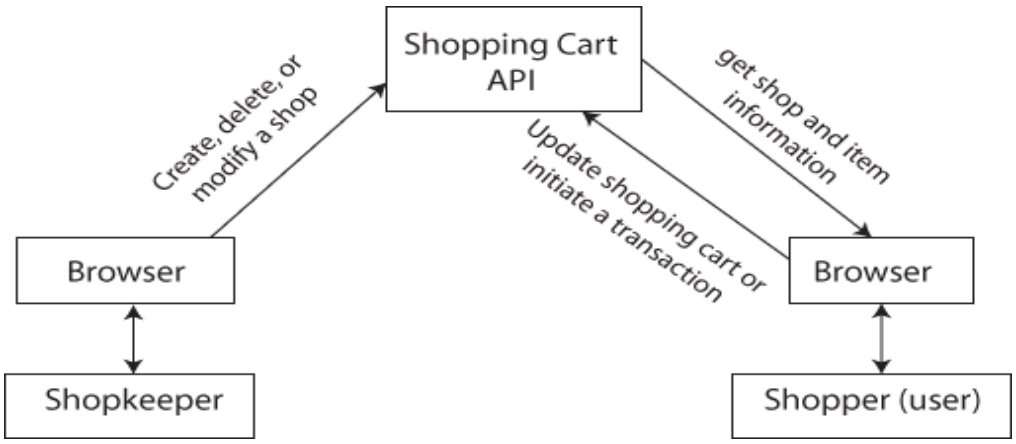
Overview:

In this project I am implementing a shopping-cart application that can be used by shopkeepers to sell items and by users to buy those items. This service is for people who want to create their own shops online with emphasis on their shop rather than on the individual items they are selling. This differs from Amazon in that, instead of being focused on helping users find the items they want, we help users discover shops they might like. This change in focus makes it easier for shops to build trust and rely on brand loyalty much more than currently possible with services like Amazon.

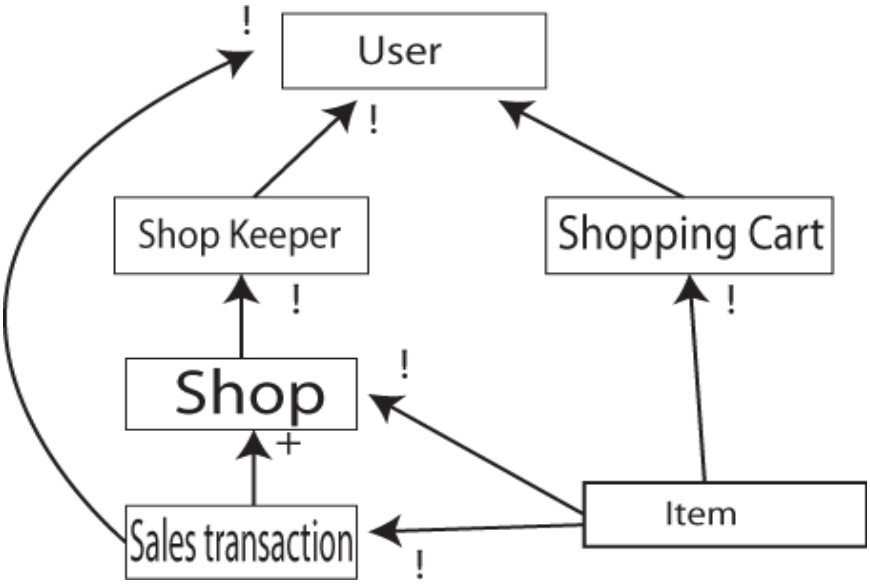
Context Diagram:



Concepts:

User—has own **shopping cart**.
Shopping cart—holds **items**.
Item—is sold by a **shop**.
Shop—has a selection of **items**.
Shopkeeper—owns and manages **shops**.
Order—is placed by a **User** and given to a **Shop.**

Object Model:



Behavior:

Feature Descriptions:

A user is created upon adding an item to the shopping cart if there is no user currently active (A user is said to be active if a valid session (in their cookie) exists). If the user is already active then we will add the item to their shopping cart. If the user is registered and logged in then their shopping cart will persist from their last usage. If a user has a session and then decides to login, then we will ask the user if he/she wants to merge the shopping cart tied to their account and the shopping cart from their session that exists when they weren't logged in. Upon visiting the site the user will see many shops, these shops can be clicked on and this page will lead to the shop's page which will have a list of items currently being sold. Upon submitting an order, the user's shopping cart is emptied and the order is submitted to the shop for its shopkeeper to either accept or decline.

Shops can be created by registered users. Registered users are automatically "shopkeepers" in that they have the ability to create a shop if they are so inclined. When a shop is created it must be given a name and an icon to be displayed next to it. There will be a preset of icons to select from, but you can also upload an image if you are so inclined. These images will be scaled to a preset file size for showing on the homepage. My motivation for requiring the image is for aesthetic purposes.

When registering a user is only asked for a username and a password. As of now there are no plans to implement a "forgot password" feature. User's may have multiple shopping carts (they can name them if they want), but by default they are called "<User's name || Anonynomous>'s Shopping Cart". There will be a "Browse Shopping Lists" page

that users can visit to see other user's shopping carts by name. They can then click on the name to see more information about the shopping cart.

