EchoLink: A LAN-based Online Chat Software Platform Utilizing WebRTC

Team Kind Koalas: Zhongsheng Wang, Kairui Liu, Xiaojun Xiao, Hongfei Rong, Siyu Lu, Shuzheng Mi, Jiahao Wang

All project plans were unanimously approved by the entire group after three rounds of discussions in Week 3.

Introduction

With the evolution of technology and changes in work methodologies, instant messaging software has become an essential tool for business communication and collaboration. However, existing communication solutions typically rely on internet connections and may not fully meet organizations' needs in terms of security, privacy, and cost-effectiveness. Addressing these challenges, our team plans to develop a LAN-based online chat software platform, EchoLink, utilizing WebRTC technology. This platform is designed specifically to meet the internal communication needs of small businesses.

Motivation

Our core objective is to provide a secure and reliable internal communication platform that allows users to conduct real-time text, voice, and video communications within a LAN environment, without depending on external internet connections. We believe that such a platform will offer users unique value through:

- Enhanced security and privacy: By operating within a LAN, data does not traverse external networks, significantly enhancing data security and privacy protection.

- Low-latency communication experience: Leveraging the advantages of LAN, we aim to provide a communication experience with lower latency than traditional internet solutions, making voice and video calls smoother.
- Ease of deployment and maintenance: Considering the IT resource constraints of small businesses, we are designing a solution that is easy to deploy and maintain, lowering technical barriers and operational costs.

Related work

Our team has intensively used and analyzed existing similar software & platforms in the market, summarizing their strengths and weaknesses as follows (focusing only on functionalities we can realize or improve, acknowledging that our team's capabilities cannot compare to a full development team):

Google Meet

Strengths:

- Seamless integration: Google Meet's integration with Google Workspace provides a seamless workflow for managing schedules, emails, and documents.
- Usability: A clean user interface and easy-to-understand control options enable users to quickly start or join meetings.
- Accessibility: As a browser-based solution, users can join meetings without needing to download an app, lowering the barrier to entry.
- Stable performance: Google's vast infrastructure ensures the stability and reliability of meetings, maintaining good communication quality even under poor network conditions.

Areas for improvement:

- Feature richness: Compared to Zoom, Google Meet has room for improvement in advanced meeting features like breakout rooms and meeting recording.

- Customizability: Corporate users may require more customization options to better integrate Meet into their internal communication systems. (This may not need to be considered for EchoLink, but it is one of its strengths.)

Slack

Strengths:

- Excellent collaboration features: Slack offers powerful messaging and file-sharing capabilities, supports cross-team collaboration, and enhances work efficiency through channels and private messages.
- Rich integrations: Supports integration with a variety of third-party services and tools, such as Trello, GitHub, and Salesforce, making Slack a centralized communication and collaboration platform for businesses.
- Customization and automation: Provides a wealth of customization options and bots, reducing the burden of repetitive tasks through automation. (This application of AI tools may not be within our development scope.)

Areas for improvement:

- User interface: For new users, Slack's user interface can seem complex, requiring a learning curve.
- Feature-based pricing: For some features that businesses use frequently, corporate membership services are often required, leading to relatively high expenses. (We aim to implement some of these features.)

Zoom

Strengths:

- High-quality video conferencing: Zoom is renowned for its high-quality video and audio conferencing capabilities, supporting large-scale video meetings and webinars.

- Powerful features: Advanced functionalities like breakout rooms, meeting recording, and screen sharing meet the diverse needs of users.

Areas for improvement:

- Security and privacy: Concerns about Zoom's information security continue to be an issue for some, as software vulnerabilities could potentially impact user frequency and intention to use. (Refer to blog link: https://cloudsecurityalliance.org/blog/2022/03/13/an-analysis-of-the-2020-zoom-breach)
- Resource consumption: The Zoom application may consume significant computer resources under certain circumstances, affecting the performance of other applications.

