across browser sessions
- 7) share shopping carts with other users

For a shopkeeper, AZCart will provide them a simple admin interface to:

- 1) add items to the catalog
- 2) edit a catalog item's title, price, and quantity
- 3) approve or decline orders submitted by a user

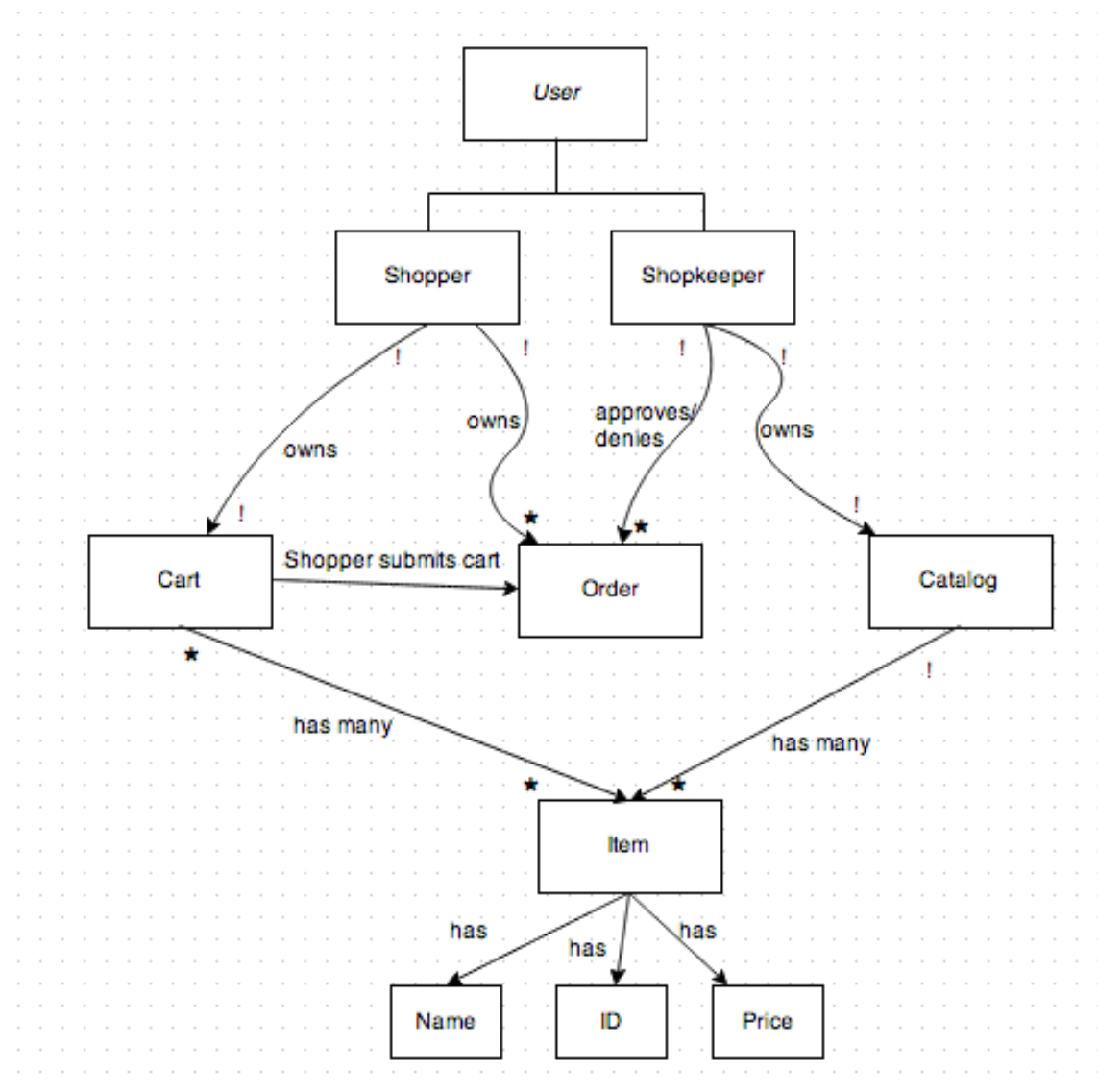
CONCEPTS



BEHAVIORS

Buyer can add browse catalog and add items to cart before logging in, however, they must log in first to submit a cart for approval by the seller. Buyers cannot post an item for sale.

Sellers all manage the same catalog (think a single store with multiple managers/sales clerks). Once logged in, sellers can create items in the catalog, view all the carts (submitted or not) in the system, approve/deny carts submitted by buyers. Sellers cannot purchase items (since they are all managing one catalog that is their own).



UPDATES (3/12/13):

- Shopper now owns multiple Carts, each with a "status" tag (-1= denied, 0 = unsubmitted, 1 = submitted, 2 = approved).

- Order class is removed from the design
- The website holds a total of 1 Catalog, but may have multiple Shopkeepers/Sellers who share the catalog and have the same set of permissions.
- When an Item is added to a Cart, a CartItem is created with a foreign-key into the Cart. Each Cart holds multiple CartItems.

UI

Home Page - Logged Out:

Catalog Items

Name	Price	Add to Cart
Apple	10.0	Add to Cart
Banana	20.0	Add to Cart
Cactus	30.0	Add to Cart

[Sign up](#) or [log in](#)

Home Page - Seller:

Catalog Items

Name	Price	Edit
Apple	10.0	Edit
Banana	20.0	Edit
Cactus	30.0	Edit

Submitted Orders

Cart ID	Buyer Email	Show & Approve
4	buyer2@mit.edu	Show

Approved Orders

Cart ID	Buyer Email	Show
1	buyer1@mit.edu	Show

Denied Orders

Cart ID	Buyer Email	Show
3	buyer2@mit.edu	Show

[New Item](#)

Logged in as seller3@mit.edu, a Seller. [Log out](#)

Cart Page – Buyer (buyer clicks “View Cart” to view the current cart or one of the saved/submitted/approved carts:

Items In Cart

Buyer: buyer2@mit.edu

Status: Unsubmitted

Name	Price	Quantity	Edit	Delete
Apple	10.0	2		

[Continue Shopping](#)

[Place Order](#)

All buttons take the buyer back to catalog page.

Design Concerns

Below is a list of potential challenges:

- Access control.

Allowing users to create account, log in, and log out. Retaining the cart across browser sessions, specifically, from not logged-in to creating new account or logging into existing account.

- I decided to let unregistered buyers view the catalog and add items to cart, but they must log in to purchase. One buyer cannot view another buyer's carts.
- As for seller, I decided that they have all permission to view buyer's carts. This decision makes seller ubiquitous, which, in a production environment, should be rethought. For example, sellers probably shouldn't be able to edit those carts owned by buyers.

- Catalog page layout.

I decided to have all the lists (the entire catalog, all submitted/approved/denied carts, saved carts for buyer) on one page. This is because I think it would be best for users to view everything with a minimum required # of clicks.

- Lack of order item.

Currently, if the seller updates an item, this item will also be updated in all submitted and approved carts. In retrospect, this is a poor design decision. I should have frozen the item price/title once the cart is submitted by the buyer.