CONCEPTUAL ArchitecturE Document

**Project Title:** **CHAT APPLICATION**

**Team Members:** Christopher Cargile

Gayathri Parthasarathy

Nicklaus Rhodes

**Table of Contents Page No**

Architecturally Significant Requirements 3

Expected system qualities 3

Conceptual Architecture diagram 4

Client Component 4

Server component 5

CONCEPTUAL-ArchitecturE document

*Team Cupric seckel*

## ArchitecturalLY siGnificant requirements

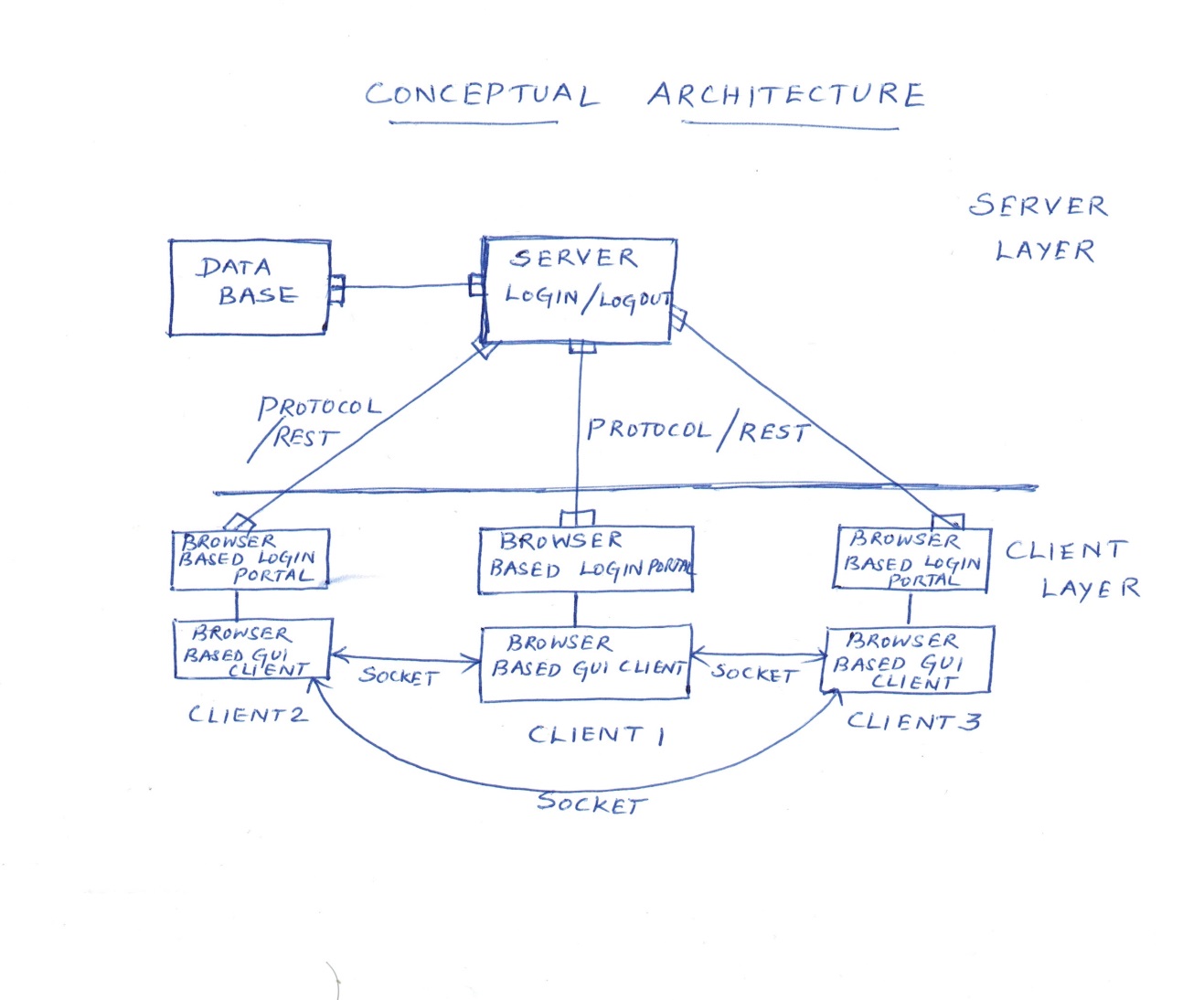
1. Chat application should be distributed and peer to peer application.
2. Server should provide the centralized control for the application.
3. Application should be browser based.
4. Clients should be able to send and receive texts.
5. Clients should be able to chat with all the online peers.
6. Server should notify all the online clients, whenever any client comes online and logs out.

**EXPECTED SYSTEM QUALITIES**

1. Chat Application should provide secure chatting environment.
2. Application should provide ease of use to the clients.
3. Application should provide privacy for the chat clients.

Conceptual Architecture of the chat application consists of two components – Client and Server.

The following picture shows the conceptual architecture of the chat application with the components and their interactions.

**CLIENT COMPONENT**

Client Component includes Browser based login portal and Browser based GUI chat application.

* Client types in the server URL in the browser and requests for the login page.
* Client types in the username and password into the required fields and clicks log in.
* If new client, they have to registers with the server to participate in the chat and logs in with the credentials.
* When the login credentials are verified, the client enters into the chat application.
* Application GUI shows chat history, List of users online, Text area to type messages and a submit button, Logout button.
* Once the client wants to communicate with a peer, it establishes a socket through which further communication takes place.
* When the client wants to go out of chat, clicks the logout button. The server disables the connection and the socket connection between the peers is also disabled.

**Responsibilities:**

* Allows client to initiate chat by logging in to the application through browser.
* Allows client to send and receive texts from online peers.
* Keeps track of the chat history.

**Collaborators:**

* Server verifies the client with the credentials stored in the database.
* Server notifies all online clients, whenever a client comes online and goes offline.

**Rationale:**

* Provides mutual independence between chat clients and facilitates distributed system.

**SERVER COMPONENT**

**Responsibilities:**

* Provides authentication of the clients.
* Allows centralized control of the chat application.
* Notifies clients about all online users and clients going offline.

**Collaborators:**

* Interacts with the database, to store the user information.
* Verifies the client login username and password with the information in the database.
* Client submits login and logout requests to the server.

**Rationale:**

* Provides centralized storage of information and control of the application.