**Q’s Spy Shop Website Design**

### Introduction:

### This is an e commerce website servicing customer requests for Q’s spy shop. The shop stocks spy gadgets and accessories. The website offers an online portal to display the catalogue and add products to a cart and order them.

### Purpose of the document:

### This is meant for the developers and the customers together.

### Scope of the Document:

### This document provides the essential requirements, from a high-level, for an spy equipment e-commerce website; this document includes the site flow, use cases, sequence diagrams, class diagrams, database design, as well as other implementation approach details.

### General Information:

### Q’s Spy Shop is a small to mid-size supplier of international spy gear for the contemporary spy on the go.

### Product Layout:

* Q’s Spy Shop website will consist of the following web pages:
  + the Home,
  + Landing,
  + Gallery,
  + Search,
  + Result
  + Product,
  + Functional,
  + Shopping Cart,
  + Check Out,
  + Payment,
  + Confirmation,
  + Shipping,
  + Tracking,
  + Location,
  + Contact and
  + Feedback pages.
* To better service the double and triple agents, Q’s Spy Shop will include a Check Out page that does not require membership.

**Product Requirements:**

* CPU
  + Intel Pentium 4 and above / Amd x2 and above
* Operating System
  + Windows XP or a later version
  + Mac OS X Leopard and above
* Supported Browsers:
  + Google Chrome
  + Apple Safari
  + Mozilla Firefox
  + Opera
* Internet Access

**Functional Requirements:**

* The Website should continuously update the inventory and the stock according to the corresponding sales.
* The website must list out the available categories of products on sale at all instances.
* The website must provide a search function to the customers to offer search by product name, type or popularity.
* The website must maintain a Shopping Cart at all instances which would allow the customers to add desired products into a collection and check out at desired time.
* The website control should allow online payment of the order using various methods to attract a larger customer base. Now we are considering Credit Card and PayPal only.
* The website must provide a customer with **Order Fulfillment** report in order to give feedback and a summary on the purchased goods.
* The website must make a note of all the activity performed by the system, typically in a database.
* The website should allow returning customers to easily place orders by providing retrieval of information using a login page.

**Non Functional Requirements:**

* Usability Requirements: The program should provide easy accessibility for the required tasks while the demands exist. With a few clicks the user can reach the desired checkout page in our project.
* Reliability and Uptime Requirements: No reliability issues would be encountered as the data entered is immediately displayed once on the screen before getting recorded. Also the final setup would involve physical server security and disaster recovery plan to ensure 99%+ uptime.
* Security Requirements: Since we use the ASP inbuilt security libraries, the system is already secure. The login system will protect it from unauthorized access.
* Performance and Scalability Requirements: The Q’s Spy shop application is optimized for a single server setup. But it can be highly optimized and scaled to multiple servers accepting a lot more requests than the current model does.
* Maintainability and Upgradeability Requirements: Making changes to the system would not be difficult as using programming many features could be converted by the system into newer counterparts.
* Supportability and Operability: The system would be supported during its entire lifecycle.

**Coding Standards adopted:**

For HTML CSS: The comments would be delimited between 🡨 and --!>

For ASP: ASP Application Directory Structure

The directory and file conventions suggested in this section optimize ASP application directory structure for consistency, clarity, and ease of use.

The directories and files described are listed here:

/Application\_Root

Default.aspx : It contains the basic page that would turn up in case of a dynamic page request.

Web.Config: It contains the configuration settings for the particular application.

Global.asa : Maintains globals around the project.

/Controllers: These folders contain Controller classes.

/Views: These contain the Views configurations.

/Images: The required images for the banner and nav content are stored here.

/Themes: Application themes can be stored here. They can be used from the web config declarations.

/Data: Contains the Database and logs.

Indentation: All the code is Left indented with using only tabs for nesting

Whitespaces: These will be included after every statement end.

Naming: The object/ variable names would always be in small case while class names would be always CamelCase.

**Website Design:**

The website contains the following main pages:

The Home: It is the welcome page for the shop. It lists the various available new products and navigational links to the categories available.

Gallery: This page will list the images of various products in action, for a better understanding to the customers.

Search: This page features the filters and text labels to be searched against in the product database.

Result: This page is responsible for listing out the results of the performed search listed in the order of the preference.

Product: This is a dynamically generated page dedicated to a product as a whole. It lists the product image, Description and reviews information.

