Secure Chat Room Sever

Arush Sirotiya   
Department of Computer Science and Engineering with specialisation in Information and Technology   
S.R.M. Institute of Science and Technology  
Chennai, India  
as2733@srmist.edu.in

Sanyog Dani   
Department of Computer Science and Engineering with specialisation in Information and Technology   
S.R.M. Institute of Science and Technology  
Chennai, India  
sd6513@srmist.edu.in

*Abstract*: Our project is dedicated to breaking down geographical barriers and fostering communication through the use of technology. It comprises two vital components: a client application that operates within a user's web browser and a server application hosted on a network server. To initiate chat sessions, users

establish connections with the server, enabling both private and group conversations. The driving force behind our research is the recent surge in smartphone-based instant messaging applications, which, despite their popularity, often grapple with security vulnerabilities and privacy issues. Our primary research objective is to identify the requisite security features for an instant messaging application and devise a comprehensive system that

enhances data integrity, confidentiality, and privacy. At the heart of our approach lies the concept of End-to-End Encryption (E2EE), which fortifies the protection of user data. To achieve our goals, we scrutinize current security features in leading mobile messaging apps, distilling a set of security requirements, which, in turn, inform the architectural design of our secure messaging application. The ultimate validation comes through the implementation of a demo that is subjected to rigorous evaluation, ensuring it fulfills the defined security prerequisites while offering a robust level of privacy and data security.

Keywords: Secure Chat, Server, Client, Server Socket,

Java Swing.

# Introduction

Messaging apps serve as powerful tools to bridge geographical divides and facilitate instant communication, effectively shrinking the world. Whether it's staying connected with loved ones living far away or collaborating seamlessly with colleagues in different time zones, these applications play a vital role in fostering

connectivity. Furthermore, chat apps provide an empowering platform for individuals to express their ideas, share experiences, and engage in dynamic discussions. These apps have evolved to create vibrant communities where people come together to

discuss their interests and provide mutual support. In the ever-evolving landscape of communication technology, chat apps continually innovate, introducing new features and enhancing functionality. This constant evolution means there's always room for inventive and pioneering chat applications to enter the market, redefining how people interact. Beyond connecting people and enabling self-expression, chat apps also present promising business opportunities. The global chat app market is on track to reach a substantial $1.6 trillion by 2026, making the development of a successful chat app a potentially lucrative business endeavor.

1. *Objective:* The Software offers an intuitive Graphical User Interface (GUI), making it accessible to user with minimal system operation knowledge. It is designed for cross-platform compatibility, functioning seamlessly on various operating system, thus ensuring its usability on any system, regardless of the underlying OS. Remarkably, the messenger supports an unlimited number of concurrent users without compromising the server's performance. The primary objective of this project is to develop an instant messaging solution that facilitates seamless communication between users. Notably, the project places a strong emphasis on user-friendliness, ensuring that even those with limited technical experience can easily use the application.
2. *Problem Statement:* This endeavor aims to construct a chat application encompassing a server and user interfaces, facilitating interactive conversations among its users. The central objective is the development of a real-time messaging solution, designed to foster seamless communication between users, bridging gaps and enhancing collaboration. A primary focus lies in ensuring the project's accessibility, making it user-friendly to the extent that even those with minimal technical experience can navigate and employ it effortlessly. Importantly, this undertaking holds the potential to serve as a valuable asset within organizational settings, offering a means for employees to connect via Local Area Network (LAN), thus promoting efficient communication and teamwork.
3. *challenges:* The issue of scalability arises as the user base of a chat application expands. Scaling the system to handle the growing demand can be a complex task, potentially resulting in performance issues such as sluggish load times and message delivery problems.Security stands as a paramount concern in the realm of chat applications, necessitating robust measures to safeguard user data from unauthorized access. This challenge becomes more pronounced as chat applications evolve, introducing advanced features like file sharing and video calls.Achieving real-time communication in chat applications is another noteworthy challenge, particularly when dealing with varying network conditions and spanning considerable distances. Maintaining seamless and instantaneous interactions can be intricate.The user experience is a critical aspect of chat applications, demanding an interface that is both user-friendly and easy to navigate. This becomes increasingly challenging as chat applications incorporate more and more features, potentially complicating the user interface.In a competitive landscape with numerous established chat applications, breaking through and gaining recognition presents a formidable challenge for new entrants to the market.

# Matirals and Methods

Let's break down the functionalities within the provided Java code for a basic chat server using Swing methods:

Material Required:

1. Java Development Kit (JDK): Needed for compiling and running Java code.

2. Integrated Development Environment (IDE): Like IntelliJ IDEA, Eclipse, or NetBeans for code editing and compilation.

3. Images for Icons: Required images referenced in the code (e.g., "video.png", "phone.png") should be available in a directory accessible by the program.

4.Client Code:A corresponding client-side code that connects to this server and exchanges messages.

Explanation of Functionality:

Sending Messages:

User Input Handling:

When the user clicks the "Send" button (`actionPerformed` method), it retrieves the text from the input field (`JTextField`).

Creates a panel (`JPanel`) to format the outgoing message using `formatLabel`.

Adds the formatted message panel to the UI to display the sent message.

Writes the message to the `DataOutputStream` connected to the client socket to transmit the message.