Summary

By summarizing the strengths and issues of the above software, we aim to integrate certain functionalities into EchoLink and optimize for weaknesses:

- Security and privacy protection: The security and privacy issues highlighted by Zoom remind us to prioritize security in software design. Implementing end-to-end encryption, ensuring data is transmitted within the LAN, and providing transparent privacy policies are key measures to enhance user trust.
- User interface and experience: EchoLink will pursue a simple and intuitive design, reducing the difficulty for users to understand interaction logic and minimizing cumbersome and meaningless operation flows, to achieve the best user experience.
- Performance and resource consumption: Considering Zoom's potential for high computer resource consumption, we will focus on software performance optimization during development to ensure smooth operation across various hardware configurations and reduce system resource usage. (This is the part of development we hope and expect to have technical capabilities)

Through a comprehensive analysis of these core requirements and improvements on existing issues with competitors, EchoLink aims to offer a secure and efficient internal communication solution that meets the specific needs of small businesses in data security, communication quality, and team collaboration.

Requirements

Based on our team discussions and a rational analysis of each team member's technical stack, we have outlined the functionalities that our team can implement in EchoLink. We will attempt to learn and solve some of the potential technical challenges during the future development cycle. We have categorized the functionalities into the following parts: Must-Haves, Should-Haves, Could-Haves, Nice-to-Haves. (Due to word count limitations, this is just a brief listing of functionalities.)

Must-Haves

- User authentication and management: Implement basic user registration, login, and personal information management functionalities to ensure the security and uniqueness of user identities.
- Real-time communication features: Including text chat and the basic functionality of sending pictures and simple files, to meet users' fundamental communication needs.
- Security and privacy protection: Implement basic data encryption and user authentication to ensure communication security.
- Message storage and history: Store and display users' chat history, providing basic viewing and search functionalities.

Should-Haves

- Interface design and user experience: Intuitive and friendly user interface design, including clear chat windows and contact lists.
- Cross-platform compatibility: Support basic versions for major operating systems (Windows, macOS) to ensure the software can be used on multiple platforms.
- Network and performance optimization: Ensure a smooth communication experience in a stable LAN environment.

Could-Haves

- Voice and video call functionalities: Provide simple voice or video call options on top of basic chat functionalities.
- Message notifications and reminders: Offer notification and reminder functionalities for new messages to enhance user experience.
- Extended file transfer functionalities: Provide more comprehensive file transfer progress indications and support for more file types.

Nice-to-Haves

- Broader cross-platform support: Expand to mobile platforms like iOS and Android to increase software accessibility and convenience.
- Advanced message management: Offer more advanced chat history management features, such as tagging, categorization, advanced search, etc.
- Enhanced network stability: Further optimize the software to perform well in unstable network conditions, improving user satisfaction.

Technologies

To optimize the development process of EchoLink and ensure its performance and availability, our team has decided to make some minor changes to the traditional MERN stack to better meet the needs of instant messaging and multimedia data transmission, while considering future scalability and cross-platform compatibility. Below are the necessary technical stacks for the development of EchoLink and potential optimizations we are considering for the future.

Core Technology Stack

- React.js: As the core framework for front-end development, React not only provides a rich ecosystem but also offers excellent compatibility with real-time communication technologies like WebRTC, facilitating the rapid development of high-quality user interfaces and interaction designs.

- Node.js: Leveraging its non-blocking I/O model and solid support for WebRTC and WebSocket, Node.js will serve as our signaling server development platform, ensuring efficient and stable real-time communication.
- Coturn (Reference link: https://github.com/coturn/coturn): Choosing Coturn as our STUN and TURN server, based on its popularity in the networking service open-source projects, provides reliable NAT traversal services, which are crucial for enhancing the stability and accessibility of P2P video conferencing applications.

Advanced Technology Planning

To further enhance user experience and system performance, we plan to introduce the following advanced technologies in the later stages of the project, time and resources permitting:

- GoLang: Considering GoLang's exceptional performance in concurrent processing and network services, especially its native concurrency model goroutine, we plan to use GoLang to refactor our signaling server as well as STUN and TURN servers, aiming for higher performance and efficiency.
- React Native (Reference link: https://reactnative.dev/): By using React Native, we will be able to extend the front-end web into cross-platform applications, allowing users to enjoy a seamless EchoLink experience on mobile devices.
- WebAssembly (WASM) (Reference link: https://developer.mozilla.org/en-US/docs/WebAssembly): Consider using WASM to add special effects to video processing, which can efficiently perform complex computing tasks on the client, such as image processing, to further enrich the user's interactive experience.

