

Державний вищий навчальний заклад
«Запорізький Інженерний навчально-науковий інститут»

Кафедра програмного забезпечення автоматизованих систем

КУРСОВА РОБОТА

з об'єктно-орієнтованого програмування

на тему Web-Конференція

Студента 2 курсу, групи 6.1210-1

Керівник:

доц. Попівцій Василь Іванович

м. Запоріжжя – 2021 рік

Завдання

Потрібно розробити об'єктну модель програмного забезпечення web-конференції.

Web-конференція являє собою сховище повідомлень у мережі Інтернет, доступ до якого здійснюється за допомогою браузера. Для кожного повідомлення конференції зберігаються значення наступних полів: номер повідомлення, автор, тема, текст повідомлення, дата додавання повідомлення, посилання на батьківське повідомлення. Початковою сторінкою конференції є ієрархічний список повідомлень. Верхній рівень ієрархії становлять повідомлення, що відкривають нові теми, а підрівні становлять повідомлення, отримані у відповідь на повідомлення верхнього рівня. Повідомлення-відповідь завжди має посилання на вихідне повідомлення. У списку відображаються тільки теми повідомлень, їх автори й дати додавання. Переглядаючи список, користувач вибирає повідомлення й по гіперпосиланню відкриває сторінку з текстом повідомлення. Крім тексту на цій сторінці відображається список (ієрархічний) повідомлень, що є відповідями, відповідями на відповіді і т.д. Для зручності користувачів необхідно передбачити пошук повідомлень по автору або по ключових словах у темі або тексті повідомлення.

Повідомлення додаються в конференцію зареєстрованими користувачами, які при відправленні повідомлення повинні вказати своє ім'я й пароль. Реєструє нових користувачів модератор конференції - її ведучий. При реєстрації користувач заповнює спеціальну форму, уміст якої потім пересилається модератору й запам'ятовується в базі користувачів. Модератор вирішує, реєструвати користувача чи ні, і відправляє свою відповідь.

При додаванні повідомлень користувач має можливість почати нову тему або відповісти на раніше додані повідомлення. Після додавання повідомлення воно доступно для читання всім користувачам (навіть незареєстрованим), і список повідомлень оновлюється.

Модератор має право по тим або іншим причинам видаляти повідомлення будь-яких авторів. Він також може карати користувачів, що порушують правила поведінки в конференції, позбавляючи на якийсь час користувача можливості додавати й редагувати повідомлення.

Зміст

КУРСОВА РОБОТА	1
Завдання.....	2
Вступ	4
Ціль роботи.....	4
Аналіз специфікації вимог технічного завдання.....	5
Об'єктно-орієнтований аналіз досліджуваної проблеми	5
Об'єктно-орієнтоване проектування програмної системи.....	7
Розробка програмної системи	7
Комунікація клієнта та сервера	7
Обробка подій графічного інтерфейсу	7
Об'єктна модель	7
Аутентифікація користувача.....	8
Тестування і результати виконання програми	8
Висновки	16
Перелік джерел посилання	16

Вступ

Ціль роботи

Ціллю роботи є створення додатку, що дозволить обмін повідомленнями між зареєстрованими користувачами за допомогою засобів платформи WPF та мови програмування C#, вивчення цих засобів у поєднанні з об'єктно-орієнтованим підходом до створення додатків.

Аналіз специфікації вимог технічного завдання

За завданням необхідно розробити додаток Web-конференції, який надаватиме користувачам можливість спілкуватись один з одним створюючи та відповідаючи на повідомлення. Всі користувачі мають бути авторизовані.

Об'єктно-орієнтований аналіз досліджуваної проблеми

Для аналізу проекту були створені діаграми класів та сценаріїв використання:

