MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG445 Software Development with Scripting Languages

Assignment 2: Bookstore Application

This assignment aims to help you practice multithreading and concurrent programming, network programming, and graphical user interface components in Python. On the successful completion of this assignment, you will also practice Python basics. Your main task in this assignment is to develop a client-server application for a bookstore based on Transmission Control Protocol (TCP).

The client application will be used by both bookstore managers and cashiers, and multiple manager and cashier panels should be able to be opened at the same time, so the server should be able to communicate with multiple clients at the same time as illustrated in Figure 1.

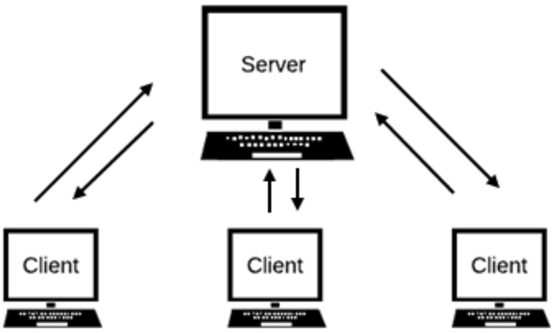


Figure 1: Client-Server Model

Overview

In this assignment, you will develop a client-server application for a bookstore. The application will allow the bookstore managers to manage the bookstore's inventory and allow the cashiers to track sales transactions. The client application will have different interfaces and functionalities based on the user's role.

Each book item in the inventory must include:

- Book ID (unique identifier)
- Title
- Author(s)
- Genre
- Price
- Quantity in Stock

Both bookstore managers and cashiers must login to the client application to perform their operations.

<u>Cashier Role:</u> Cashiers are responsible for processing sales transactions. A single transaction may involve multiple books.

- Processing a Sale: The cashier must enter the Book ID and quantity for each book being purchased.
- Discounts: A customer may have a discount code. If applied, the total price is reduced by 10%.
- Transaction Data: Each transaction must record the cashier, the date/time, a list of books/quantities sold, and whether a discount code was used.

<u>Manager Role:</u> Bookstore managers are responsible for managing the inventory and can request certain statistics.

- Add New Book: Enter details (Title, Author, Genre, Price, Initial Stock) for a newly acquired book.

- Update Stock: Increase the Quantity in Stock for an existing book (e.g., when a new batch of used copies is acquired).
- Generate Statistics: By using the client application, the managers can request the following analytical statistics related to the sales transactions, which the server must compute and return:
  - Top-Selling Author: Which author has the highest total number of books sold overall?
  - Most Profitable Genre: Which genre generated the highest total revenue?
  - Busiest Cashier: Which cashier processed the highest total number of transactions?

The server should manage four text files: **users.txt**, **inventory.txt**, **discountcodes.txt**, and **transactions.txt**.

1. The **users.txt** file will keep track of the details of the cashiers and managers as follows (username;password;role):
   ```
   john;csh123;Cashier
   sara;csh456;Cashier
   mike;csh789;Cashier
   managera;mgrpass1;Manager
   managerb;mgrpass2;Manager
   ```

2. The **inventory.txt** file will keep track the details of the books as follows (bookid;title;author(s);genre;price;quantity)
   ```
   1001;The Name of the Wind;Patrick Rothfuss;Fantasy;12.99;15
   1002;Sapiens: A Brief History of Humankind;Yuval Noah Harari;History;18.50;8
   1003;Gone Girl;Gillian Flynn;Mystery;9.75;25
   1004;The Hitchhiker's Guide to the Galaxy;Douglas Adams;Science Fiction;10.00;4
   1005;The Secret Garden;Frances Hodgson Burnett;Young Adult;7.25;11
   1006;Atomic Habits;James Clear;Self-Help;15.99;6
   1007;The Expanse: Leviathan Wakes;James S. A. Corey;Science Fiction;14.50;18
   1008;Educated: A Memoir;Tara Westover;Biography;17.99;3
   1009;A Storm of Swords;George R. R. Martin;Fantasy;11.25;22
   1010;The Whole-Brain Child;Daniel J. Siegel and Tina Payne Bryson;Self-
   Help;13.00;9
   1011;Where the Crawdads Sing;Delia Owens;Mystery;10.50;5
   1012;One Hundred Years of Solitude;Gabriel Garcia Marquez;Literary
   Fiction;16.75;1
   1013;The Hunger Games;Suzanne Collins;Young Adult;8.99;14
   1014;Thinking, Fast and Slow;Daniel Kahneman;Psychology;20.00;7
   1015;Good Omens;Terry Pratchett and Neil Gaiman;Fantasy;12.00;16
   1016;The Vanishing Half;Brit Bennett;Contemporary;11.50;0
   1017;My Lady Jane;Cynthia Hand and Brodi Ashton and Jodi Meadows;Young
   Adult;11.99;10
   1018;The Glass Ocean;Beatriz Williams and Lauren Willig and Karen
   White;Fiction;13.50;4
   1019;The Legacy of Heorot;Larry Niven and Jerry Pournelle and Steven
   Barnes;Science Fiction;15.25;7
   1020;Ask A Policeman;Dorothy L. Sayers and Milward Kennedy and Anthony
   Berkeley;Mystery;9.00;20
   ```

