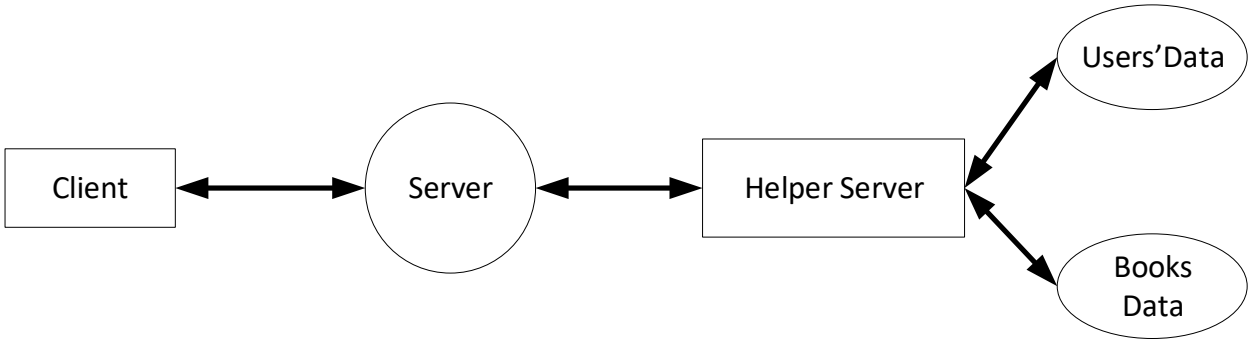
Networking Assignment

Assignment

The networking assignment requires developing a program to establish communication between programs over a network. This communication involves a scenario in which clients and servers exchange data.
The assignment simulates a library application. We assume the client sends a request to the server (librarian) asking about a specific book. The requested data may be on one of the data files maintained by the helper server as shown in the figure below.



The helper server uses two data files:

- A list of users and their contact information (email address, and phone number) stored in a json file. The format is Name:*user name*, Email: *user email*, Phone: *user phone*
- A list of books. This list includes [Book title, Author, Status (*Available*, *Borrowed*), Borrowed by, Return date] and is stored in a json file. If a book is available in the library, the status is *available* and the *Borrowed by*, and *Return date* fields are null.

The communication should be through TCP sockets between the client and the server, and a second TCP socket between the server and the helper server. The following messages will be communicated between the client and the servers:

1- The client sends a hello message identifying itself. The message should be a json object having Title:Hello, Sender:[*Client ID*]
2- The server replies with a json message having the format Title: Hello, Content: Welcome
3- The client sends the following message: Title:BookInquiry, BookName:[*Name of the book*]
4- The server forwards the message to the helper server
5- The helper server finds the corresponding information and returns it to the server
6- The server sends the information to the client

7- If the status of the book is *Borrowed* the client creates another message which includes Title:UserInquiry, UserName: [*user name*]. The user name is the name of the user who has borrowed the book and should be extracted from the first message
8- The server forwards the message to the helper server.
9- The helper server finds the information of the user and returns it to the server
10- The server forwards the message to the client.
11- If the requested book is not found in the list, the status field will be null and the second message (UserInquiry) will not be sent.
12- The client closes the socket and prints the information on the screen.

The list of books and users will be provided as json files named Books.json and Users.json. The client program should get the names of the books to be enquired as arguments.
Your program will be tested using other json files therefore, makes sure that the data files are read from the same location as the program, and their names are as specified here.
The application should not use any external libraries except for socket, json, and sys.
Sample sequence of messages:

1- Title:Hello, Sender:User1
2- Title: Hello, Content: Welcome
3- Title:BookInquiry, BookName:Programming (from client to server)
4- Title:BookInquiry, BookName:Programming (from server to helper server)
5- Title:Programming, Author:Knuth, Status:Borrowed, BorrowedBy:Hans, ReturnDate:10-06-2021 (from helper server to server)
6- Title:Programming, Author:Knuth, Status:Borrowed, BorrowedBy:Hans, ReturnDate:10-06-2021 (from server to client)
7- Title:UserInquiry, BookName:Programming (from client to server)
8- Title:UserInquiry, BookName:Programming (from server to helper server)
9- Name:Hans, Email:hans@gmail.com, Phone:0654321 (from helper server to server)
10- Name:Hans, Email:hans@gmail.com, Phone:0654321 (from server to client)