Shopping Cart: This area of the website is responsible for keeping a tab on the number of products in the customer’s area of interest. It keeps a total of the quantities and rate of the products included in there.

Check Out: This page shows a final amount of products in the customers order and gives an option to checkout.

Payment: This webpage is responsible to take care of the payment method of the customer in order to successfully place the order.

Confirmation: This page is responsible for showing a receipt.

Shipping: This page is responsible for giving the user an option for selecting the shipping options.

Tracking: This page is responsible for giving the tracking number back to the customer according to the shipping method selected.

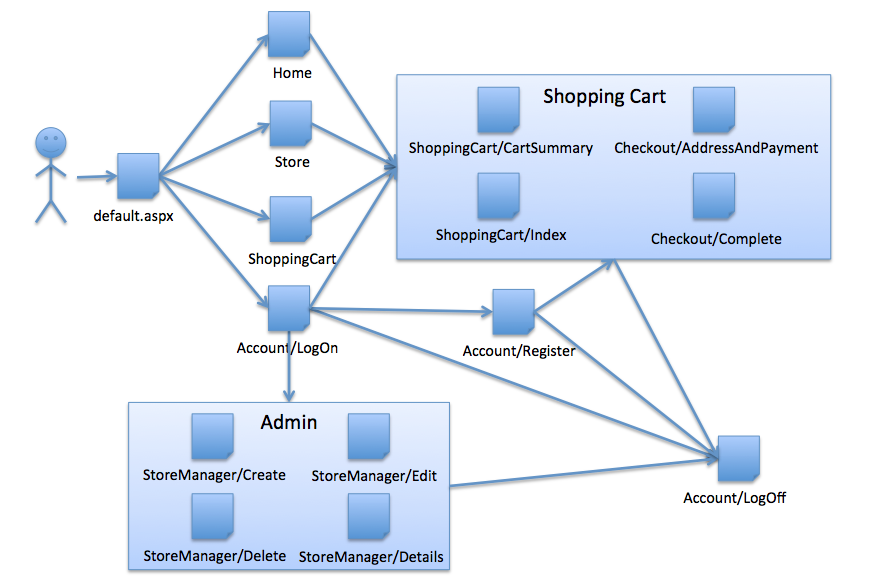
Location: The location page gives the information about the location of shipping to the customer.

Contact: The contact page lists the contacts of the distributors to be contacted according to the placed order.

Feedback: This page lists the option to provide a feedback to the distributors.

**Site Flow:**

The diagram shown below, Q’s Spy Shop Site Flow, illustrates the initially planned navigation of the site’s web pages. Furthermore, this diagram helps enumerate and solidify the high-level functional requirements of all parties involved.



(Q’s Spy Shop) Site Flow

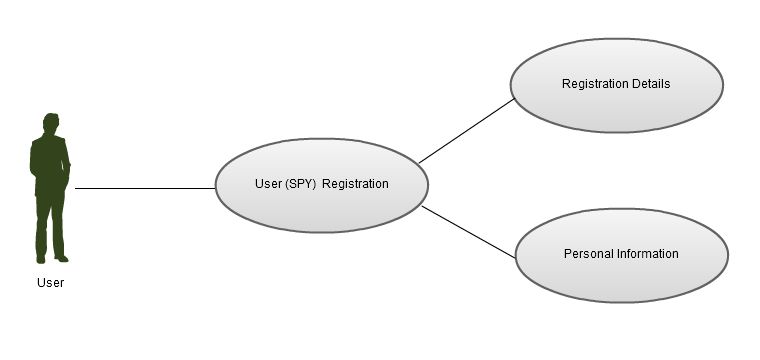
Once the site flows have been established, the high-level objects are more easily fleshed out. That is, the objects are elaborated with detailed descriptions of the associated actors, roles, and external system interactions. It is by this manner, or by utilizing Q’s Spy Shop Site Flow as a springboard; the following use cases are created:

Use Case: Register an Account

|  |  |
| --- | --- |
| Use Case: | Register and Account |
| Actors: | User |
| Type: | Primary |
| Purpose: | Allow a user (Spy) to register on the site thus enabling the upkeep of his/her information. |
| Description: | The blank page where a user can create account by entering personal information, for a more personalized experience. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) User clicks on the register button. | The Register page, with necessary fields, is generated. |
| 2) The user enters the required information and then submits the page. | The page verifies input and acts accordingly. |

