Sarah Edris
03/03/2013
6.170

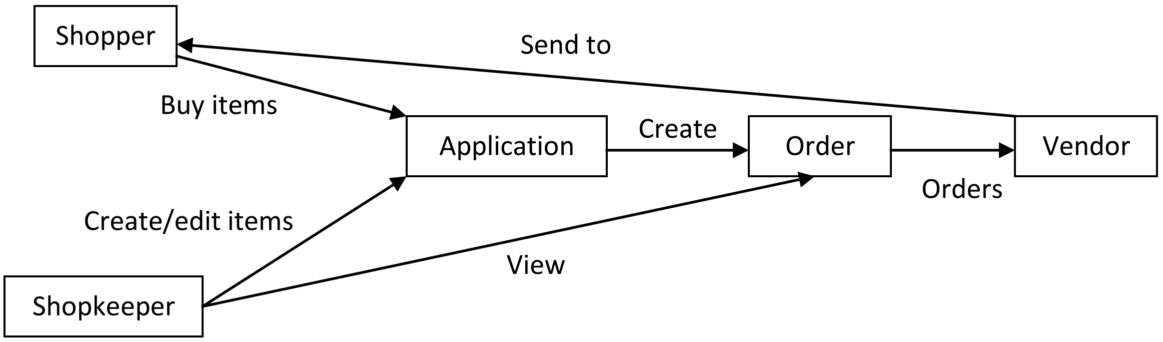
Project 2.1: Shopping Cart

Overview

Purpose and Goals

This system is a shopping cart application, with separate shopper and shopkeeper interfaces. It allows for the shopper to add items to a cart and checkout, and for the shopkeeper to edit items and view recent orders. This is meant to be a simple and easy to use system, with only the necessities of a shopping application. It is designed to be easy to extend, specifically in a later part of this project. It is also a way to become more familiar with Ruby on Rails, authorization, authentication, sessions, and cookies.

Context Diagram



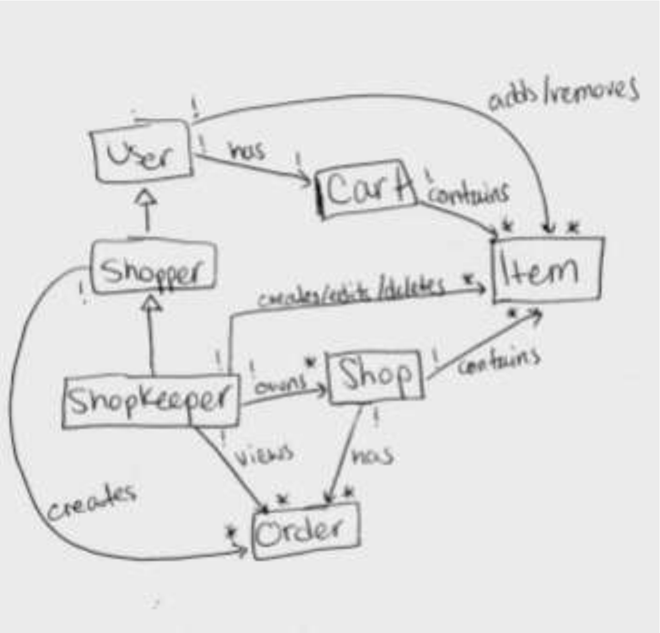
Concepts

Key Concepts

The key concepts are:

- A **user** is a person on the site that may put items in a shopping cart
- A **shopper** is a user of the application that can purchase items
- A **shopkeeper** is a user of the application that can sell items and edit their information, as well as purchase items
- An **item** is any good that is sold through the application
- A **cart** is a collection of items a shopper might buy
- An **order** is a collection of items that a shopper has decided to buy

Object Model



Behavior

Feature Descriptions

The following features are provided:

- **Signing up/Logging in:** The currently logged in user is saved using a cookie. Shoppers do not have to initially create an account, however shopkeepers must.
- **Shoppers putting/removing items from a cart:** A shopper can add and remove items from his cart. The shopper does not have to be a registered user to do this, but must be a user to place an order. Shopping cart information is saved in a session.
- **Shoppers placing their order:** A shopper can place an order from the items in his cart.
- **Shopkeepers creating shops:** Shopkeepers can own more than one shop to sell items from.
- **Shopkeepers editing item prices:** A shopkeeper can edit item prices. A shopkeeper must be a registered user.
- **Shopkeeper viewing orders:** A shopkeeper can view orders placed in their stores.

