Distributed Computing (COMP3008)

Assignment 1

Part B

Individual Report

K. Joel Shameera Kumara

20963675

Introduction

I'm Joel Kumara, and I was essential in the success of our team's Chat App project. I'll discuss my contributions to the project, my responsibilities, and my personal reflections in this report. My overall contribution to our two-person team—where emphasis is placed on equal involvement—is almost equal to 50%.

To ensure the success of this endeavor, I took on a variety of tasks and duties. My contributions to the project covered a wide range, from building the server and database to ensuring Windows communication by developing the Private Messaging Window and instances to creating the user interface for the chat application's login window, chat room selection window, and user windows.

It required a considerable commitment of time, effort, and competence to carry out these duties. I worked closely with my fellow team members to realize our group's shared vision.

I will go into detail about the precise techniques we used and the difficulties we ran across while working on the project in the parts that follow. I'll also share details about our decision-making procedures and the methods we used to get through obstacles. I will also discuss the dynamics of teamwork in our group, such as the communication challenges we encountered and the successful strategies we used to promote fruitful cooperation.

The goal of this report is to give a thorough overview of my contributions to ChatApp while highlighting the commitment and effort that our entire team put into this project.

Methodology and Challenges

We used a thorough methodology in our ChatApp project that included the creation of three separate projects within a single outcome. These projects featured a Console App for server hosting, a Windows Presentation Foundation (WPF) application, and a separate Dynamic Link Library (DLL) project for shared features.

Project Structure:

1. WPF Application (ChatAppClient):

The development of an engaging and user-friendly chat experience was our main goal and the inspiration for the ChatAppClient project. This WPF (Windows Presentation Foundation) application served as the ChatApp's main building block and assumed prominence in the following crucial areas:

- User Interface Design:

The aesthetic and interactive components of our ChatApp were created using the WPF application. User interfaces, chat windows, and messaging features all originated in this creative hotspot. It created a visually appealing and user-friendly platform for communication through careful design..

- Real-Time Communication:

Our ChatAppClient's core functionality was real-time communication. The complexity of instant message delivery was managed by this project, guaranteeing that users had seamless and immediate interactions with their peers. It was crucial in generating the sensation of taking part in a dynamic discourse..

- Multiple Windows:

  Our ChatApp's ability to effortlessly manage several chat windows was a remarkable feature. The WPF Application made it possible for these windows to be dynamically created and managed, whether they were for private communications or group conversations. The Private Message Window (DMWindow), where numerous copies of the same window were created to give users a customized and effective messaging experience, was where this was most obvious.

- Dynamic Refresh:

  The real-time updating of chat messages was a crucial component of our ChatAppClient. The WPF Application's features allowed conversations to automatically reload anytime a new message received. The entire user experience was improved since this functionality eliminated the need for manual refresh or pull requests..

- Intuitive User Experience:

  Users were able to quickly browse the WPF Application, start chats, manage contacts, and have frictionless interactions thanks to the user-centric design of the application. It was crucial to ensuring that our ChatApp was usable and pleasant for users of all technological backgrounds.

In summary, the ChatAppClient project functioned as the entrance to our ChatApp by offering an aesthetically pleasing, real-time, and user-centric platform for communication. A totally immersive chat experience is made possible by its cutting-edge features, such as dynamic window management and automatic message refresh..

2. Console App (ChatServer):

The Console App project was crucial to the ChatApp ecosystem since it provided the framework for the system as a whole. These were the main duties it was in charge of.:

- User Connections:

Clients could securely connect to the server since the Console App scrupulously regulated user connections. In order to enable seamless user interaction within the ChatApp, it created a secure channel for data exchange.

-Chat Room Functionalities:

This element was in charge of setting up and running the chat rooms where users may join, converse, and send messages. It preserved the integrity of chat rooms, allowing members to have productive conversations, share knowledge, and collaborate.

- Message Routing:

The server quickly routed communications between users, ensuring that messages were delivered accurately and on time. It managed the difficulties of message distribution, ensuring that members in a chat room received the messages intended for them.

- System Reliability:

The Console App, as the ChatApp's backbone, ensured the system's dependability and stability. It monitored connections, maintained resources, and dealt with any problems, all of which contributed to a seamless and uninterrupted user experience.

In essence, the Console App (Server) served as the foundation of our ChatApp, organizing user interactions, ensuring data integrity, and nurturing a strong and stable communication platform for all users.

3. DLL Project (ChatDataTier):

The ChatDataTier DLL Project served as the backbone of our ChatApp, containing important functionality and utilities that were critical in delivering a fluid and feature-rich user experience. This project was created with the utmost care and attention to detail, spanning a wide range of critical components.:

- Data Handling:

ChatDataTier was responsible for efficiently managing data within the ChatApp. It handled user profiles, chat room information, and message histories, ensuring that user data was stored securely and retrieved as needed.

- User Authentication:

This component provided strong user authentication procedures, ensuring that only authorized users could access the ChatApp. It validated user credentials, kept track of user sessions, and prevented unwanted access.

- Message Synchronization:

ChatDataTier made real-time message synchronization possible for all users and chat rooms. It ensured that messages were delivered on time and reliably, creating lively and involved discussions.

- User Management:

    The DLL project was critical in the management of user accounts, chat room memberships, and other user-related processes. It allowed for the construction and deletion of chat rooms, the addition and removal of users, and overall user management.

ChatDataTier served as the heart of our ChatApp, providing it with the tools and capabilities needed for user administration, message handling, and data integrity. It highlighted our commitment to providing our users with a strong and extensible chat platform.

We drew inspiration for the ChatApp from a variety of places. We used Microsoft's WPF instructions and tutorials to create a beautiful and responsive user interface. We also investigated open-source chat applications on GitHub, where we discovered useful code snippets and design patterns. Stack Overflow was a valuable resource for resolving code challenges.

Reflection

Teamwork Reflection:

Challenges

Communication Difficulties: In a small team, one might expect communication to be simple. However, we encountered difficulties with synchronous and asynchronous connection. We required quick input or assistance at times, but we also had to wait for the other person to become available. These inconsistencies occasionally caused delays and misunderstandings.

Solutions

Solution Through Availability Planning:

> To address these communication issues, we created a mechanism that clearly indicated our availability. We agreed on certain times for each team member to work on the project. We made certain that we were both online and responsive to messages during these times. This methodical approach assisted us in aligning our schedules and addressing significant issues as soon as possible.

Balancing Workloads:

> Another aspect of effective teamwork was workload balancing. Due to the nature of our tasks, there were times when one team member had more responsibilities than the other, which could have caused frustration, but we addressed it by maintaining open and honest communication about our workloads. If one of us was overwhelmed, we would discuss it and redistribute tasks accordingly.

In conclusion, while working in a small group brought unique challenges, our dedication to organized communication and task transparency enabled us to overcome these obstacles. This experience underlined the necessity of teamwork and adaptation in accomplishing project objectives..

References

A References:

We mostly relied on official documentation, open-source libraries, and online communities for advice and problem-solving when developing our ChatApp. While we did not directly mention external sources in our code, the resources listed below were important in developing our knowledge and approach to various elements of the project:

1. Stack Overflow : The Stack Overflow community was extremely helpful in troubleshooting and fixing code challenges. We frequently discovered answers and ideas from experienced developers who had faced similar problems.

2. GitHub : We looked through open-source projects and GitHub repositories to see how others had incorporated specific functionality or design principles. This aided us in optimizing our code and learning from industry best practices.

3. ILectures and Lecture slides : The course slides were crucial in the creation of the Assignment's user interface.