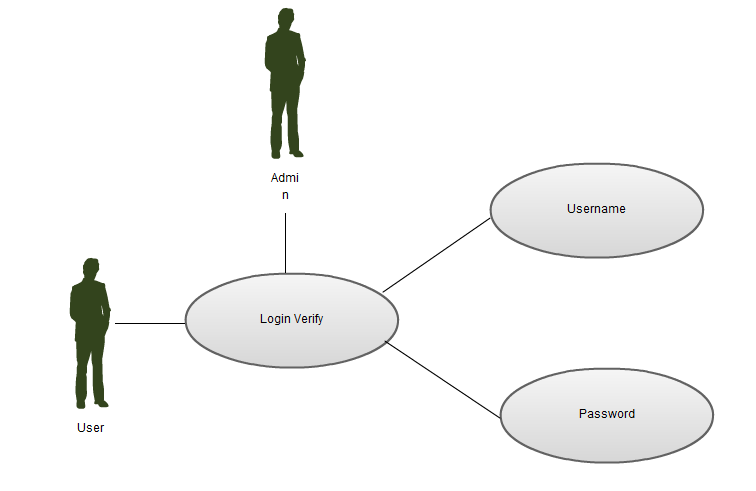


Use Case: Log in to Account

|  |  |
| --- | --- |
| Use Case: | Log in to Account |
| Actors: | User, Admin |
| Type: | Primary |
| Purpose: | This allows the Actors to see persistent settings and information. |
| Description: | A page shown to the actor where his/her identity will be verified. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) The actor clicks on the Account button. | The log in page is displayed. |
| 2) The actor enters their username and password. | The username and password is verified and then the user name is displayed indicating that they are logged in. |

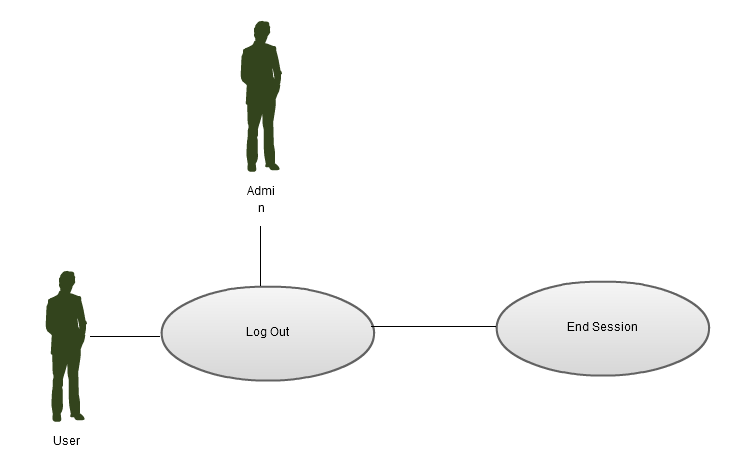


Use Case: Log out of Account

|  |  |
| --- | --- |
| Use Case: | Log out of Account |
| Actors: | User, Admin |
| Type: | Primary |
| Purpose: | End a session with the actor. |
| Description: | The actor, after opting to end their shopping, will be shown a confirmation that the session is ended thus conveying a sense of securely. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) The user clicks on the log out button. | The users session is terminated and their name is no longer shown. |

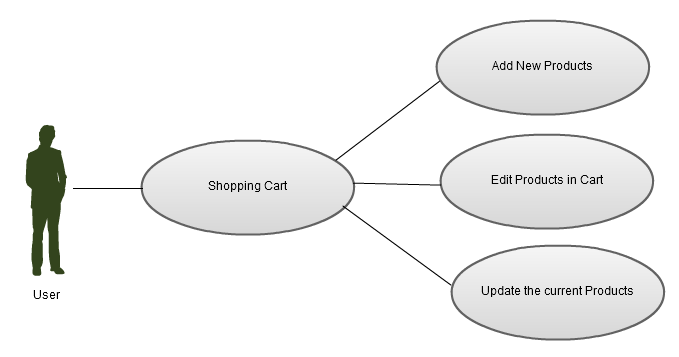


Use Case: Add to Shopping Cart

|  |  |
| --- | --- |
| Use Case: | Add to Shopping Cart |
| Actors: | User |
| Type: | Primary |
| Purpose: | The important ability to add items to be purchased. |
| Description: | The user will be shown an updated shopping cart and shopping cart item count. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) The user clicks the add to cart button. | The shopping cart counter is incremented. |