Security concerns:

Problem: Storing passwords

Option 1: Store passwords in plain text. This option was not chosen because if a hacker were to somehow break into my database then they'd be able to steal password information

Option 2: Hash the passwords. This option was chosen because even if a hacker were to somehow break into my database then they would not be able to steal any password information. The only con is that we have to hash a user's password every time they login which is not a big performance concern.

Problem: Accessing resources specific to a user (Shop management, item management, and list management)

Option 1: Use a URL with a special hash to the resource (making it impossible to guess). This option was not chosen because a hacker could steal browsing information from a user and use this to modify the resource.

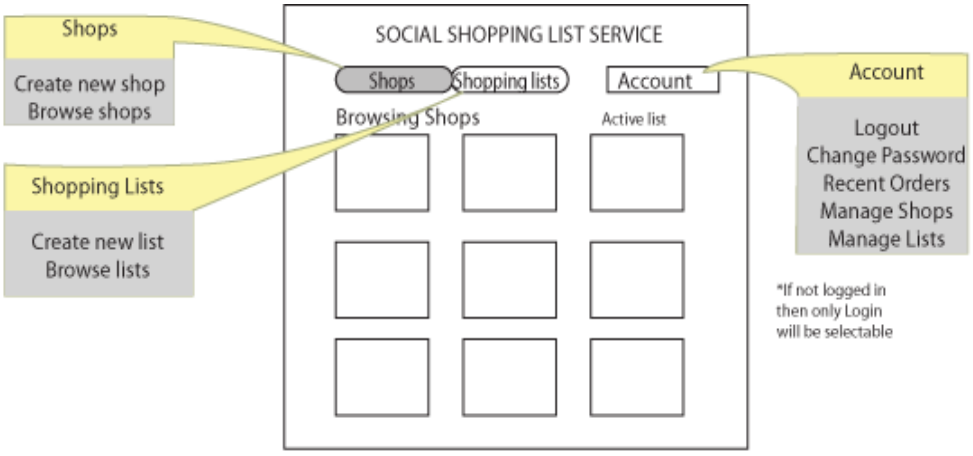
Option 2: Use a normal route with no special URL that uses the user's session to determine what information the user can access. If the user is not logged in then they will be redirected to a 404 page if trying to navigate to the resource modification page.

Problem: Bandwidth. Letting users upload images can consume a lot of resources

Option 1: Scale the images down and lower the quality when uploading. This option was not chosen because I don't know how to implement it.

Option 2: Let the users upload many images. This option was chosen because not only can I not implement the other option, but because I want to provide a quality service. It would be a disservice to the user if I lowered their picture quality.

Wireframes:


Shops
Create new shop
Browse shops
Shopping Lists
Create new list
Browse lists
SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Browsing Shops
Active list
Account
Logout
Change Password
Recent Orders
Manage Shops
Manage Lists
*If not logged in then only Login will be selectable


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Create New Shop
Shop name:
Upload image
Select Preset
Preview
OR


Modal
SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Browsing Shops
Create New List
Name:
Submit


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Recent Orders
Order #, View more(Shopping Cart link), Date, Amount
Order #, View more(Shopping Cart link), Date, Amount
Order #, View more(Shopping Cart link), Date, Amount
Order #, View more(Shopping Cart link), Date, Amount
Order #, View more(Shopping Cart link), Date, Amount
Order #, View more(Shopping Cart link), Date, Amount
Order #, View more(Shopping Cart link), Date, Amount
Order #, View more(Shopping Cart link), Date, Amount
<- Previous Next >


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Manage Shops
Shop Name
Shop Name
Shop Name
Shop Name
Shop Name
Shop Name
Shop Name
Shop Name
Shop Name


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Manage Lists
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete
List Name Modify Delete


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Managing Shop Name
Item Name, Quantity, Price Per Unit, Update Remove
Item Name, Quantity, Price Per Unit, Update Remove
Item Name, Quantity, Price Per Unit, Update Remove
Item Name, Quantity, Price Per Unit, Update Remove
Item Name, Quantity, Price Per Unit, Update Remove
Item Name, Quantity, Price Per Unit, Update Remove
Item Name, Quantity, Price Per Unit, Update Remove
Item Name, Quantity, Price Per Unit, Update Remove


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Managing List Name
New name:
Submit
Item Name Remove
Item Name Remove
Item Name Remove
Item Name Remove
Item Name Remove
Item Name Remove
Item Name Remove
Item Name Remove
Item Name Remove
Item Name Remove


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Updating Item Name
Item Name: Current name
Price Per Unit: Current price
Image: Current image link
Current description
Submit


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Browsing Lists
List name (clickable link)
List name (clickable link)
List name (clickable link)
List name (clickable link)
List name (clickable link)
List name (clickable link)
List name (clickable link)


SOCIAL SHOPPING LIST SERVICE
Shops
Shopping lists
Account
Change Password
Username
Old Password:
New Password:
Verify New Password:
Submit