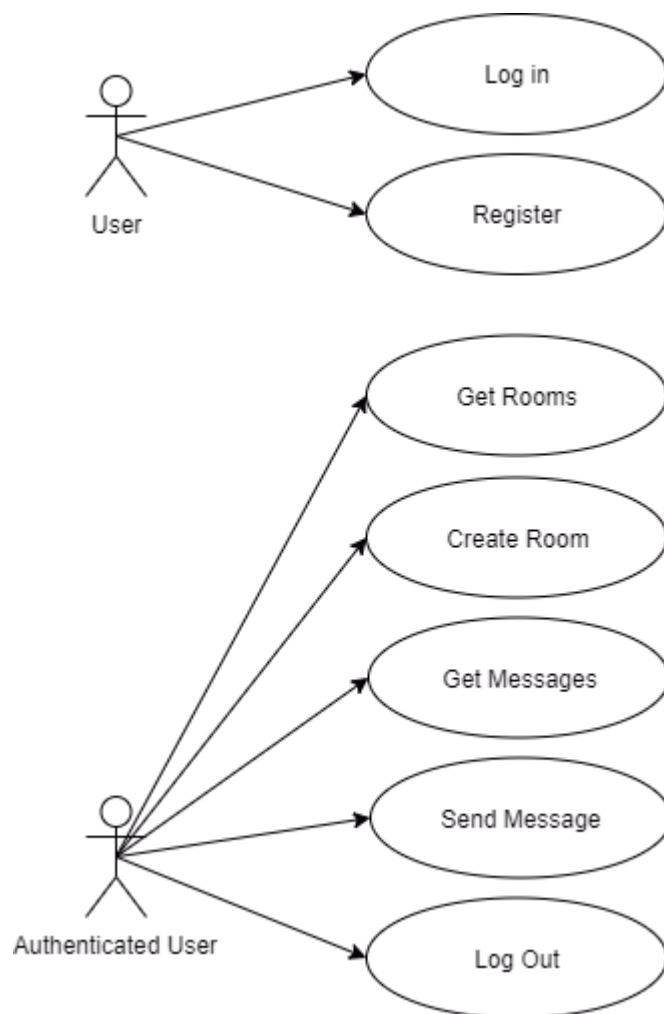


Рисунок 1 Діаграма Use Case

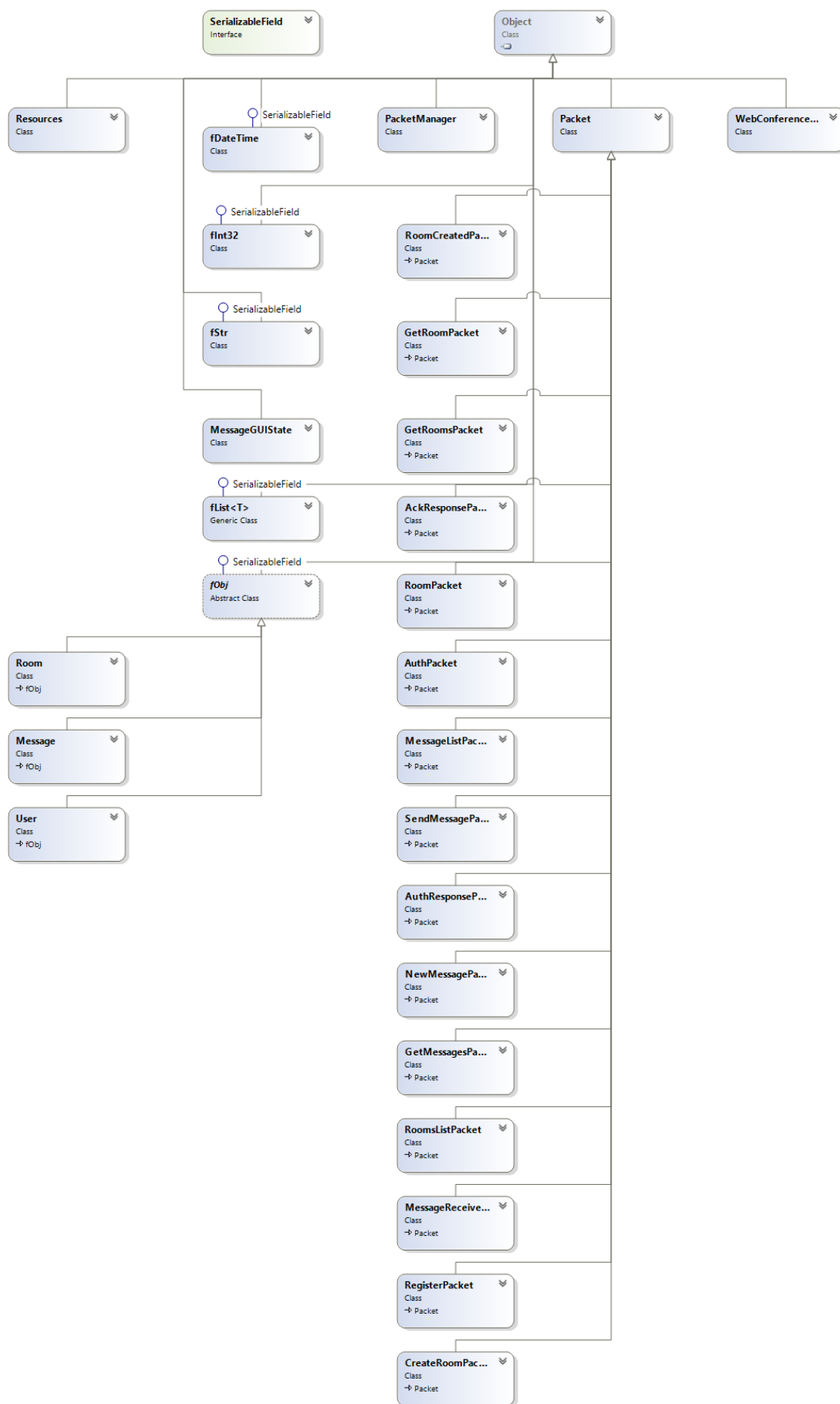


Рисунок 2 Діаграма класів

Об'єктно-орієнтоване проектування програмної системи

Система має клас `WebConferenceModel`, що відповідає за обробку дій, що були зроблені за допомогою інтерфейсу клієнту. Виклики методів цього класу відповідають надсиланню пакетів на сервер для подальшої обробки, редагування бази даних та повернення відповіді.