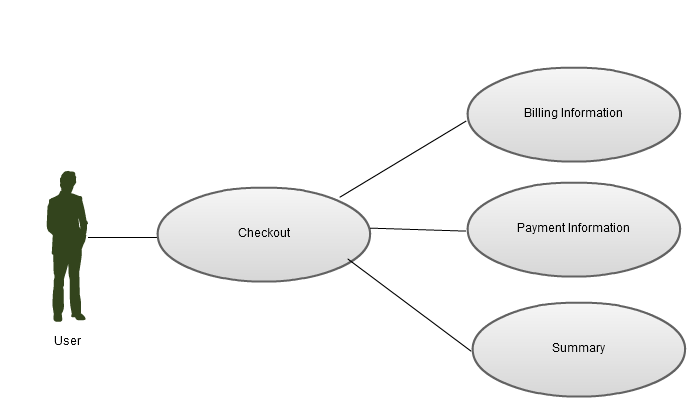


Use Case: Checkout

|  |  |
| --- | --- |
| Use Case: | Checkout |
| Actors: | User |
| Type: | Primary |
| Purpose: | Collect the payment and shipping information for the desired items. |
| Description: | The user will see a summary of the transaction. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) The user clicks on the checkout button. | The checkout page is shown. |
| 2) The user verifies the shopping cart and enters their payment/shipping information. | The transaction is processed and a confirmation page is shown. |

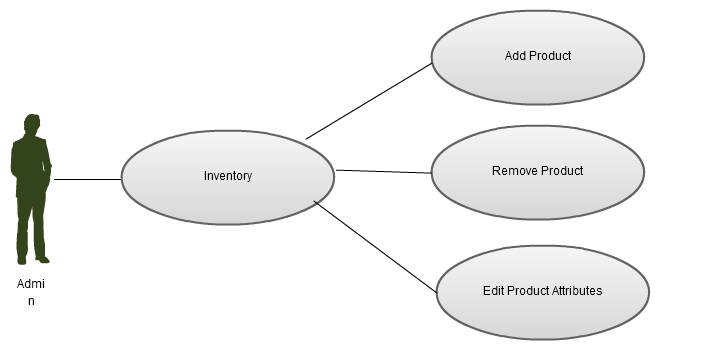


Use Case: Edit Inventory Item

|  |  |
| --- | --- |
| Use Case: | Edit Inventory Item |
| Actors: | Admin |
| Type: | Primary |
| Purpose: | A means by which the manager or administrator can edit an item in the sites inventory. |
| Description: | A populated inventory items page, where an administrator can manipulate item properties or attributes. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) The admin clicks on the edit button for a particular inventory item. | The items inventory page is shown with editable fields. |
| 2) After the admin modifies the fields he/she presses the save button. | The updated information is saved and the un-editable inventory page is shown. |

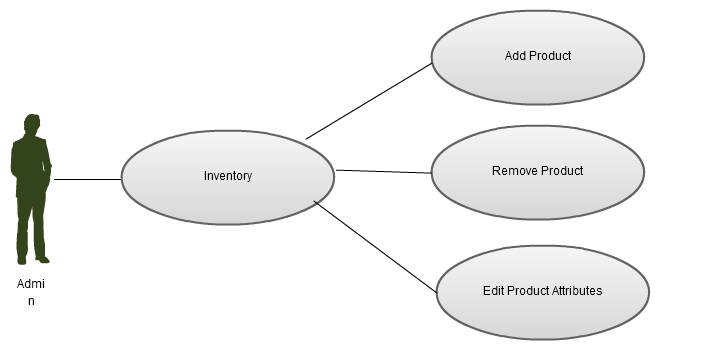


Use Case: Add Inventory Item

|  |  |
| --- | --- |
| Use Case: | Add Inventory Item |
| Actors: | Admin |
| Type: | Primary |
| Purpose: | Allows new items to be added to the site. |
| Description: | The administrator will be shown blank inventory fields and will be able to populate and ultimately add new items to the site. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) The admin clicks on the add item button. | A blank inventory page is shown. |
| 2) After the admin adds details about the inventory item they press the save button. | The updated information is saved and the un-editable inventory page is shown. |

