Objects Involved for UC-01

User: The person on the computer using client application
LoginWindow: Login form GUI the user sees and interacts with, job is to collect what is typed and pass it along
WebSocketClient: Sends messages from client to server
WebSocketServer: Receives messages from the client on server side and passes them along
AuthenticationManager: Handed the credentials and handles validation logic, does not store account data
UserRespository: Read the .json file and only job is to store and retrieve data
MainWindow: The screen the user sees after a successful login

Step #	Who Initiates	Who Receives	Why (Intent)
1	User	LoginWindow	Start the application and show login form
2	User	LoginWindow	Provides credentials so they can be validated
3	LoginWindow	WebSocketClient	Formats credentials into a JSON login request and passes to WebSocketClient for sending
4	WebSocketClient	WebSocketServer	Sends the login request to the server
5	WebSocketServer	AuthenticationManager	Parses message and passes credentials for validation
6	AuthenticationManager	UserRepository	Retrieve the stored account data
7	UserRepository	AuthenticationManager	Returns stored data so credentials can be compared
8	AuthenticationManager	WebSocketServer	Reports that validation was successful
9	WebSocket Server	WebSocketClient	Sends the success response back to the client
10	WebSocketClient	LoginWindow	Passes response to the client UI
11	LoginWindow	MainWindow	Closes login form and shows main application

Step-by-Step Narrative

1. The User launches the client application, the LoginWindow is created and displayed with empty username and password fields, a disabled login button, and an option to create a new account.
2. The User enters a username and password into the LoginWindow. The LoginWindow then scans that both fields are non-empty and enables the login button.
3. The User clicks the login button, the LoginWindow formats the entered credentials into a JSON login request and hands it to the WebSocketClient to send to the server.
4. The WebSocketClient sends the JSON login request to the WebSocketServer for processing.
5. The WebSocketServer receives the message, parses the JSON to extract the username and password, and passes the credentials to the AuthenticationManager for validation.
6. The AuthenticationManager receives the credentials and requests the UserRepository to look up the provided username.
7. The UserRepository reads the JSON file, finds a matching account, and returns the stored account data including user's full name back to the AuthenticationManager.
8. The AuthenticationManager compares the provided password with the stored password. Validation succeeds, so it returns a success result along with the user's full name to the WebSocketServer.
9. The WebSocketServer builds a JSON success response with the user's full name and delivers it back to the WebSocketClient.
10. The WebSocketClient receives the response and passes it to the LoginWindow.
11. The LoginWindow closes, and the MainWindow is created and displayed.