Our application is going to be structured in a **client-server architectural style**.

**FLEXIBILITY** is achieved by allowing for easy addition or removal of servers and clients without disrupting the overall system functionality.

**RELIABILITY** ensures that the client-server application consistently performs its intended function without failures or errors. Implementing techniques such as redundancy, fault tolerance, and error handling is crucial to achieve a reliable system.

**MAINTAINABILITY** enables easy maintenance and updates by separating client and server components, allowing them to be modified independently.

**PERFORMANCE** optimize the system to provide efficient response times and minimize network latency for improved user experience.

**SECURITY** is essential to protect the client-server application from unauthorized access, data breaches, or malicious attacks. Implement appropriate security measures such as authentication, encryption, access controls, and secure communication protocols to safeguard the system and user data.

**COMPATIBILITY** ensures compatibility across different platforms, operating systems, and devices, allowing clients and servers to communicate effectively regardless of the underlying technology stack.

**LOW COST** design the system to maximize resource utilization, such as CPU, memory, and storage, to ensure cost-effective usage of server resources. This can involve optimizing algorithms, implementing caching mechanisms, or employing virtualization techniques.

**RAPID DEVELOPMENT** Adopting agile development methodologies, such as Scrum or Kanban, that promote iterative and incremental development. This allows for quick feedback, continuous improvement, and faster delivery of features and updates.