Розробка програмної системи

Комунікація клієнта та сервера

Комунікація між клієнтом та сервером здійснюється за допомогою надсилання пакетів. Будь-який пакет має реалізовувати клас `Packet`, що має методи `byte[] serialize()` `static Packet deserialize(byte[])`. Цей клас також містить в собі поле `packetID_field`, що є ідентифікатором цього пакету. При надсиланні пакета відбувається збереження ідентифікатора пакета та об'єкт `TaskCompletionSource`, що містить майбутню відповідь на цей пакет, що може бути отримана через очікування `Task<Packet>`, що буде повернений методом `PacketManager#send(Packet)`. Потім відбувається його серіалізація в байти та запис у сокет. При отриманні пакета зчитані байти передаються спочатку в статичний метод `Packet::deserialize` для створення об'єкта пакета, після чого `Task<Packet>` створений при надсиланні відповідного пакету буде помічений як виконаний, і `await PacketManager#send(Packet)` поверне відповідь.

Обробка подій графічного інтерфейсу

При виникненні події, спричиненою активністю користувача в графічному інтерфейсі відбувається виклик відповідного обробника класу об'єктної моделі `WebConferenceModel`. Після завершення обробки події зміст графічного інтерфейсу на екрані оновлюється щоб відповідати всім змінам, що відбулись.

Об'єктна модель

Програма включає в себе такі класи для роботи з даними:

- `common.Message:fObj` (репрезентує повідомлення, має поля ідентифікатор, автор, текст, дата, ідентифікатор попереднього повідомлення)
- `common.Room:fObj` (репрезентує чат, має поля ідентифікатор, ім'я та власник)
- `common.User:fObj` (репрезентує користувача, має поля ідентифікатор, ім'я користувача)
- `WebConfServer.database.LoadedMessage:Message, LoadedObject` (Репрезентує об'єкт повідомлення на сервері)

- `WebConfServer.database.LoadedRoom:Room, LoadedObject`
(Репрезентує об'єкт чату на сервері, має додаткове поле список членів)
- `WebConfServer.database.LoadedUser : User, LoadedObject`
(Репрезентує об'єкт користувача на сервері, має додаткові поля хеш пароля та токен сесії)

Всі класи, що перелічені вище, реалізують інтерфейс `common.field.SerializableField`,

Що дозволяє їм бути серіалізованими та десеріалізованими при пересиланні по мережі та збереженні.

Аутентифікація користувача

Для аутентифікації користувач має ввести логін та пароль на відповідному екрані входу в систему. Після надсилання даних сервер застосовує хеш-функцію до пароля та перевіряє наявність користувача з такими даними в базі даних. Якщо користувача знайдено, до пакета відповіді буде додано поле `auth_token_field`, що містить токен, що збережений на сервері і відповідає цьому користувачу. Таким чином, наступні запити користувача будуть авторизовані.

Тестування і результати виконання програми

Розберемо застосунок на прикладі двох користувачів.

Перший крок – реєстрація.

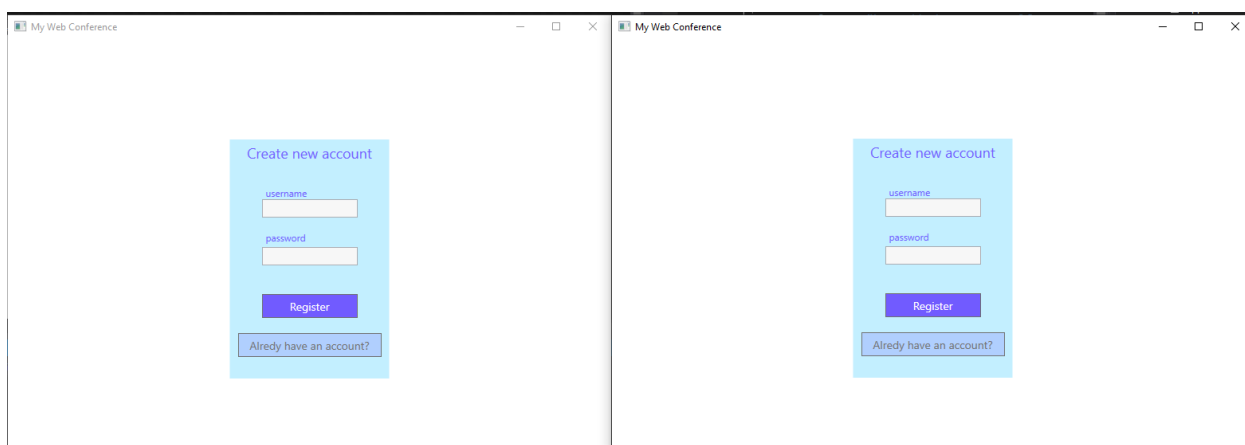


Рисунок 3 Реєстрація користувачів

Для цього користувач має ввести бажане ім'я користувача та пароль у відповідну форму. У разі, якщо ім'я вже зайняте, користувач побачить повідомлення про помилку.

