Overview of the System

- **Purpose**: The application is a multi-threaded server-client system designed for managing user accounts and facilitating financial transactions.
- **Architecture**: The system follows a client-server model, with a server handling multiple client connections concurrently.

Server-Side Components

Provider (Main Server Class)

- **Role**: It initializes a **ServerSocket** and continuously listens for incoming client connections.
- **Functionality**: On accepting a connection, it creates a **ServerThread** instance to handle client-server communication.
- **Design Choice**: Using a fixed port (2004) for simplicity and consistency in client-server communication.

ServerThread

- **Purpose**: Manages individual client connections, handling requests and responses.
- **Key Operations**:
  - **User Registration and Login**: Processes user registration and authentication requests.
  - **Transaction Handling**: Facilitates financial operations like money lodging and transfers.
  - **Communication Management**: Establishes and maintains input/output streams with the client.
- **Design Choice**: Implementing as a thread allows handling multiple clients simultaneously.

Library

- **Functionality**: Acts as a data manager for user accounts and transactions.
- **Key Features**:
  - **User Data Management**: Stores and retrieves user information.
  - **Transaction Processing**: Manages financial transactions, ensuring data integrity.
- **Design Decision**: Using a LinkedList for storing user data for efficient data manipulation.
- **File Operation:** Implement methods for reading from and writing to files.

Client-Side Component

Requester

- **Role**: Acts as the client, initiating requests to the server and handling responses.
- **Operations**:
  - **Connection Setup**: Establishes a socket connection with the server.
  - **User Interactions**: Handles user inputs for various operations like registration, login, and financial transactions.
  - **Server Communication**: Sends and receives data from the server.

Key Features

Registration and Login

- **Implementation**: Validates user details and credentials before adding or authenticating users.

Financial Transactions

- **Types**: Includes lodging money, transferring funds, and viewing transaction history.
- **Management**: Ensures accuracy and security in transaction processing.

Data Persistence

- **Method**: Uses file I/O for storing and retrieving user and transaction data.
- **File Management**: Separate methods for handling different data aspects like user info and transaction history.