3. The **discountcodes.txt** file will keep track the discount codes as follows (discountcodes):
   ```
   12a345
   678b91
   23c456
   789d10
   ```

   These discount codes can be used only once and should be deleted from the file once they are used.

4. The **transactions.txt** file will keep track the details of the orders as follows (cashier; date/time;discountapplied;totalprice;bookid-quantity;bookid-quantity;…). Please ensure your code matches the formatting shown (timestamp format, price format, etc.).
   ```
   john;2025-11-15 10:30:00;N;26.75;1003-2;1005-1
   sara;2025-11-15 11:45:00;Y;20.93;1009-1;1015-1
   mike;2025-11-15 14:00:00;N;54.49;1006-1;1014-1;1002-1
   john;2025-11-16 09:15:00;N;53.50;1007-3;1004-1
   sara;2025-11-16 12:30:00;Y;36.67;1017-1;1018-1;1019-1
   ```

```
mike;2025-11-16 15:45:00;N;12.99;1001-1
sara;2025-11-16 17:00:00;Y;52.20;1020-5;1010-1
```

Requirements

Authentication: When the client application is started and the connection is established with the server, the server will send a confirmation message to the client (message: **`connectionsuccess`**). Once this message is received, the client will then show the following login screen to take the username and password from the user (See Figure 2).

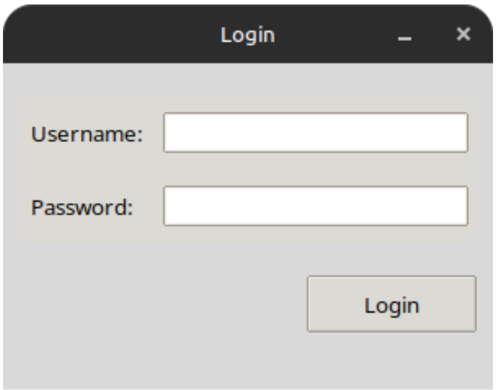
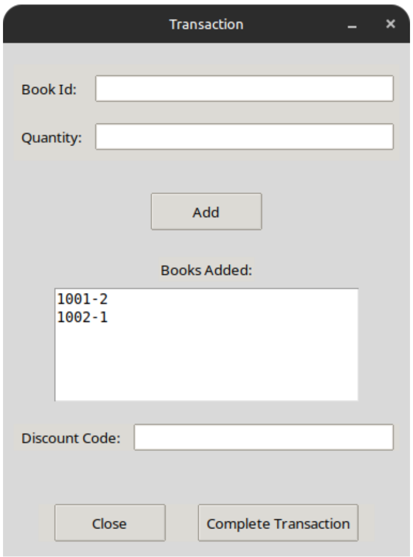


Figure 2: Login screen

Once the client sends the username and password to the server (message format: **`login`**`;`*`username`*`;`*`password`*), the server will check the **users.txt** file and send an approval message (message format: **`loginsuccess`**`;`*`username`*`;`*`role`*) or rejection message (message format: **`loginfailure`**) back to the client. If the login is not successful, then the error message box will be shown on the client side. However, if the login is successful, then the appropriate user panel will be shown based on the user role.

Cashier Panel: Figure 3 shows the user panel for a cashier. When a cashier wants to add a transaction, s/he needs to provide a book id and its quantity one by one for each book. If the customer has a discount code, then the cashier also needs to add the discount code. When the "Create" button is clicked, the details of the transactions will be sent to the server as follows: (message format: **`transaction`**`;datetime;`*`discountcode;cashier;bookid-quantity;bookid-quantity;…`* *or* **`transaction`**`;datetime;`*`;cashier;bookid-quantity;bookid-quantity;…`* if discount code is not provided)



```
1001-2
1002-1
```

Figure 3: Cashier Panel

The server then computes the total price by using the prices available in **inventory.txt**, and applies the 10% discount if the discount code is provided and available in **discountcodes.txt**. If a discount code is used, the code will be removed from **discountcodes.txt.** The server then adds the details of the transactions to **transactions.txt, updates inventory.txt**, and finally sends the confirmation message for the transaction with total price to the client (message format: **`transactionconfirmation`**`;`*`totalprice`*). The total price will be shown in a message box on the client side. In case of transaction failure for any reason (for example, discount code may not be available in the discountcodes.txt file or the inventory does not include the required number of books), the failure message will be sent to the client (message format: **`transactionfailure`**`;`*`failurereason`*)

When the "Close" button is clicked, the client will send a message to the server (message format: **`closeconnection`**) and the connection between the client and the server will be terminated.

