EECE 351:

AUBoutique Project Report

Muhammad Natour – ID: 202400340 – mia67@mail.aub.edu

Maria El Farran – ID: 202401368 – mse60@mail.aub.edu

Sandra Fawaz – ID: 202401374 – shf15@mail.aub.edu

Dr. Ayman Kayssi

December, 2024

Workload Distribution:

- Muhammad: 33%
- Maria: 33%
- Sandra: 33%

1. Introduction

AUBoutique is an online marketplace designed specifically for the AUB community to facilitate the buying and selling of various products. The project was developed in two phases:

1. **Phase I:** Focused on building a basic online marketplace whose main functionalities include account management, product listing, product selling, and communication.
2. **Phase II:** Enhanced the system with a user-friendly Graphical User Interface (GUI) using PyQT5, peer-to-peer communication, product ratings, multi-currency support, and additional features.

---

2. System Architecture

2.1 Client-Server Architecture (Phase I)

The initial implementation was designed to employ a client-server architecture. The server acts as a central repository for product listings and user data. Clients (buyers or sellers) interact with the server through commands that manage product listings, view items, and handle transactions. All interactions between clients were routed through the server.

2.2 Hybrid Architecture (Phase II)

In Phase II, the system was evolved into a hybrid model:

- The server manages user accounts, product database, and initial connection setup still making use of the TCP connection.
- Peer-to-peer communication allows direct interaction between users, reducing server load and enhancing real-time capabilities.

---

3. Protocol and Communication

3.1 Communication Protocols

- **TCP (Transmission Control Protocol):** Used for communication between the client and the server to ensure reliable data transfer for critical actions like login, product management, and notifications.
- **UDP (User Datagram Protocol):** Used for peer-to-peer communication between users in the chatting system to enable low-latency and efficient message transfer.

3.2 Client Request Handling

The client sends JSON-encoded requests to the server, specifying the desired action (e.g., `login`, `add`, `buy`, `rate`, `convert currency`, `follow user`). Each request includes relevant data, such as user credentials, product details, currency requested, or message content.

3.3 Server Response Handling

The server processes the request, interacts with the databases as needed, and sends a JSON-encoded response back to the client. The response indicates the status of the action (e.g., success, failure) and may include additional data (e.g., product lists, messages).

---

4. Detailed Features and Implementation

4.1 Account Management

Registration (Client-Side)

- User inputs name, AUB email, username, and password.
- Passwords are hashed using SHA-256 before being sent to the server.
- The client sends a `reg` action to the server.

Registration (Server-Side)

- The server checks if the username already exists in the database.
- If valid, user details are added to the `userInfo` table.

Login

- The client sends a `login` request with hashed credentials.
- The server verifies the credentials and updates the user's status to online in the database.

4.2 Product Management

Adding Products

- The client sends an `add` request with product details (name, price, description, quantity, image).
- The server also keeps track of the quantity of each product.
- The server stores the product in the `objForSell` table and notifies followers of the seller of the newly available product.

Buying Products

- The client sends a `buy` request specifying the product ID and quantity.
- The server updates the product's quantity in the database, logs the transaction, and adds points to the buyer's account.
- A confirmation message is sent to the client, including a collection date.

4.3 Messaging and Chat

Text Messaging (Phase I)

- Messages between users are routed through the server.
- The server stores undelivered messages and sends them when the recipient logs in.

Peer-to-Peer Messaging (Phase II)

- The server provides the recipient's IP and port to the sender and vice versa.
- Direct communication is established using UDP sockets.

4.4 User Following and Notifications

Following Users

- The client sends a `follow` request specifying the username to follow.
- The server records the relationship in the `followers` table.

Unfollowing Users

- The client sends an `unfollow` request specifying the username.
- The server removes the relationship from the database.

Notifications

- The server notifies clients when a user they follow adds a new product.

4.5 Currency Conversion

- Clients can select their preferred currency from a dropdown menu.
- The default currency is set to USD.
- Exchange rates are fetched from an external API (e.g., CurrencyBeacon) and used to display prices in the table in the selected currency.

