CS540 Group 6 - Crabchat

Final Team Presentation

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Introduction & Motivation

- Developing a cutting-edge messaging application centered on user privacy.
- Addressing privacy concerns prevalent in existing messaging apps.
- Focusing on user data security without compromising performance.

Objectives

- Emphasizing client-side storage and encryption for heightened privacy.
- Improving performance and scalability using Tokio, a high-performance IO library.

Needfinding

Method: Semi-Structured Interviews

- Flexible
- Provided Qualitative Data
- Provided Pain Points & Opportunities

Crab Chat Focus: Privacy, essentials, and unique gaming.

Key Findings:

★ Usage: Daily utilization of Discord, Messenger, Snapchat.

★ Expectations: Emojis, media, group chats.

★ Privacy: Reject data commercialization; doubt competitor security.

★ Games: Mixed interest; prefer Wordle, Uno, Chess.

★ Features: Request voice recognition, customizable UI.

★ Goals: Emphasis on privacy, intrigued by gaming.

User Story/Scenario

User Story 1:

Bob wants to set up a chat room for his co-workers. However, he does not want to use external infrastructure for this. To do this, he sets up Crab Chat on an internal server, and distributes the information for his co-workers to join. He can ensure any confidential details are never exposed outside the company.

Goals: Set up a private chat room hosted on internal resources

User Story 2:

After the disappearance of Omegle, Charlie wants to set up an anonymous chat room for anyone to join. He sets up Crab Chat and posts the client and server information online. He is able to chat with any curious clients joining the server anonymously.

Goals: Set up an anonymous chat room to casually talk with other people.

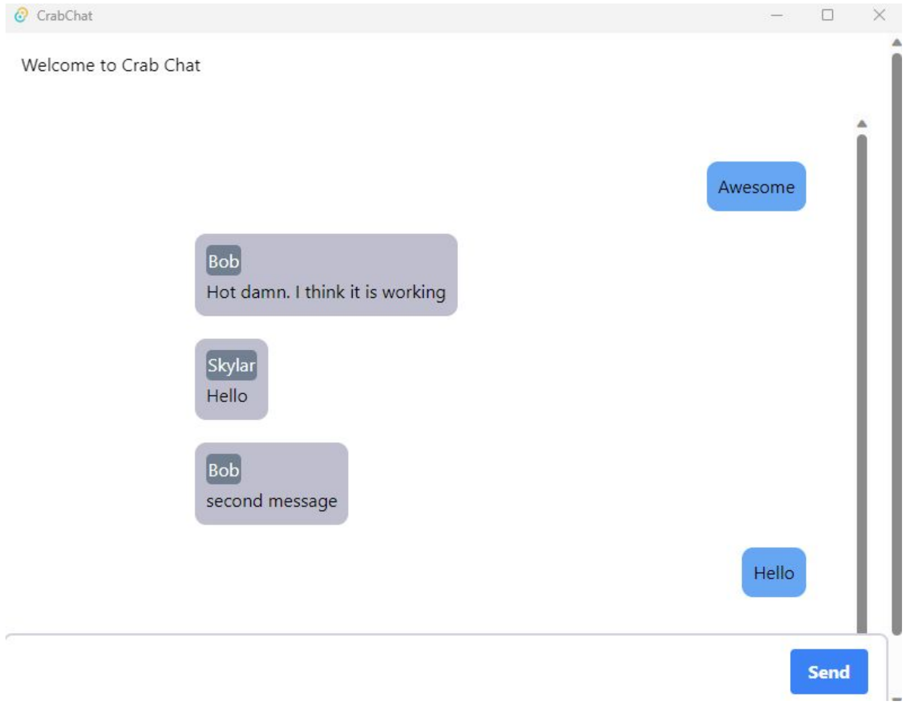
General Results

Final Product:

- Group chatting
- Secure communication
  - Client-side message storage
  - Encrypted messages
- UI that displays sent and received messages

If we had more time:

- Games (Chess, Checkers, etc)
- Emojis
- Replying to specific messages



Welcome to Crab Chat

Awesome

Bob
Hot damn. I think it is working

Skylar
Hello

Bob
second message

Hello

Process Retrospective

Opportunity to learn a new language and ecosystem of libraries or crates

Lessons Learned

1) Inter Module Communication Strategy Should Have Been Decided Upfront
2) Tauri with Tokio Introduced Unnecessary Complexity
3) Building vs. Extending Existing Architecture in Light of Using a new Language
4) Choice of Programming Language and Learning Curve
5) Testing Started with Units/Modules Instead of Integrated
6) Broad Project Scope and Overly Ambitious Features

Project Next Steps

- Implement One-on-one chatting
  - Currently just N=1 group / server
- End-to-end encryption for secure data transmission
- Simple text-based games
  - Wordle, Hangman, etc.
- Various user experience features
  - Specific message replying
  - Mentions/tagging specific users
  - Reactions
  - Chat Emojis
  - Simple user interface customization (light/dark mode)

Conclusion

- Initial goal was to create functional chat application & learn Rust
  - Completed with minimal viable product but *LOTS* of struggles
  - Learning Rust was very challenging
- Spent too much time “recreating the wheel”
- Should have decided architecture ahead of time for parallel development of separate modules
- Did not scope project well enough in beginning
  - Bit off more than development team could chew
- Many lessons learned about software development process