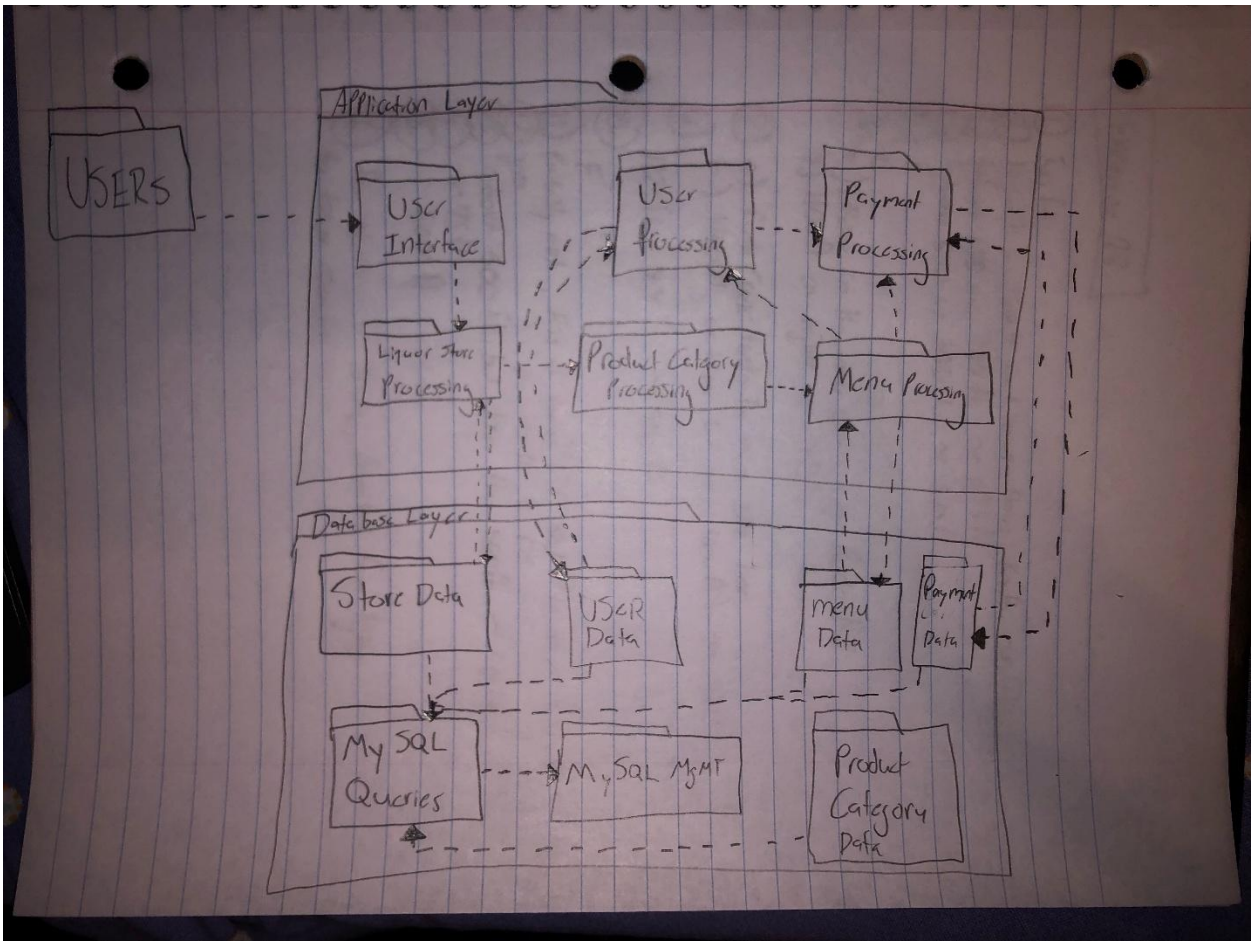
Alcohol Shop Website Information System

System Architecture and System Design:

- **Architectural Styles**
    - The Liquor Store system will be a traditional client server architectural style. Client is the user's computer that sends out requests to a server and gets back information. Client will not know how the server processes information or how the database is set up. The server will be a computer set up to deal with the client's requests. My system will be cross-platform and I will be using applications like XAMPP with APACHE, MYSQL, and PHP to allow me to set up the site on my local webserver. Apache will display the web pages on my localhost and will sort through the requests from clients. The PHP will take requests from Apache for further processing. After those requests are filled, the system will use MYSQL as the database that stores all the important data from the user and the system (such as the inventory, user accounts, cookies, etc) in tables accessible for certain requests. The user interface will run on the client side and the data will be stored on the server to protect it. I will have to run this system on only one machine, since I only have access to one. The Liquor Store system is a local server on my computer and will only have one client computer available.
- **Identifying Subsystems**
    - I'll provide the setup for my UML in the form of a package diagram that will show the layout of the system and the subsystems of my Liquor Store system. This system will have a typical client server architecture style discussed above, so there will be need for both an application layer and a server layer containing the databases for storage of data. The application layer will need a "Users" package that connects with the "User" interface package and decides what information the is display on screen. I will use HTML, CSS, and some JavaScript to display information using data taken from certain buttons, forms, and other even handlers based on clicking of certain items and increasing values that will change depending on information being input from the user. This information will be used to display certain products in categories, user login, inventory processing and editing, menu processing, checkout processing and editing of shopping cart. Based on user input, these packages will either get or post store info, product info, menu items, credit card information, from our database storage layer or to it.

The database using MYSQL will filter information from Queries sent to our SQL database. Our database is then updated or outputs information based on the query to the user.

UML:



- **Persistent Data Storage**
    - We will need to be able to store and save data that we will endure after the execution of our system. The objects will need to persist in the form of Usernames and passwords, inventory and product information, pertinent credit card information, and past orders. The plan for the storage of this data and persistent objects is to save them in a relational database like MySQL. Also need product table for product id, name, description, price, and quality. Orders table with orderDate, and orderID.

Global Control Flow:

- **Execution Orders**
    - The Liquor Store systems will be an event driven system that waits for user actions in loops. The system will have a series of web pages with buttons that a user can click and perform an action. The events will prompt certain actions like going to a different category page, user login page, editing product amount, and adding or going to the checkout cart. Some of this information will need to be associated with SQL queries and information charts like the user login and credit card information section. Most of the buttons will be displayed on all the different webpages and allow users to do certain tasks across the webpage, like searching, going through certain products, editing the products, and login and cart options and allows for some user flexibility and more control.
- **Time Dependency**
    - No timing is needed for the site since it is mainly an event driven system. There will be periodic updates to things like inventory, but time is not exactly a factor for the system and there are no real-time constraints for each period. The homepage and other pages will be based on the data received from queries sent from our database that will display the relevant information on a user's webpage. It will change based on user clicks and information input and relay information after the queries based on the data received back from our database.
- **Concurrency**
    - In the real world, the system would need to be able to handle several users browsing at once and the system should be able to handle these concurrent tasks. The backend of the system should be capable of handling multiple actions.
- **Hardware Requirements**
    - Basic technology requirements. I'm using my laptop to build the system, so this should be able to work on any computer or mobile device. It does need a color screen to display information and will need a hard disk drive to store database data and the XAMPP application software.
        - Needs Color Monitor or display screen: Minimum resolution :64-x 480PX
        - Computer: Desktop, laptop, or mobile device
        - Reliable Internet Connection
        - Memory: no more than 2 Gb RAM
        - Hard Drive: Minimum 1 Gb hard disk drive space