4.6 Product Ratings

- Buyers can rate purchased products (1-5 stars).
- The server updates the product's total rating and rating count.

4.7 Product Bumping (Creative feature)

- Buyers gather points the more they purchase products from AUBoutique.
- Sellers can “bump” their products to the top of the product display table in exchange for the points they have collected in AUBoutique.
- Sellers can promote their products for up to 24 hours.

---

5. Feature Summary

Feature	Phase	Status
Account Registration	Phase I	Implemented
Login/Logout	Phase I	Implemented
Product Listing (Add/View)	Phase I	Implemented
Product Purchase	Phase I	Implemented
Messaging (Text)	Phase I	Implemented
GUI Integration	Phase II	Implemented
Peer-to-Peer Messaging	Phase II	Implemented
Product Ratings	Phase II	Implemented
Multi-Currency Support	Phase II	Implemented
Quantity Tracking	Phase II	Implemented
Product Search	Phase II	Implemented
Follow/Unfollow Users	Phase II	Implemented
Point System	Phase II	Implemented
Product bumping	Phase II	Implemented

---

6. Implementation Details

6.1 Client-Side

The client is built using PyQt5 for the GUI. Key components include:

- **Login/Registration Pages**: Interfaces for account management.

- **Dashboard**: Tabs for browsing products, managing user inventory, chatting, and viewing followed users.
- **Search Functionality**: Allows users to search for products using a specific search term.

6.2 Server-Side

The server is built using Python and SQLite. Key responsibilities include:

- **Database Management**: Handles user data, products, transactions, and messages.
- **Request Handling**: Processes client requests and sends appropriate responses.
- **Notification System**: Alerts followers about new products.

6.3 Database

The database contains the following tables:

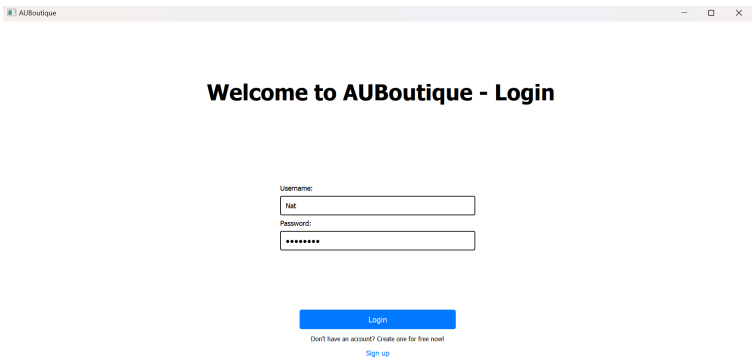
- `userInfo`: Stores user credentials and statuses.
- `objForSell`: Tracks product details.
- `log`: Records transactions.
- `messages`: Stores undelivered messages if user is offline.
- `followers`: Manages follow relationships.

