**Project Description**

Our system is designed for a fictitious online vendor known as “Small Shops ‘R Us.” Their goal is to sell the products of smaller businesses. The purpose of our system is to automate taking inventory in response to customer orders. Inventory can be manually altered to account for shipments to inventory. New inventory items can be added, the names of inventory items can be altered, the price per unit of inventory items can be altered, and the quantity of inventory items can be altered. The value of the current inventory quantity is calculated by multiplying the inventory price per quantity by the units of quantity. The main limitation is that the ID values cannot be manually entered or altered. Inventory data is stored in an Inventory table within a database file. Primary key values are automatically assigned to new entries, and the primary key values in the database act as the ID values for the inventory items.

Customers of the system will be able to create user profiles, and it is through these profiles that customers can add items to digital carts and place orders. The user profiles will contain all basic relevant information that is needed for ordering items: a user ID, a user name, an email address, a login password, a credit card, a city, a state, a country, an address, and a phone number. Users can update all of their profile’s information, except for changing their user ID or email. As with inventory item ID, the user ID is automatically generated when a new entry is entered into the database file. The database file will have a table for Users. The email is used for logging in, and it must remain unique for each user. It is easier to accomplish this by keeping user email static. As with most websites, a single email address can only create a single account.

Orders will involve all of the basic information that is required for ordering items on the web: order ID, user ID of the person that placed the order, the address of the user, the credit card of the user, the timestamp for the order, the list of ordered items, the dictionary of prices per item, the dictionary of quantities per item, the dictionary of cost per item, the dictionary of original quantity in inventory per item, the dictionary of original value in inventory per item, the dictionary of altered quantity in inventory per item, the dictional of altered value in inventory per item, and the total cost of the order. As with inventory item ID and user ID, order ID needs to be unique. Once an order is finalized (i.e., entered into the database file), it will be given a primary key value that acts as its ID value. It will also be given a date that matches the current date. The quantity of each ordered item will automatically be deducted from the quantity of each item in inventory.

There will be object classes for Inventory, User, and Order. That will be enough to cover the basic functionality of the system. The Inventory class covers the inventory items of the digital store, the User class covers the customers that can place orders, and the Order class covers the orders that customers place while also automatically updating the inventory that is defined in the Inventory class. The attached UML Class Diagram goes in depth for how the desired functions of the system will map into the classes.

The basic assumption for this project is that it is an object-oriented assignment. Its purpose is to create an application that serves a practical function through object-oriented programming. The current necessity to make the final project a web application is more of a recent addition to the course, and the material that introduces students to web frameworks does not provide a thorough enough introduction to allow the interface to be completely user-friendly or attractive. The main point is to create an object-oriented application that serves a practical purpose, not to create a fully fleshed out website or backend to a website. For this project, the web interface is primarily used for input and output. Basic HTML templates are used, as are views in Django with basic JSON dictionaries.

The Django framework is used for this project, but it is not used thoroughly. All major processing of data is in the views.py folder, an independently created database file is used, the database file is directly accessed through SQL using sqlite3, and the object classes are directly defined in views.py. Beyond altering settings.py to include the name of the application, creating an application level urls.py file, and altering the project level urls.py file to include the application level urls.py file, nothing else in the framework is used. There are no administrative functions and there are no models in models.py. The project is only being used to process and store data for inventory, users, and orders in the correct way.