Use Case Realizations Server Side

- User starts server. Server ensures it is capable of connecting to the proper port. If not it throws an error message; otherwise it moves to a resting state to wait for response from a client.
- Client sends request to log in. Server used internal computation to validate the login. If it works, it allows the client to connect and echos that it was successful; otherwise it denies the request and echos the problem.
- Client sends request to create an account. The server does the same thing as above and ensures that it meets the requirements. Stores account and sends an echo if it does work, otherwise it echos the failure.
- Client successfully logs in. Server checks to see if they have any friends and if those friends are online. If so, it sends a message to those friends letting them know that that person is online.
- Client requests to add a new contact. Server checks to see if contact is held on the server. If so, adds contact to client's list; otherwise it refutes the request.
    - *Elaboration:* The user A presumably has access to the name of user B. User A sends a request to the server to add B to their contact list. The server receives the request and checks its storage information (model) for user B. If user B is found, it adds B to A's contact info in storage (model) and allows A to see B on his contact list by sending A an updated contact list (sort of a view - communicating to the client), however, B must add A as a friend before they can chat.
- Client requests to delete contact. Server checks and confirms the contact's existence. Removes the contact if it exists, refutes the request otherwise. Server responds with success or failure status.
- Client A requests to chat with client B. Since A has to have B as a contact for them to chat, the server does need to check for that. Server opens a communication window between the two and sends responses to both clients. If A or B add C to the chat, C receives the entire chat log between A and B
- Client logs off. Server informs contact list of the client that the client is offline.
    - *Elaboration:* The user closes the client. This sends a message to the server. The server receives the message, marks them as offline in the storage (model). Then the server checks the user's contact list for anyone that's online. If so, the server sends that user a message that they're friend is offline.