---

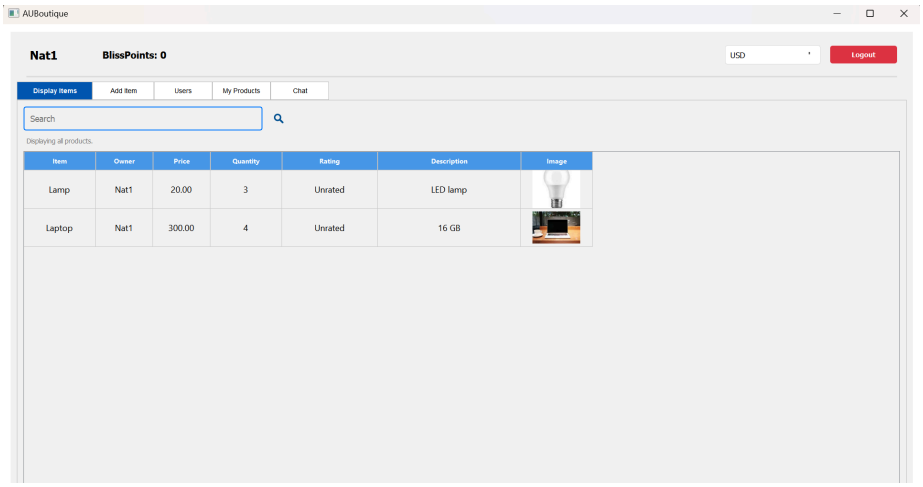
7. Appendix

7.1 Application Snapshots

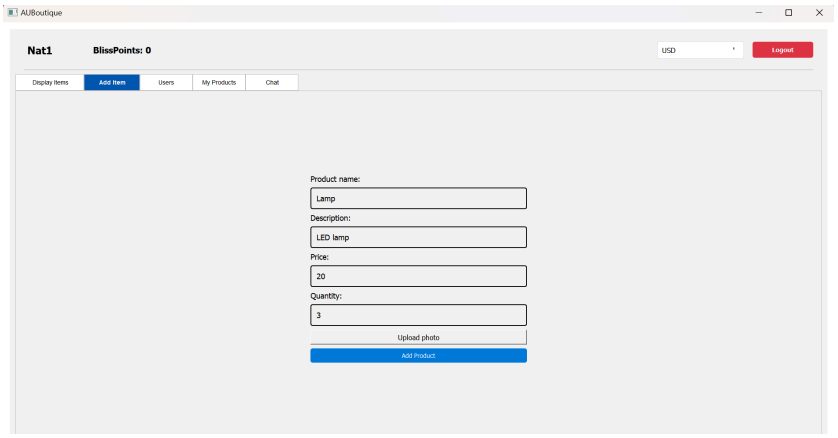
- **Login Page**: Allows users to log in.



- **Dashboard**: Displays products and user actions.

