AUBoutique - Phase I

EECE 351 - Computing Networks and Services

Team Members:

Joey Saade (202305993, jhs15@mail.aub.edu, 33.33%)

Tatiana Nohra (202400396, tsn08@mail.aub.edu, 33.33%)

Christophe El Chabab (202401778, cge24@mail.aub.edu, 33.34%)

November 8, 2024

1. System Architecture and Protocol Design

The AUBoutique platform is built as a client-server application designed to facilitate the buying and selling of products within the AUB community. The system follows a client-server architecture where clients (users) interact with a central server to manage user accounts, list products, and handle transactions.

Client-Server Communication Protocol

The communication protocol between the client and server is structured using JSON messages exchanged over a TCP connection. Each message from the client includes a `command` field specifying the action (e.g., login, register, add product). The server interprets the command and performs the corresponding action, then sends a response back to the client.

2. Project Features

The following table summarizes the features implemented in AUBoutique for Phase I.

Feature	Implemented	Remarks
Account Registration	Yes	Allows new users to create accounts
User Login	Yes	Authenticates users with username and password
View All Products	Yes	Displays available unbought products
Add Product	Yes	Allows users to list new products
Buy Product	Yes	Confirms purchase and updates product status
Message Owner	Yes (Additional)	Sends messages (even if recipient is offline)
View Buyers	Yes	Shows who bought the user's products
Check Owner Status	Yes	Checks if a product owner is currently online

Demo YouTube link: https://youtu.be/qGZ41bwaVeg
GitHub repository link: https://github.com/Nohra2005/AUBoutique

3. Implementation of Functionalities

Client Implementation

The client application connects to the server using a socket on the specified port and provides the following functionalities:

- **Registration and Login:** The client allows users to register by providing their name, email, username, and password. Registered users can log in with their credentials, which the server verifies.
- **Product Listing and Purchase:** Users can view all products, view specific products by owner, and buy a product, receiving a confirmation message upon purchase.
- **Messaging System:** The client enables users to check if a product owner is online and send messages. If both parties are online, messages are sent immediately; otherwise, they are queued.
- **Product Management:** Users can add products with details like name, description, and price. Each product can only be bought once.

Server Implementation

The server application listens for client connections and handles multiple clients simultaneously. Key functionalities include:

- **User Management:** The server stores user information and manages login and registration requests.
- **Product Management:** The server keeps track of available products and updates product status after purchase.

- **Messaging System:** The server handles messaging between users, delivering messages directly if the recipient is online or saving them for later retrieval, sending them at login.

Protocol Implementation

The protocol implementation is the file that stores all the function definitions that are used in the client and server implementations. The function of this file is to keep the client and server files clean and readable.

- **Client Handler:** It is used by the server to handle the client based on the command.
- **Miscellaneous:** Other functions used by client and server (clearly stated in protocol file).

4. Code Structure and Database Schema

Client Code (AUBoutique_client.py)

The client code includes functions for sending commands, handling responses, and maintaining a listener thread for incoming messages. Each command, such as `register` or `login`, is sent as a JSON message to the server, and responses are processed accordingly. The following are functions included in the client code:

- **Client Setup:** Connect client and server through TCP socket.
- **send_command() Function:** This function takes a command and the data associated with it and sends a message containing this command to the server.
- **listen_for_responses():** New thread that is constantly listening for incoming responses from the server and differentiates between arriving messages and responses.
- **Registration/Login Interface**
- **List of Actions:**
  1. Add a Product
  2. View my Product's Buyers
  3. View All Products
  4. View Products by Owner
  5. Buy Product
  6. Check if Owner is Online
  7. Message Owner
  8. Quit

Server Code (AUBoutique_server.py)

The server code consists of a main server file that accepts client connections and spawns a new thread for each client. The server stores user and product information in an SQLite database. The following are functions included in the server code:

- **Server Setup:** Bind server socket to an IP address and a port number, accept connection with client
- **Multi-threading:** Open a new thread for each client, using client handler function.

Protocol Code (protocol.py)

The `protocol.py` file defines the `client_handler` function, which processes commands and manages responses. The following are functions included in the protocol code:

- **client_handler():** This function is used by the server, it assigns a username to the client, receives a command from client and calls the corresponding function accordingly:

  1- If the command is to register, call the registration_handling function also defined in the protocol, then add the username in the database.

  2- If the command is to login, call the login_handling function and send the appropriate response based off the success/failure of login. If the login is successful, append to the response any pending messages for the user name by calling get_pending_messages.

  3- If the command is to add products, call add_product also defined in protocol: enter product information and store in product database.

  4- If the command is to view buyers, call view_buyers also defined in protocol: enter in the product database and extract all the buyers of a specific owner (client caller of function).

  5- If the command is to view products, call view_products also defined in protocol, there is also a filter to view products by owner, both require accessing the product database.

  6- If the command is to buy a product, call buy_product also defined in protocol: find the product in the product database using the product name, if it is available, update status to bought in database, otherwise, inform the client that is product is unavailable (does not exist or already bought). Then calculate the pick-up day (2 day after order) and add it to the response.

  7- If the command is to check online status of another owner, call check_online_status defined in protocol: if the username of the owner is in the online_users dictionary: the user is online.

  8- If the command is to send a message, call send_message defined in protocol: if the receiver user is online, send the message via the server marking its status as sent; if the user is offline, mark the message as pending which will be later checked.

  9- If the command is to get pending messages, call get_pending_messages defined in protocol: retrieve messages from the receiver's inbox from the message database and return them, then mark them as sent.

Database Schema

The database, `auboutique.db`, contains the following tables:

- `users` - Stores user information such as name, email, username, and password.
- `products` - Stores product details like name, description, price, owner, and buyer.
- `messages` - Stores sent and pending messages between users.

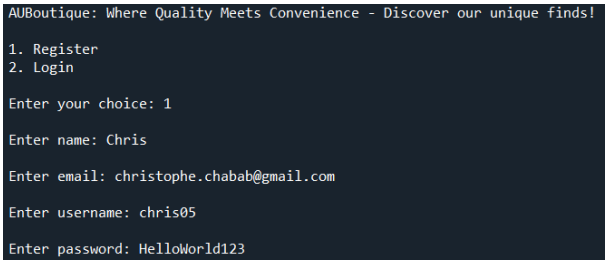

Figure 1: User Registration Interface

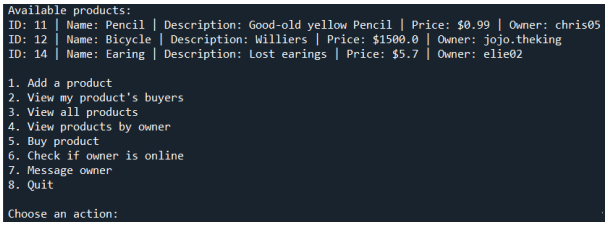

Figure 2: Product Listing Interface

5. Appendices

Appendix A: Application Snapshots

Appendix B: Project Task Breakdown

Responsible Team Member	**Tasks**
Tatiana (33.3%)	Client and Server Setup - Registration and Login Handling - Client Handler in Protocol - User Table (Database) - Documentation
Joey (33.3%)	Add Product - View Buyers - View Products (with filter for owner) - Buy Products - Product Table (Database) - Documentation
Christophe (33.4%)	Send Message - Check Online Status - Multi-threading - Storing Offline Messages - Buy Products - Messages Table (Database) - Documentation

Table 1: Breakdown of Project Tasks and Team Member Responsibilities