**Reflection on Developing a Chat Application**

Working on the chat application project was both a rewarding and challenging experience. One of the major difficulties I encountered was implementing the multithreading and concurrency control mechanisms on the server side. Initially, I struggled with handling multiple client connections simultaneously without causing race conditions. However, after researching best practices in thread synchronization and implementing a locking mechanism, I was able to resolve these issues.

Throughout this project, I gained a deeper understanding of WebSockets and how they allow real-time communication between clients and servers. Prior to this, my experience with client-server communication was mostly limited to RESTful APIs, so transitioning to WebSockets opened a new perspective on real-time systems. I also learned about the importance of efficiently managing database connections, as multiple clients interacting with the same resources required proper handling to prevent deadlocks or data corruption.

A significant part of my development involved learning to work with Java Swing for the client-side interface. While I had previously worked with web front-end frameworks, Swing presented a different set of challenges. Designing an intuitive and responsive chat interface while ensuring proper integration with the server’s message handling was a key learning experience for me.

Although I primarily worked on the technical side of the project, I also honed my non-technical skills. Time management was crucial, as there were multiple moving parts—setting up the database, handling server logic, and designing the UI—that required careful planning and execution. Effective communication with peers also helped me troubleshoot issues and gain new insights.

One of the main takeaways from this project is how important it is to adopt a structured and modular approach. Breaking down the system into smaller, manageable components (like separating the User and Message classes into their own modules) made the development process smoother and easier to debug.

Looking back, I would have liked to implement additional features, such as rate limiting and message encryption, to enhance the functionality and security of the chat application. If given more time, I would also explore the use of message queues to further optimize concurrency handling in a real-world, high-traffic scenario.

Overall, this project improved both my technical skills and my ability to think critically when faced with complex problems. It has strengthened my confidence in building client-server applications, and I look forward to applying these skills to future projects.