Receiving Messages:

Server Setup:

The `main` method creates a `ServerSocket` that listens on port 6001 for incoming connections.

Upon a client connection, it creates a `Socket` for communication and sets up input (`DataInputStream`) and output (`DataOutputStream`) streams.

Continuously reads messages from the client using `DataInputStream`.

Formats received messages into panels using `formatLabel` and displays them in the UI.

Client-Side Code :

The client code connects to the server and allows users to exchange messages:

Establishes a connection to the server using a `Socket`.

Sets up communication streams (`DataInputStream` and `DataOutputStream`) to communicate with the server.

Manages UI display using Swing components for incoming and outgoing messages.

Integration Steps:

To combine the server and client functionality:

1. Ensure both the server and client classes (`Server` and `Client`) are within the same package (`chatui`).

2. Initialize both the server and client instances in the `main` methods of their respective classes.

3. Make sure the server is running before starting the client.

4. Establish a connection from the client to the server using the server's IP address (`127.0.0.1` for localhost) and the server's port (e.g., `6001`).

5. Once the client is connected, it can send messages to the server and receive messages from the server.

Conclusion:

The provided code sets up a basic server that listens for incoming client connections and exchanges messages using Java Swing.

It handles sending and receiving messages between the server and connected clients.

# Conclusion

Our exploration delved into the inner workings of server-client interactions in Java. We gained insights into crafting graphical user interfaces (GUI) through Java Swing,

enhancing our proficiency in the language. This project served as a valuable educational experience, deepening our understanding of Java, Java Swing, as well as

the dynamics of server-client communication in a comprehensible manner. Ultimately, our efforts culminated in the development of a fully operational and secure chat room server, offering users the capability to engage in private and confidential conversations.

##### Acknowledgments

We express our heartfelt thanks to our honorable Vice Chancellor Dr. C. MUTHAMIZHCHELVAN, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our Registrar Dr. S. Ponnusamy, for his encouragement.

We express our profound gratitude to our Dean (College of

Engineering and Technology) Dr. T. V.Gopal, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing Dr. Revathi Venkataraman, for imparting confidence to complete

my course project

We wish to express my sincere thanks to Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business

Systems and Course Coordinators for their constant encouragement and support.

We are highly thankful to our Course project Faculty Dr. Manickam. M, Assistant Professor, Department of Networking and Communications, School of Computing School of Computing, for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our HoD Dr. Annapurani Panaiyappan. K (Networking And Communication) and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

##### References

1. “Most popular global mobile messenger apps 2014 | Statistic,” Statista. [Online]. Available: http://www.statista.com/statistics/258749/most-popularglobal-mobilemessenger-apps/
2. “Secure Messaging Scorecard,” Electronic Frontier Foundation. [Online]. Available: https://www.eff.org/secure-messaging-scorecard
3. “Hacking attack on accounts connected with popular messenger app - Australia Network News (Australian Broadcasting Corporation),” Hacking attack on accounts connected with popular messenger app. [Online]. Available: http://www.abc.net.au/news/2014-06-19/an-asia-cyber-crime/5537194 .
4. “WeChat Developers,” WeChat API Documentation. [Online]. Available: http://dev.wechat.com/wechatapi/documentation .
5. “• Number of apps available in leading app stores 2014 | Statistic.” [Online]. Available: http://www.statista.com/statistics/276623/number-of-apps-availableinleading-app-stores/ .
6. “• Number of mobile messaging users worldwide 2014 | Statistic.” [Online]. Available: http://www.statista.com/statistics/369260/mobile-messenger-users/ .
7. “Wickr | Top Secret Messenger,” Wickr | Top Secret Messenger. [Online]. Available: https://wickr.com/
8. “unhcfreg | Viber Security Vulnerabilities: Do not use Viber until these issues are resolved.” [Online]. Available: http://www.unhcfreg.com/#!Viber-SecurityVulnerabilities-Do-not-use-Viber-until-these-issues-areresolved/c5rt/BB4208CF7F0A-4DE1-92A4-529425549683 .
9. S. Schrittwieser, P. Kieseberg, L. ̈Manuel, P. Fruhwirt, M. Mulazzani, M. Huber, and E. Weippl, “Guess Who’s Texting You? Evaluating the Security of Smartphone Messaging Applications.” [Online]. Available: https://www.sba- 36 | References research.org/wp content/uploads/publications/ndss2012\_final.pdf
10. C. Cattiaux, “iMessage Privacy,” 17-Oct-2013. [Online]. Available: http://blog.quarkslab.com/imessage-privacy.html .
11. A. Iqbal, A. Marrington, and I. Baggili, “Forensic artifacts of the ChatON Instant Messaging application,” in 2013 Eighth International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE), 2013, pp. 1–6.
12. “unhcfreg,” UNH Cyber Forensics Group Reveals Smartphone App Issues Affecting 968 Million. [Online]. Available: http://www.unhcfreg.com/. [Accessed: 03-Dec-2014]. [19] “SQLCipher has 100M+ Mobile Users (Thanks to WeChat!) | The Guardian Project,” SQLCipher has 100M+ Mobile Users (Thanks to WeChat!).[Online]. Available: https://guardianproject.info/2013/12/10/sqlcipher-has-300-millionmobile-users-thanks-to-wechat/ ..