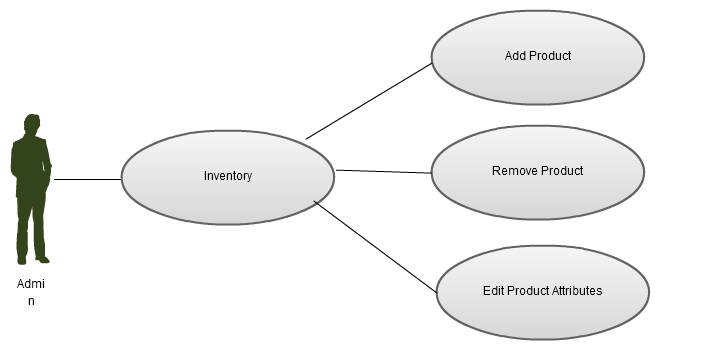


Use Case: Delete Inventory Item

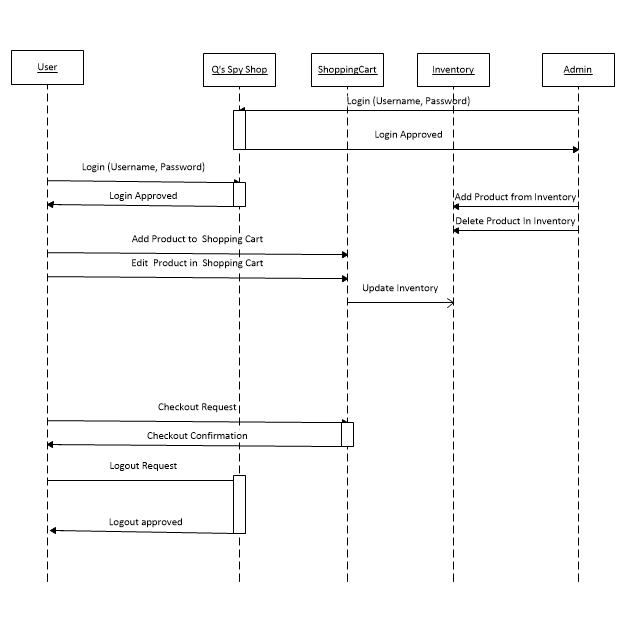
|  |  |
| --- | --- |
| Use Case: | Delete Inventory Item |
| Actors: | Admin |
| Type: | Primary |
| Purpose: | The ability for the manager or administrator to remove an item from the site. |
| Description: | This page will show the items details and allow the admin to justify then delete the item from the site. |

Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1) The admin presses the delete button. | A confirmation page is shown. |

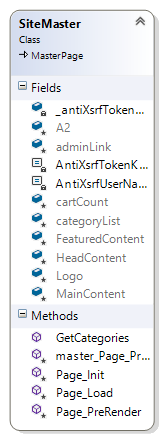
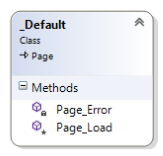
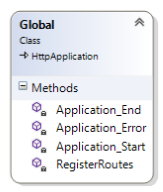
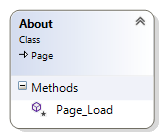


**Sequence Diagram:** Once the users and external interactions are listed in the use cases, a sequence diagram can easily document the interaction between the objects, as seen in our Sequence Diagram for Q’s Spy Shop:

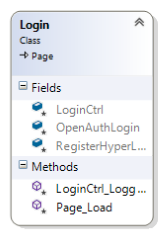
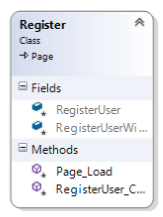
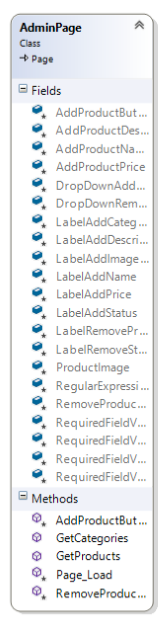
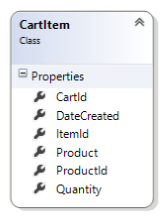
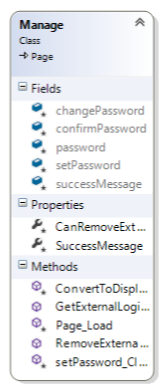


