ShopNet

The ***ProductServer*** application acts as a server to handle product subscriptions and purchases. It utilizes both UDP and TCP protocols to manage communication with clients. The server maintains a set of available products and a set of subscribers to notify about product purchases.

The ***ProductClient*** serves as a client interface for interacting with the ***ProductServer*** to subscribe to product notifications and make purchases. It utilizes both UDP and TCP protocols to communicate with the server.

Server usage:

1. **Running the Server:**
    - Execute the ***ProductServer*** Java class.
    - The server will start running and listening for incoming UDP and TCP connections.
2. **Subscription and Purchase:**
    - Clients can subscribe to product notifications using UDP and purchase products using TCP.
    - UDP is used for subscription notifications, while TCP is used for product browsing and purchasing.
3. **Exiting the Server:**
    - To stop the server, terminate the application process.

Note:

Ensure that both UDP and TCP ports (9000 and 9001) are available and not blocked by firewalls.

Client usage:

1. **Running the Client:**
    - Execute the ***ProductClient*** Java class.
    - Upon execution, the client will initiate connections to the server using both UDP and TCP protocols.
2. **Subscription for Notifications:**

- The client automatically subscribes to product notifications upon initialization using UDP.
- It listens for notifications from the server regarding product purchases.

3. **Product Browsing and Purchasing:**
   - Clients can browse available products and make purchases using TCP.
   - Enter commands as follows:
     - `'all'`: View all available products.
     - `'buy' + product name`: Purchase the specified product.
     - `'exit'`: Terminate the client application.
4. **Interaction with the Server:**
   - UDP: Sends subscription requests to the server and receives product notifications.
   - TCP: Establishes a connection with the server to browse products, make purchases, and receive responses.
5. **Exiting the client:**
   - To exit the client application, input `'exit'` while using TCP communication.
   - Upon termination, the client closes its connections with the server.

Note:

Ensure that the server is running and accessible before executing the client. Proper network connectivity is required for seamless communication between the client and server.