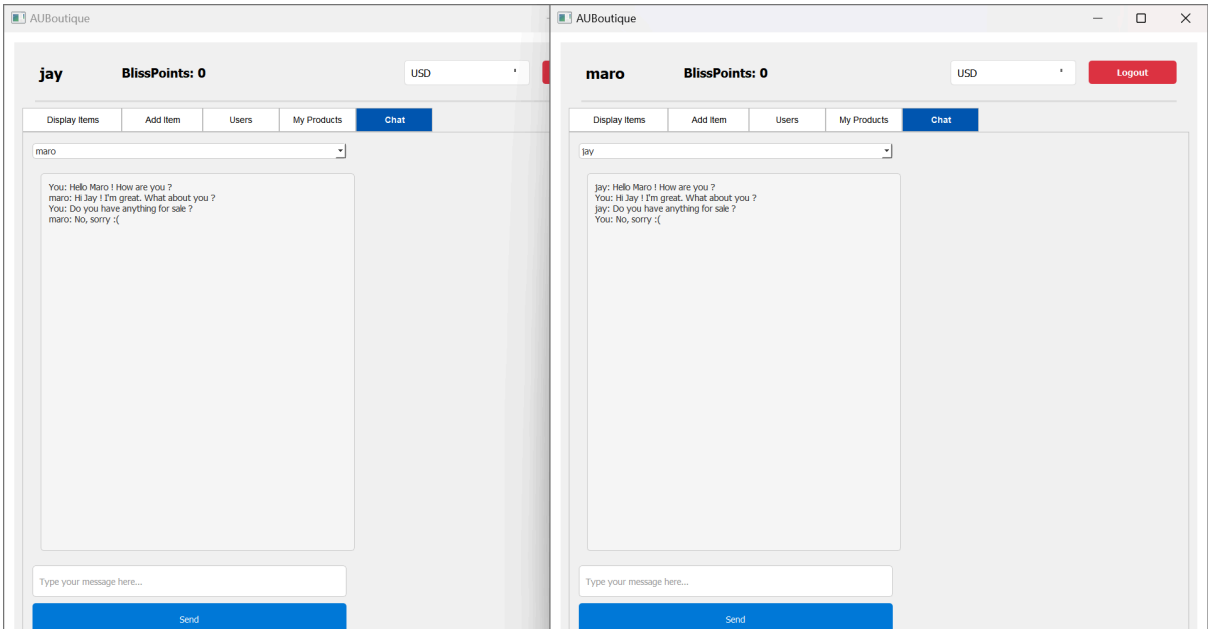


- **Add Product**: Add product to marketplace

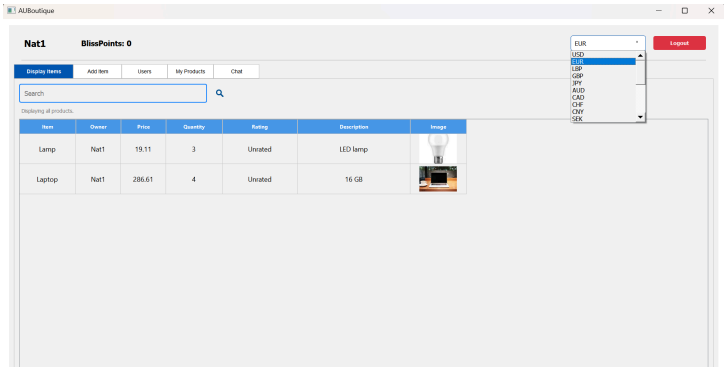


- **Chat Interface**: Facilitates peer-to-peer communication

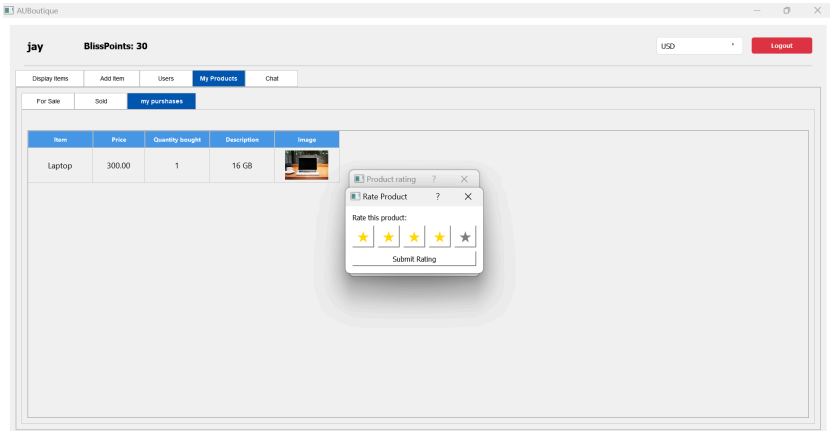
  Below is a chat between two users (maro and jay)

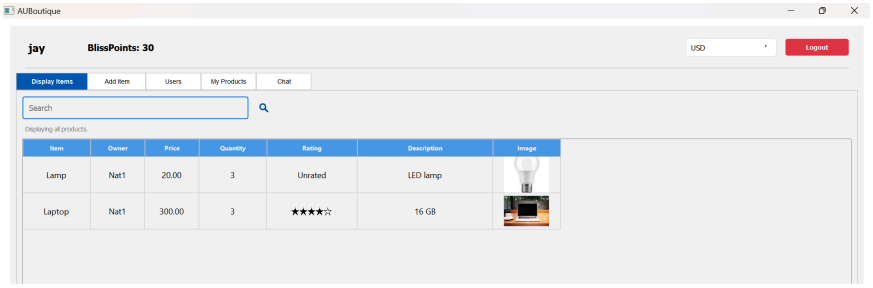


- **Currency Selector**: Enables multi-currency support.