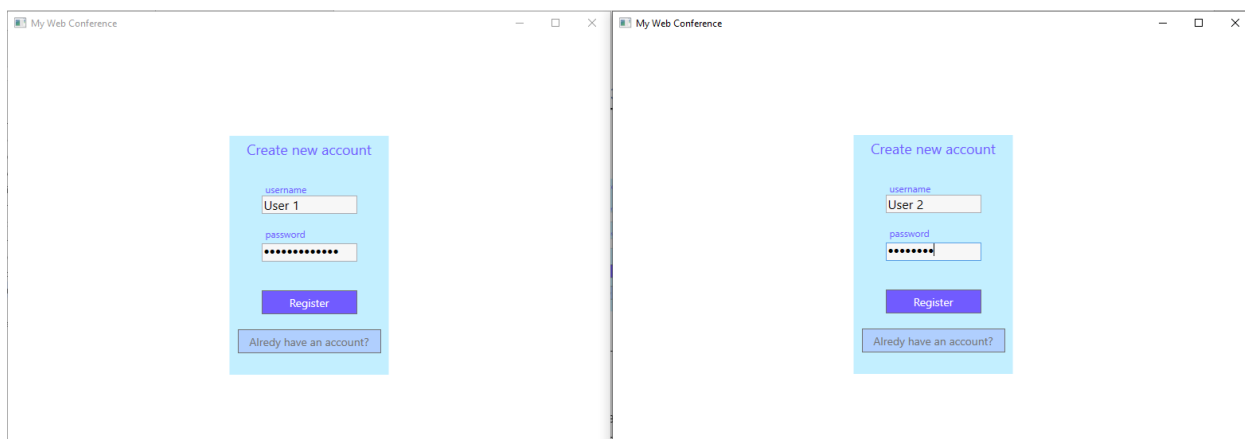
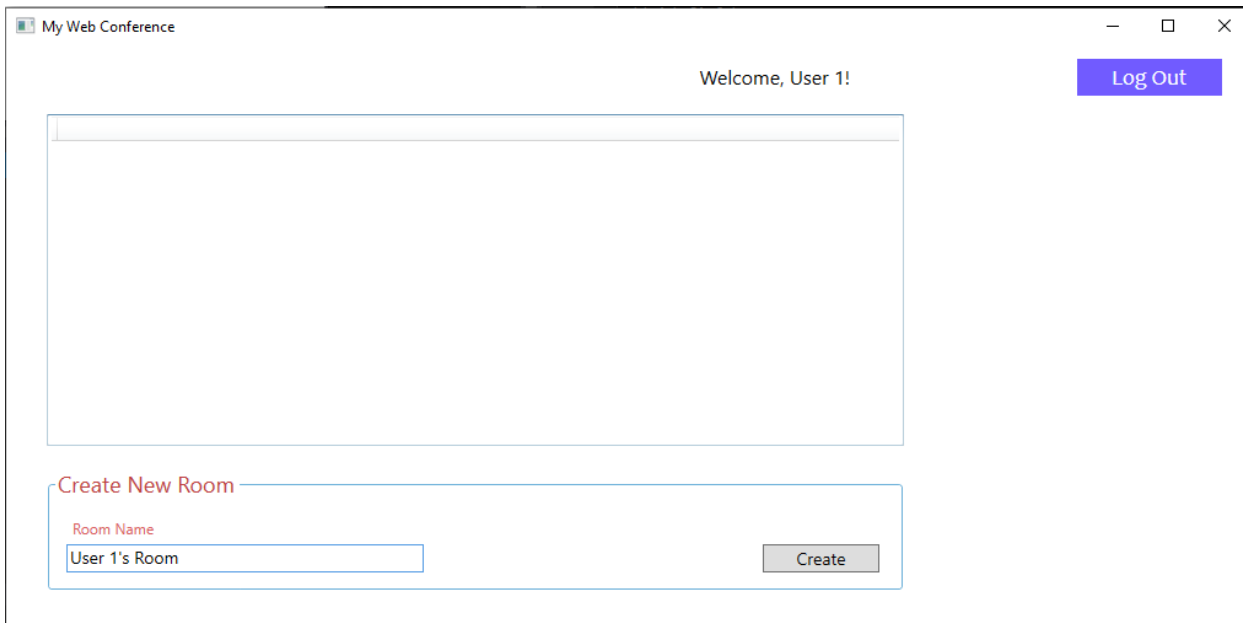


Рисунок 4 Введения данных

Далі користувач побачить список всіх доступних кімнат та форму для створення нової. Щоб створити нову кімнату, достатньо ввести її ім'я та натиснути «Create»



The screenshot shows a web browser window titled "My Web Conference". The page has a header with "Welcome, User 1!" and a "Log Out" button. Below the header is a large empty rectangular area. At the bottom, there is a "Create New Room" section. This section contains a label "Room Name" above a text input field that contains the text "User 1's Room". To the right of the input field is a "Create" button.

Рисунок 5 Створення нового чату

Після створення кімнату зможуть побачити всі користувачі застосунку.

