Individual Report: Chat Application [Reflection]

Author: *Dalton Christopher*
ID: *A00122255*
Course: *PBT205—Project-based Learning Studio: Technology*
Assessment: *3*
Date: *08/2024*

Project Repository:

- *DaltonChris/A1 PBT205 TUA at A3 Chat App (github.com)*

Overview

Throughout the project I was the lead developer, responsible for managing the repository, extending the application's functionality, and refining the existing code-base. Our project aimed to create a user-friendly chat platform, incorporating essential features such as message exchange, user notifications, and image attachments.
The following report aims to outline the key functionalities and improvements I developed, with focus on the attachment and image-handling capabilities.

Prototype and Development Approach

The prototype of the chat application provided basic features, including text messaging and real-time communication between users in a shared chatroom. However, the prototype was limited in terms of user interaction, multimedia support and lacked an Online users list. To address these limitations, I undertook the task of extending the application’s capabilities, particularly focusing on the addition of image attachment functionality.

Image Attachment and Handling

To extend the functionality I implemented was the ability for users to send and receive images within the chat.
Achieving this involved several key tasks:

- **File Attachment Functionality**: I added an "Attach" button, allowing users to select image files On there local storage using the file explorer.
  Supporting common image formats, “.jpg”, “.jpeg”, “.png”, and “.bmp”
- **Image Encoding**: To send images via the chat, I developed a method to convert image files into base64-encoded strings by first getting the bytes. This approach ensured that images could be efficiently sent through the messaging protocol without compromising their integrity.
- **Image Rendering in Chat**: When a user receives an image message, the application decodes the base64 string and renders the image within the chat list box.
  To ensure a seamless user experience, images are scaled but maintain their aspect ratio; Preventing overlap with text messages and ensures that the chat presentation is quality.
- **Image Viewing**: Users can click on an image in the chat to view it in its full scale. This feature was implemented by creating a popup window that displays the image at its original size.

UI Improvements and User Experience

To improve the user experience, I used Visual-Studio’s Forms Designer to layout the user interface. This allowed for a more intuitive layout and interaction model, making the application easier to navigate and use. Additionally, I added a custom icon for the application, adding a nice touch to the app.

User Management and Notifications

The application now features a list of active users in the chatroom, The list is automatically updated based on User join messages.
A message “’User’, has joined the chat” is sent to all clients when a user connects, enhancing the sense of community within the chat.

Summary

Throughout this project, my focus was on enhancing the existing chat application by adding these functionalities, refining the user interface, and improving the overall user experience. While Also Removing classes and parts of our projects other prototypes keeping a clean a simple code base.
This project from early prototyping to choosing an app to expand on greatly improved my ability collaborate in a group, while also improving my understanding of managing client–server relationships.