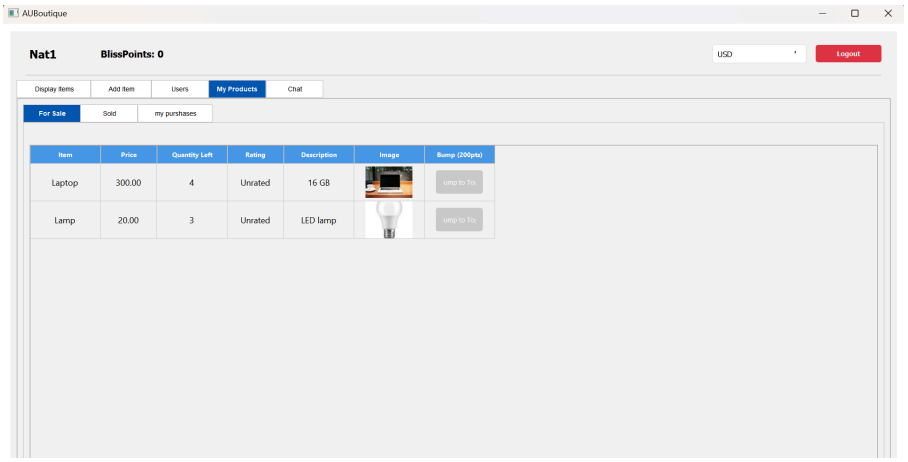
- **Rating products**: Buyers can rate products after buying them.



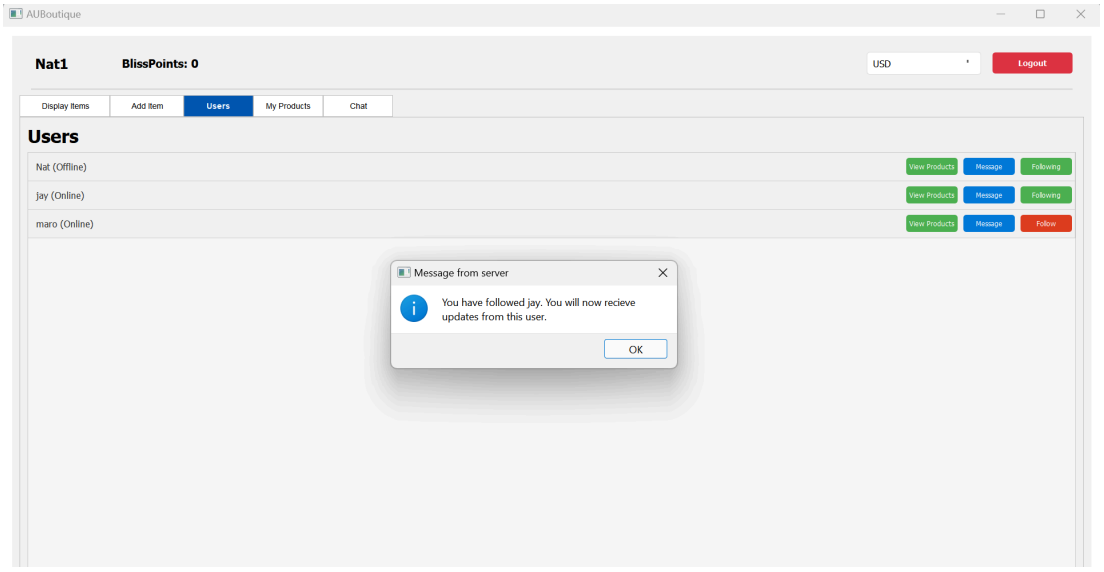


- **Bump to top**: Sellers can promote their products.

  (Button is in grey since seller does not have enough points to promote their products)



  - **Follow users**: Clients receive notifications when a user they follow adds a product to the marketplace.



7.2 Task Breakdown

Task	Responsible Member
Database Design	Maria
Client-Server Communication	All
GUI Design	Maria & Muhammad
Peer-to-Peer Integration	Muhammad
Product Ratings	Maria
Product Search	Muhammad
Purchases and Sales viewing	Maria & Muhammad
User Following	Muhammad & Sandra
Multi-Currency Feature	Sandra
Testing and Debugging	All

Link for GitHub repository :

https://github.com/SandraF13/AUBoutique---Final-Project

`