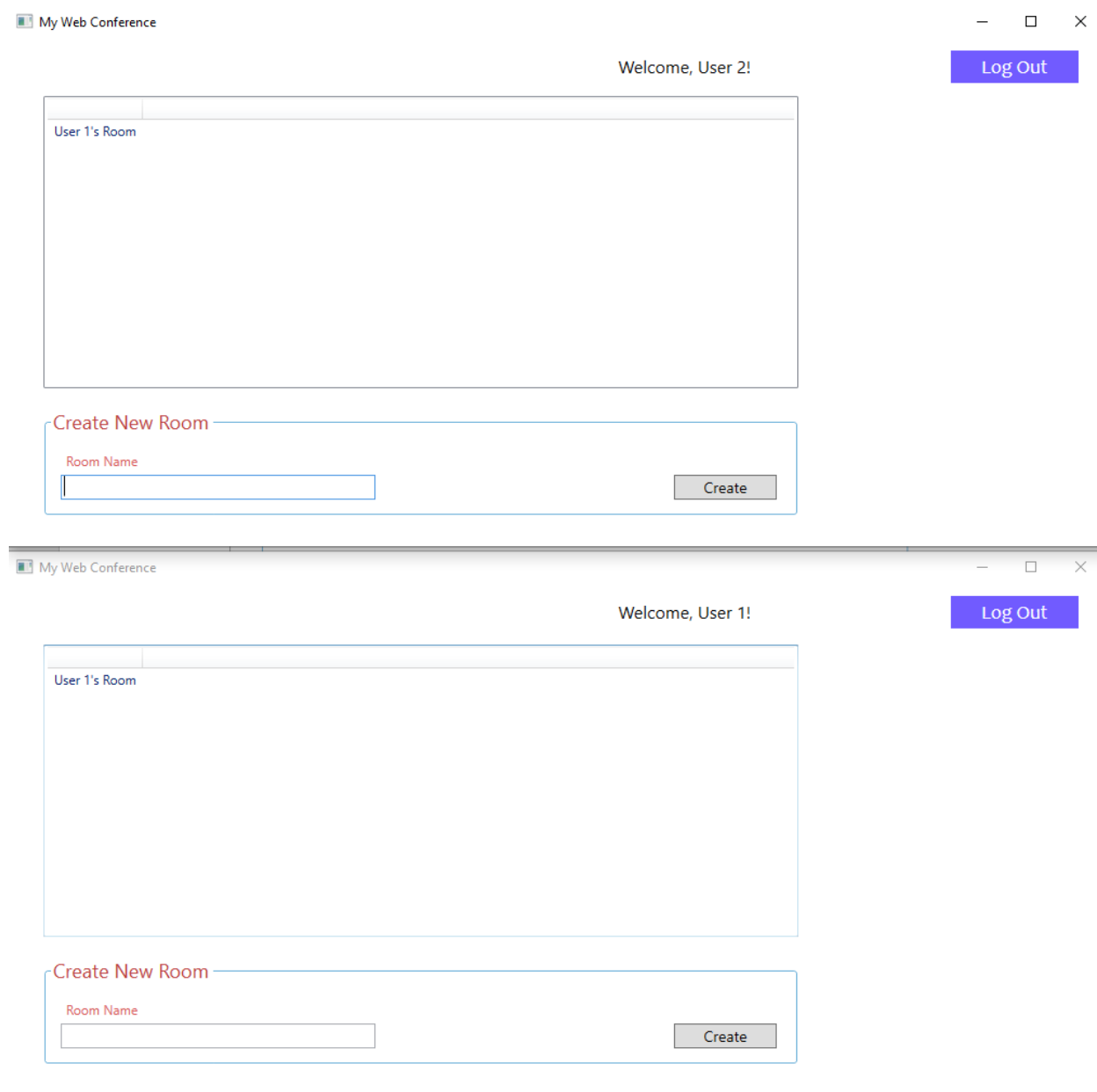


Рисунок 6 Новий чат відображено для всіх приєднаних користувачів

Щоб приєднатись до кімнати, необхідно натиснути на бажану кімнату зі списку.