Project management

To ensure the project proceeds efficiently and ensures delivery quality, we will adopt agile development strategies to improve teamwork, adaptability and rapid feedback capabilities. Here are the specific details of our project management strategy:

Scrum Framework

- Sprint Planning: The project will be divided into several two-week sprints. Before each sprint, we will hold a planning meeting to clarify the goals and task list for the next two weeks. Everyone needs to record their plans in detail and tag them accordingly for review at the end of the sprint.
- Sprint Review and Retrospective: At the end of each sprint, we will hold a review meeting to present and evaluate the work completed during the stage. This is followed by a retrospective meeting to discuss how to improve efficiency in the next sprint. The Team Leader for each sprint will rotate among team members to avoid the additional pressure of having one person in the role for too long.

Team Communication and Coordination

- Communication Tools: To ensure effective communication among team members, we will primarily use WeChat for daily message exchanges and file sharing. We plan to hold full-team meetings every Monday and Friday to plan for the week and conduct timely reviews. We have created a weekly Zoom Meeting room for online situations, although we generally meet face-to-face.
- Project Management: We choose GitHub KANBAN as our project management platform, using Issues to track development tasks and bugs, employing a custom label system for task classification, setting milestones to plan key stages, and organizing the workflow with a project board.

Version Control Strategy

- Git and GitHub: We use Git for code version control and GitHub as our code repository hosting platform.
- Branching Strategy: We will adopt a feature branching strategy, where each new feature or fix is developed on a separate branch and merged back into the main branch through PRs to ensure the stability of the main branch and facilitate code review. Each merge will require a code review from a team member not involved in the branch's development to ensure code stability.

EchoLink's development plan consists of several sprints, with the specific number flexible based on project progress and team capability. Through the above management strategies, our team will be able to collaborate efficiently while ensuring development quality and quickly responding to project needs changes.

Risks in team projects

In team projects, risks are often multifaceted. Effectively identifying and mitigating these risks is crucial for ensuring project success. Below are the risks our team has identified through detailed discussions and the corresponding mitigation strategies:

Technical Risks

Technical risks involve the possibility that the technology used in the project may not meet expectations or may be more difficult to implement than anticipated (as we have only outlined the basic requirements of the software). For our EchoLink product, potential technical risks include, but are not limited to, compatibility issues with WebRTC technology, performance issues, or security vulnerabilities (even as a local LAN development project, there are still inherent risks).

To this end, we have made the following contingency plans:

- Conduct prototype design and testing of key technologies early in the project to assess their suitability and performance. For non-essential functions listed above, we will make modifications after collective group discussion and evaluation.
- Regularly review the code and architecture for security to ensure timely discovery and fixing of security vulnerabilities. (This is necessary, as not everyone is fully proficient in development, and sufficient time must be allotted for debugging.)

Project Management Risks

Project management risks involve the scope, timing, and cost of the project being affected by unreasonable arrangements. This also includes sudden events among team members, skills

mismatches, or uneven work distribution, which could lead to project delays or stagnation.

To this end, we have made the following contingency plans:

- Clarify the scope of the entire project to avoid unlimited expansion of the project and failure to complete it. Potential future project optimizations listed will be considered after the entire project is implemented.
- Regularly conduct project reviews and retrospectives, adjusting plans timely based on project progress and personnel feedback. We plan to hold two meetings a week on Monday and Friday for task planning and review, with each meeting summarized by rotating members (aided by AI tools) for subsequent tracking.
- Clearly define each team member's role and responsibilities to ensure reasonable task distribution. (No clear technical division of labor yet, PM, UI/UX, Front-End, Back-End, Tester will be taken by those proficient in related tasks)
- Establish backup plans for each person and task to guard against sudden personnel changes and accidents.