Sequence Diagram

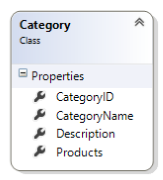
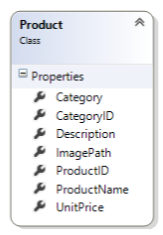
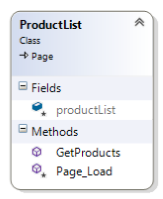
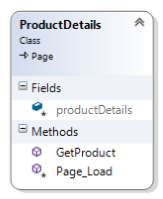
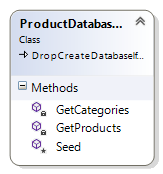
With the Use Cases and Sequence Diagrams completed, we now elaborate on the design and details for each of the classes. The following Class Diagrams, arranged by functionality, are a subset of the classes that will ultimately be used.

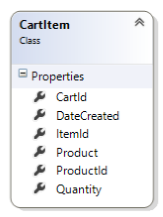
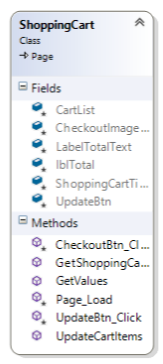
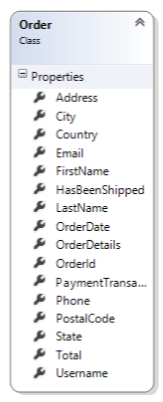
Foundational Classes