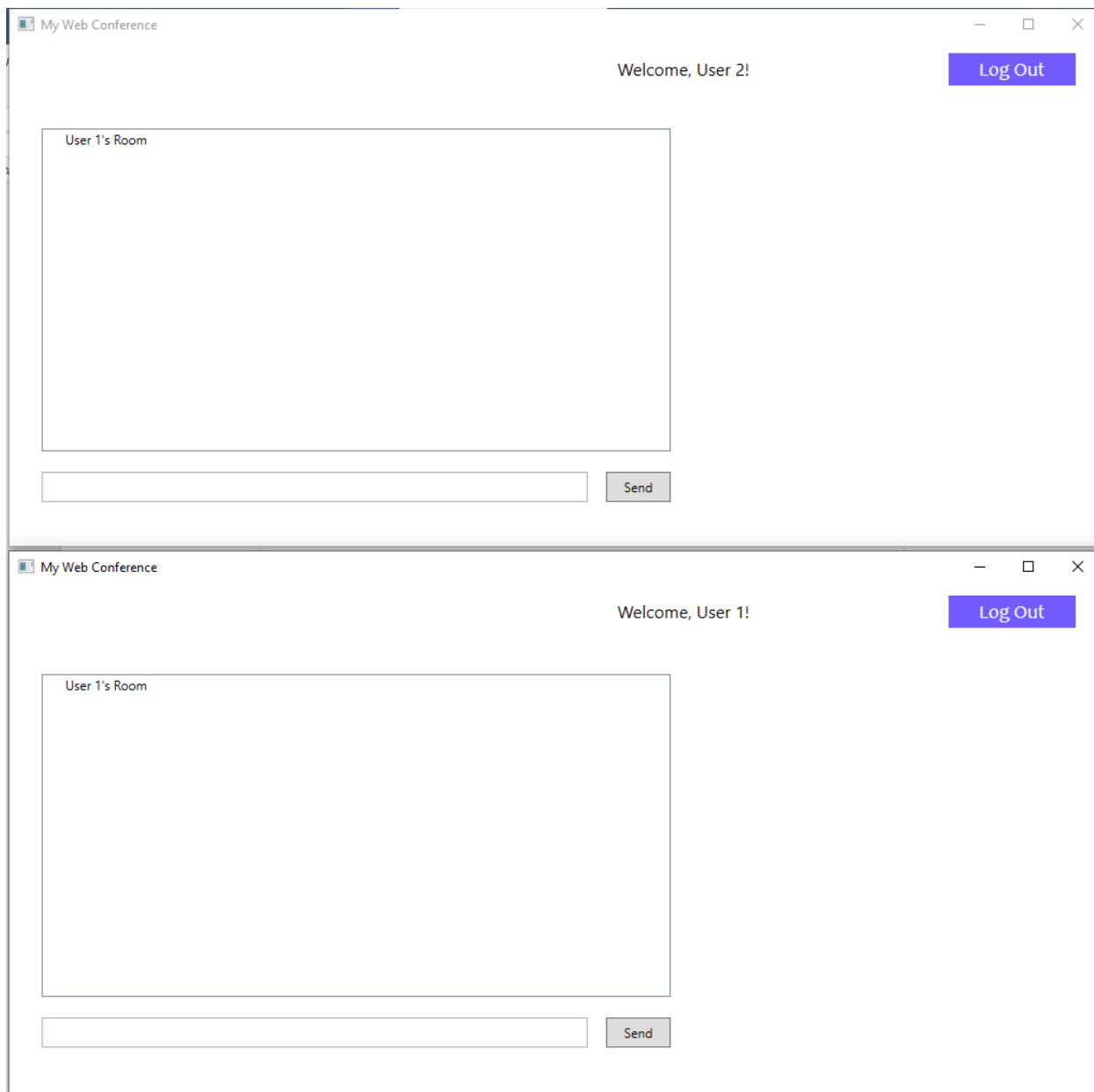


Рисунок 7 Користувачі увійшли до чату

Тепер можна написати повідомлення, яке побачать всі користувачі в цій кімнаті

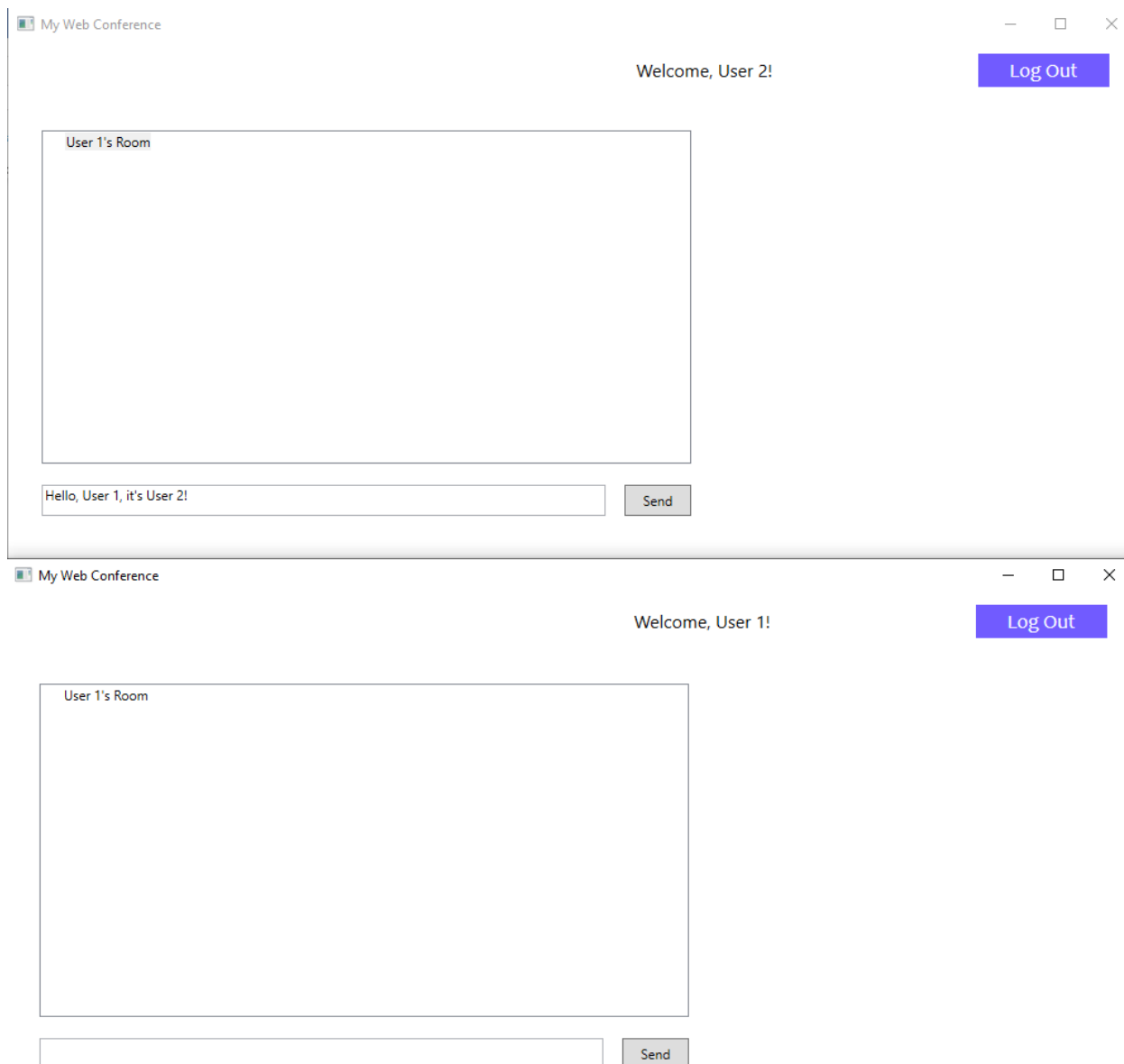


Рисунок 8 Набрано текст повідомлення

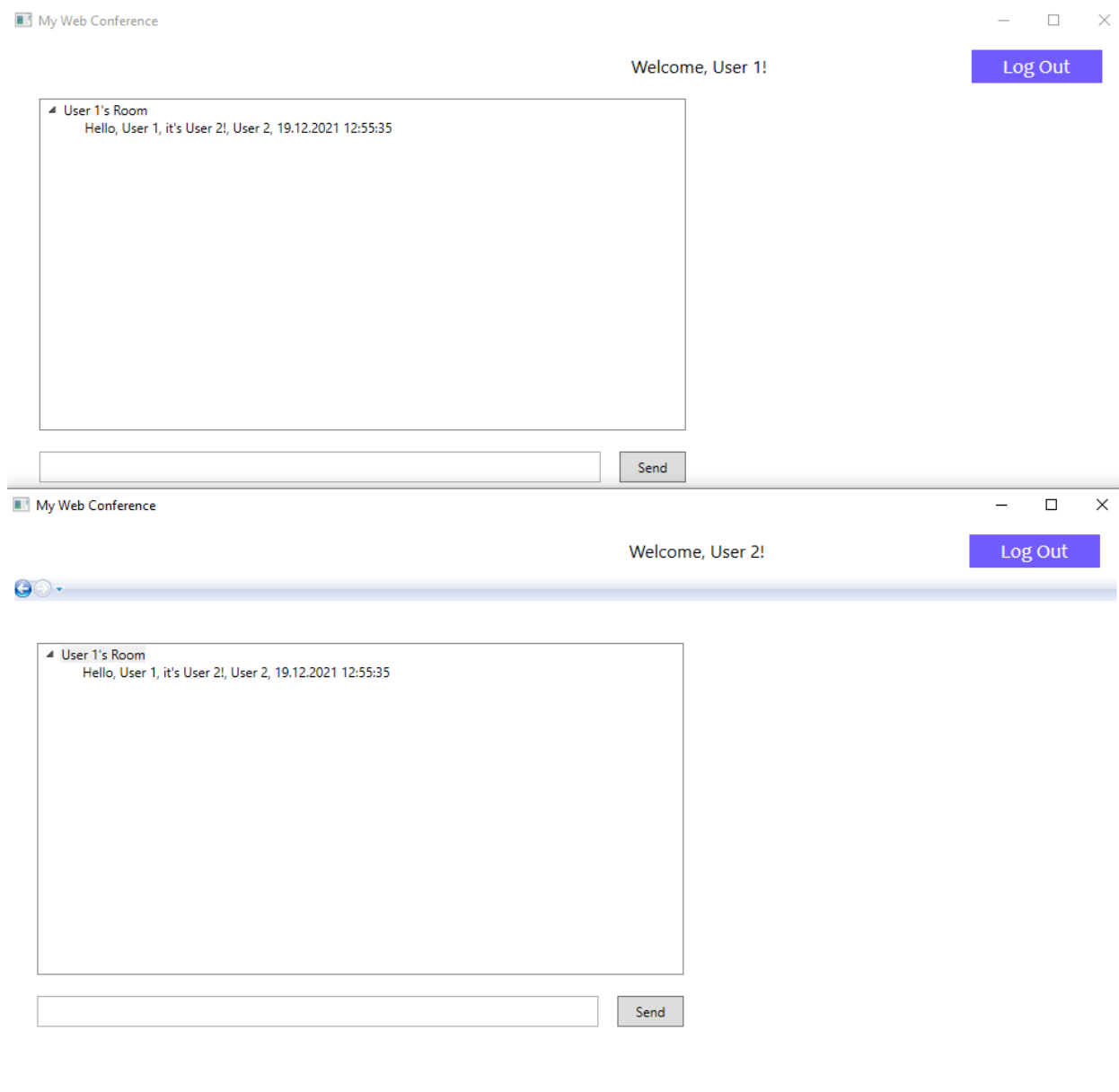


