Individual Report

Assignment 1 – Distributed Computing

Victor Marteli

Chat Application

Contents

[Introduction 2](#_Toc146366052)

[Methodology and Challenges 3](#_Toc146366053)

[Technical Challenges and Solutions 4](#_Toc146366054)

[Inspirational Sources 4](#_Toc146366055)

[Reflection 5](#_Toc146366056)

[Conclusion 5](#_Toc146366057)

[References 6](#_Toc146366058)

# Introduction

**Team Composition**

In the group project, I collaborated with two other team members: Jinwoo Kim and Harley Wilson. We made a concerted effort to distribute the workload equitably, with each team member being accountable for approximately 33.33% of the overall project. My responsibilities were multifaceted and encompassed both the development and implementation of various features, as well as the management of the project's database and server-side functionalities.

**My Role**

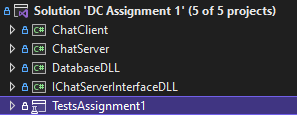
I was responsible for the following tasks:

* Implementing full file sharing functionality, covering both the GUI and back-end, for .bmp and .txt files.
* Designing the GUI for the private messaging feature.
* Implementing the database and creating a test harness for it.
* Adding logging capabilities to the server.
* Various bug fixes and quality-of-life improvements.
* Setting up Jira for streamlined collaboration.

# Methodology and Challenges

**Project Components**

Our software engineering project was strategically divided into five key components to ensure a modular and maintainable codebase:



1. **IChatServerInterfaceDLL**: This Dynamic Link Library (DLL) acted as an interface and established a contract for the implementation of the chat server. It was crucial for ensuring that the server adhered to predefined specifications, thereby facilitating easier debugging and future scalability.
2. **Database DLL**: In this component, I employed Language-Integrated Query (LINQ) to execute SQL-style queries for data storage and retrieval. This approach negated the need for an external SQL server and streamlined the data management process.
3. **ChatClient**: This was the user interface layer of the project, built using XML for layout and C# for logic. It contained various functionalities like button-click events that communicated with the server to execute specific tasks.
4. **ChatServer**: I was instrumental in developing this server layer, designed to handle multiple client requests concurrently. It was built to be robust and scalable, capable of managing a high volume of data transactions.
5. **TestsAssignment1**: I created this testing component to validate all database-related methods. It was integrated with C#'s native testing framework, ensuring a seamless testing experience.

# Technical Challenges and Solutions

One of the initial challenges was the inefficiency of our database, which used a singleton class to store hashed strings. This approach was cumbersome and lacked scalability. I refactored the database, implementing a more efficient data storage and retrieval mechanism.

Another challenge was the need for effective version control and task management. To address this, I initiated the use of Jira for task tracking and Git for version control. This significantly improved our workflow, reduced duplicated efforts, and enhanced team collaboration.

### Inspirational Sources

The methodology and design choices were inspired by various resources. Microsoft's C# tutorials were particularly helpful in overcoming specific technical challenges. Additionally, various online forums and coding communities provided insights into best practices and alternative approaches to problem-solving.

# Reflection

**Challenges in Communication**

Initially, our team faced significant challenges in communication. Despite having a GitHub repository, there was a lack of clarity on task ownership, leading to duplicated efforts and wasted time. This was a critical issue that needed immediate resolution to prevent further inefficiencies.

**Solutions and Improvements**

To mitigate these challenges, we implemented a Jira project board, categorizing tasks into "To Do," "In Progress," and "Completed." This provided a visual representation of our workflow and helped in task allocation. Furthermore, we established a Discord server for real-time communication, which proved invaluable for quick problem-solving and decision-making.

Git was extensively used for version control, with a strict policy against direct pushes to the main branch unless they were minor fixes. All major changes were made through pull requests that underwent peer review before merging, ensuring code quality and consistency.

# Conclusion

This report serves to detail my individual contributions to our group software engineering project, the methodologies employed, the challenges faced, and how they were overcome. I am pleased with the final product and believe that our team's collective efforts have resulted in a robust and scalable application.

# References

1. “How to: Open files with the OpenFileDialog - learn.microsoft.com.” Microsoft Learn. Accessed September 20, 2023.
2. “C# unit test tutorial - Visual Studio (Windows) | Microsoft Learn.” Microsoft Learn. Accessed September 20, 2023.
3. “Language Integrated Query (LINQ) in C# - C# | Microsoft Learn.” Microsoft Learn. Accessed September 18, 2023.