Actor (User & Admin) Classes

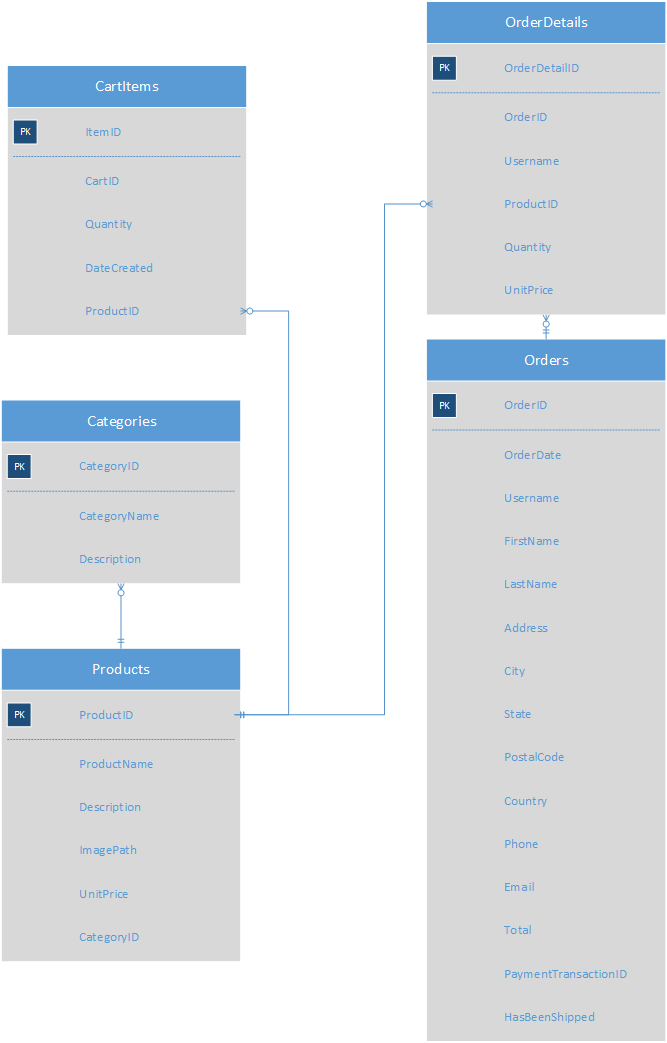
    

Inventory (Related) Classes

Transaction Classes

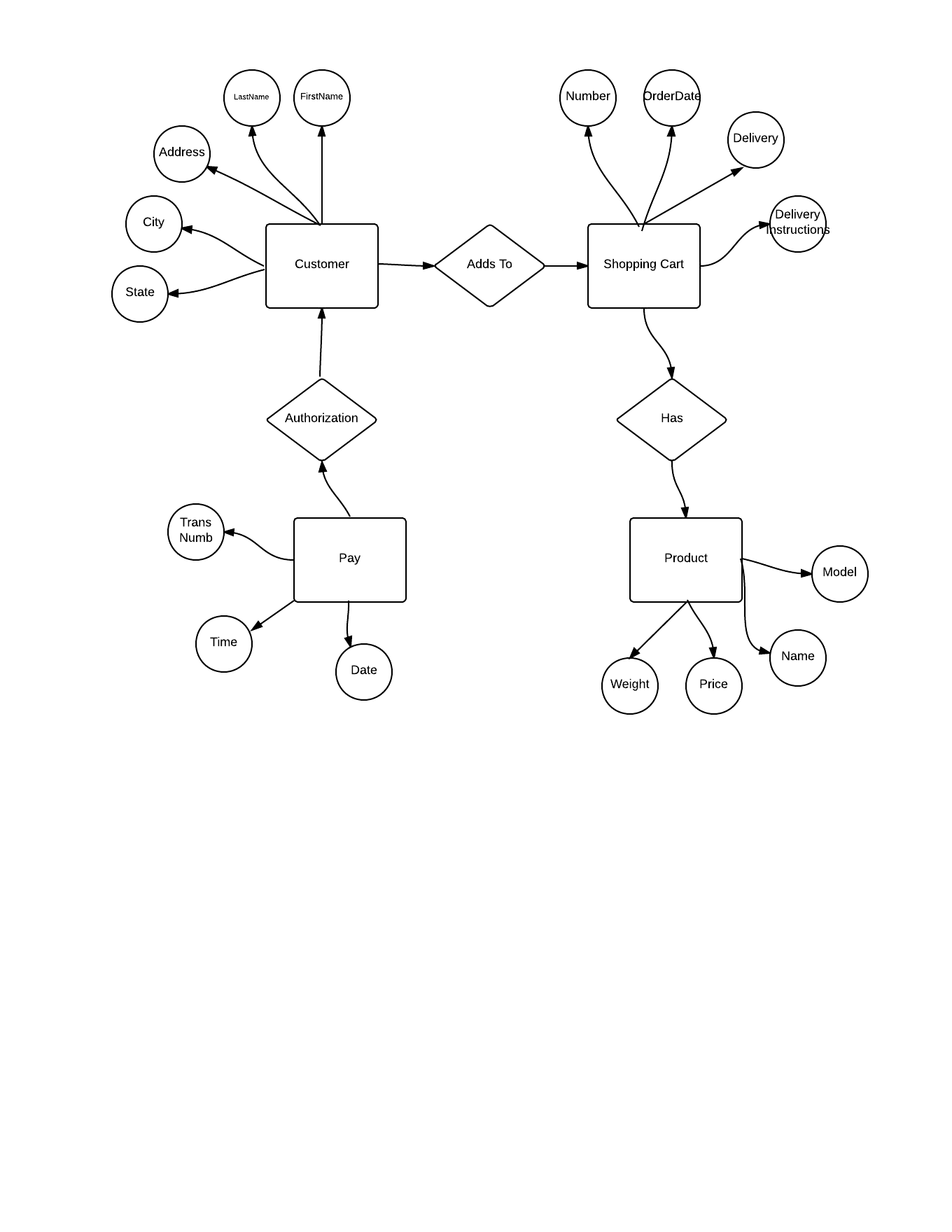
**Database Diagram:** Upon completing the detailed conceptualized of high-level objects, actors, and classes, we focus on the structure and design of the database. Given that Q’s Spy Shop is an e-commerce site with a modest inventory, we feel that Microsoft’s database tools in Visual Studios 2012 will be sufficient to build and maintain the site. Below is the database design diagram for Q’s Spy Shop.



Basic Database Diagram

With the majority of Q’s Spy Shop designed there are additional, and often minute, implementation details we wish to address. These details, or approaches, are helpful with the design throughout the implementation of an e-commerce site. For example, versioning is a key concept in many coding project. We have decided to use Git in concert with bitbucket. Coding standards are often overlooked in project, yet, an early adoption of defined standards prevent issues. During the development of Q’s Spy Shop, we will comply with that of Microsoft Visual Studio 2012 standards. These areas are ideal for given items such as company logos, and other information, that is critical and should be displayed on every page.  Sites must maintain visibility of potential options and make suggestions when tastefully possible. Finally, sites should always communicate the state to the customer and provide reachability of critical items such as the shopping cart and user settings.

**ER Diagram:**



**Acknowledgement:**

We would like to thank Prof Yuke Wang, for his continued efforts and excellent guidance in our learning and practical projects and we are motivated by his specialized method of teaching.