Рисунок 9 Повідомлення надіслане та отримане

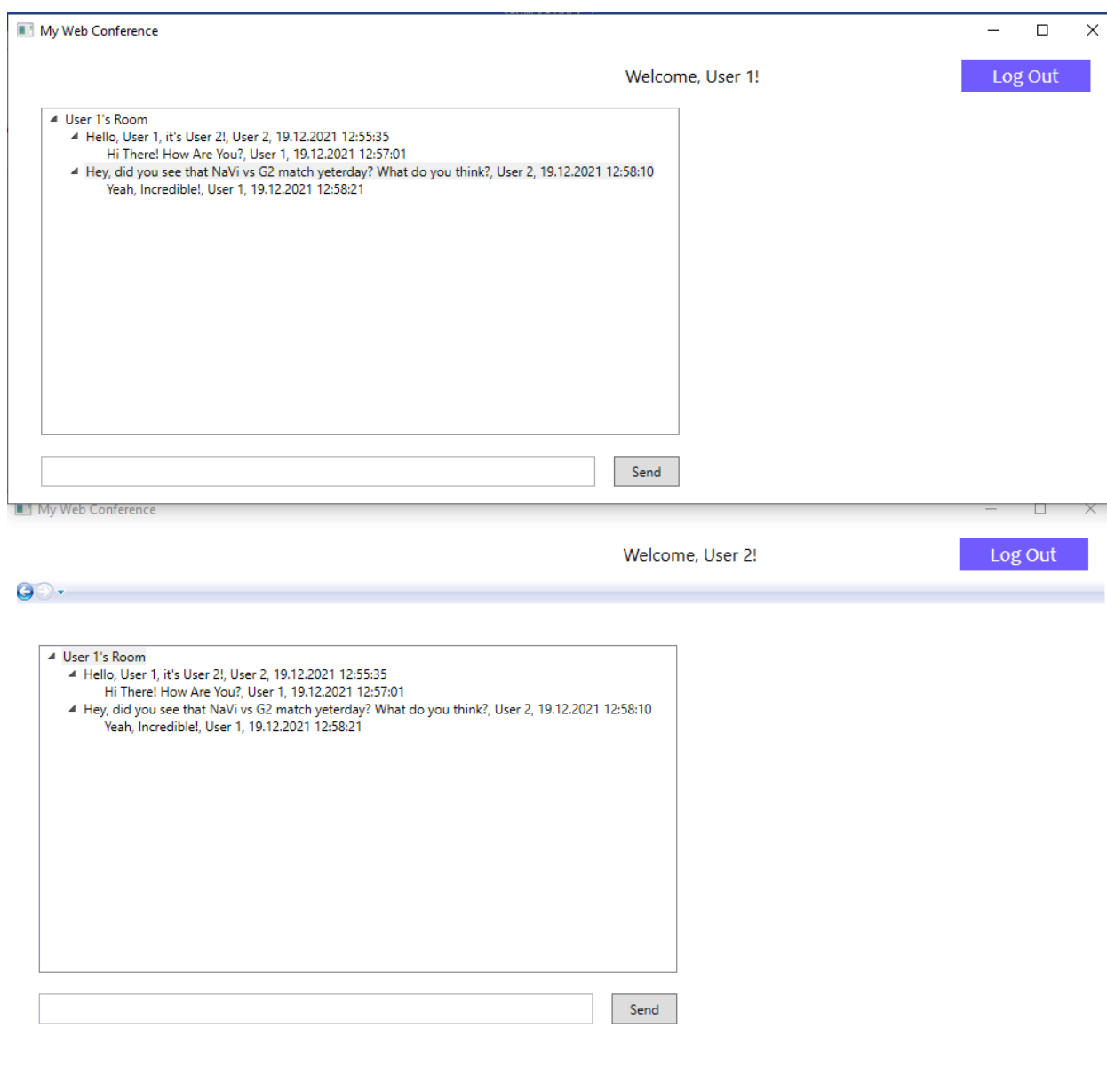


Рисунок 10 Дерево повідомлень

Висновки

В ході роботи було створено застосунок для обміну повідомленнями між користувачами по мережі на основі платформи WPF та мови програмування С#. Застосунок дозволяє надсилати та отримувати повідомлення у вигляді ієрархії повідомлень, що розділені на кімнати.

Перелік джерел посилання

- <https://www.geeksforgeeks.org/>
- <https://docs.microsoft.com/>
- <https://stackoverflow.com/>