Security Concerns

User account protection is one concern, so all users that with an account must have a username and password, which is hashed on the server side. There are potential security threats with this though:

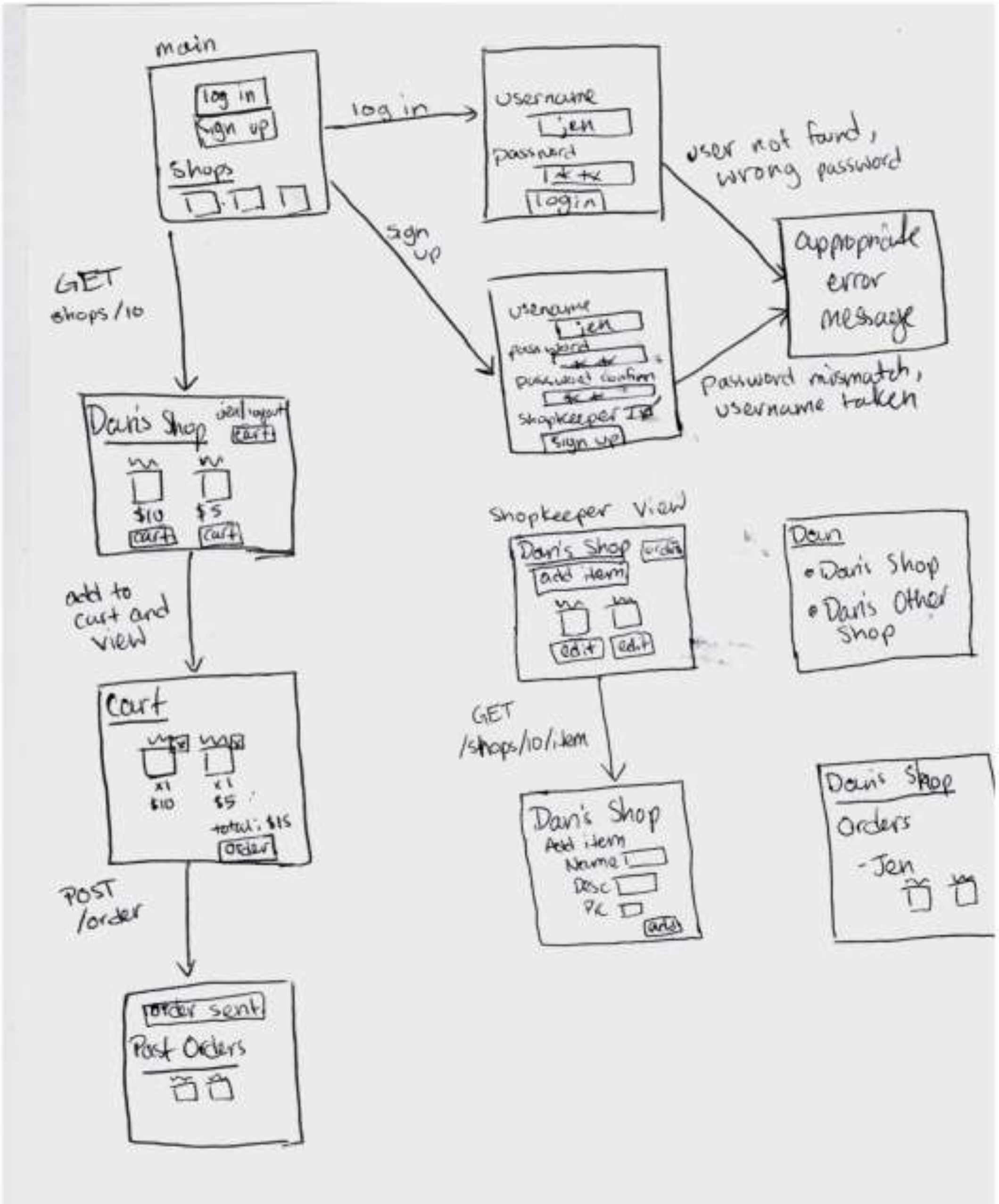
- **User not logging out**: While keeping the user's logged in state throughout browser closing and openings is convenient for the user, it could be dangerous if someone else were to use their computer while they were still logged in and bought items on their account. Because of this, a 'log out' button will be displayed prominently, and suggested to be used for safety if sharing the computer.
- **Session Fixation:** An attacker may fix a known user session id, and force the user's browser into using this id. To counter this, a new session identifier is issued after a successful login, and the old one is declared invalid by using the "reset_sesion" call in Rails.

Other more complicated forms of hijacking or attacking will be ignored in this application because of its small scale. Further security measures may be added in future versions.

User Interface

A user initially coming to the site with no account may view items and add them to a shopping cart. If they decide to place an order, they are taken to a sign up page, and then allowed to follow through with their order. A shopper with an account does not need to go through this process if they are already logged in. If a user attempts to log in and their log in information is not correct, an error display will pop up telling them this. Once an order is placed, the user may continue shopping and adding items to the now empty cart, or view past orders.

The shopkeeper has all of the same functionality as a user, but also owns shops with items. The shopkeeper can own multiple shops, and for each shop can view orders from other users.



Challenges

Design Challenges:

The challenges in designing this include:

- **Necessity of a “Shop” and “Cart”:** The shop and cart provide symmetric views of similar concepts. While the system would work if users simply selected items, and shopkeepers simply had items, separating it out provides modularity and the ability to extend this application. A user has the potential to have multiple carts, although this application will not do that. And a shopkeeper can own multiple stores, which the application does do.
- **Concept of shoppers, shopkeepers, and unregistered users:** An easier to implement design would be to force any user of the site to log in; however that isn’t very user friendly. To allow for ease of use, unregistered users (those not logged in) may begin shopping, however must log in to actually buy items. Shoppers and shopkeepers could also be separate entities, with a shopkeeper not being able to make purchases like a shopper. However, once again, this is not very user friendly or intuitive, so shopkeepers are also considered shoppers, and may make purchases.