Manager Panel: Figure 4 shows the user panel for a manager. The manager can add a new book (message format: **`addbook`**`;bookid;title;author(s);genre;price;quantity`). Please note that if the book has multiple authors are separated by "and", such as "Larry Niven and Jerry Pournelle and Steven Barnes". As such, you must ensure your code handles the white-space characters correctly when generating reports. The manager can also update the quantity of a particular book in the inventory (message format: **`updatequantity`**`;bookid;quantity`). After adding a new book or updating the quantity in the inventory for a particular book, **inventory.txt** is updated and a confirmation message is sent from the server to the client (message format: **`addbookconfirmation`** or **`updatequantityconfirmation`**). When the manager selects one of the available statistics reports, the report code will be sent to the server such as **`report2`**. The server will then generate the report and send the answer back (message format: **`report2`**`;`*`answer`* *or* **`report2`**`;`*`answer1;answer2;…`* if there are multiple answers). The answer should be shown in a message box appropriately on the manager's side.

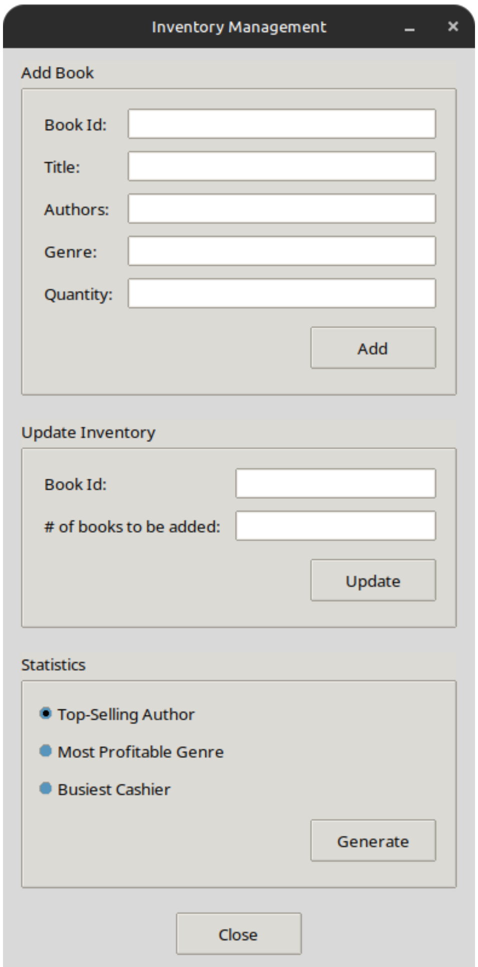


Figure 5: Manager Panel

When the "Close" button is clicked, the client will send a message to the server (message format: `closeconnection`) and the connection between the client and the server will be terminated.

Thread Synchronisation: You should consider the RLock thread synchronization technique to deal with any problems caused by attempting to access the shared data at the same file.

Rules

- You need to write your program by using **Python 3.x**.
- You can **only** use all built-in functions and modules.
- You also need to create a file called ReadMe.txt which contains the following items. Please note that **if you do not submit ReadMe.txt, your submission will not be evaluated.**
    - Team members
    - Which version of Python 3.x you have used
    - Which operating system you have used
    - How you have worked as a team, especially how you have divided the tasks among the team members (who was responsible for what?), how you have communicated, how you have tested the program, etc.
- You should name your server as **server.py** and name your client as **client.py**.
- You should not forget to submit your **users.txt**, **prices.txt**, **discountcodes.txt** and **orders.txt** files.
- You need to put all your files into a folder that is named with your student id(s) and submit the compressed version of the folder in the **.zip** format.
- **Only one team member** should submit the assignment.
- **Code quality, modularity, efficiency, maintainability, and appropriate comments** will be part of the grading.

Grading Policy

The assignment will be graded as follows:

Grading Item	**Mark (out of 100)**
Login	5
Transaction	15
Add Book	10
Update Inventory	10
Report 1	10
Report 2	15
Report 3	10
Thread Synchronisation	5
Cashier Screen	10
Manager Screen	10