UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

Applicazione Multi-chat

Giorgio Longobardo N86003571
Claudio Simonelli N86003781
Giuseppe Francione N86003734

Anno Accademico 2022/2023

Docente:
Alessandra Rossi

Indice

Capitolo 1

Introduzione

1.1 Descrizione e requisiti del progetto

Il progetto prevede la gestione e l'uso di una chat fra utenti multipli raggruppati in stanze. Gli utenti (i quali fungono da client) possono registrarsi o loggare mediante credenziali univoche. Inoltre, gli utenti possono creare stanze e possono chattare solamente con gli utenti presenti nella stanza. Gli utenti possono richiedere l'accesso a una stanza e l'utente creatore della stanza può gestire anche le richieste di accesso al gruppo, eliminando le richieste o aggiungendo gli utenti. Tutte queste operazioni devono essere effettuate interagendo con un server scritto in C.

1.2 Requisiti identificati

I requisiti identificati da soddisfare sono la gestione dell'interazione fra utenti, l'implementazione della comunicazione fra client-server e la gestione dei dati salvati del server.
Gli utenti sono implementati in un applicativo Android, il quale permette di svolegere tutte le funzioni sopracitate: al primo avvio viene richiesto di inserire le credenziali. Un utente può anche decidere di creare nuove credenziali e registrarsi. L'applicativo gestisce anche la creazione di nuovi gruppi, la visualizzazione dei gruppi a cui l'utente appartiene, la modifica delle credenziali stesse, visualizzazione dei gruppi a cui l'utente può inviare la richiesta di accesso, accettare o rifiutare richieste, inviare messaggi nelle stanze a cui appartiene e visualizzarne i messaggi.
Il server gestisce i dati tramite l'implementazione di un Database, ovvero Postgres. Ad ogni richiesta, il server effettua una query sul database e restituisce i risultati al client chiamante. Il server sfrutta la libreria libpq-dev e le funzioni di SELECT, INSERT, DELETE, UPDATE sono state implementate ad hoc per la gestione dei dati.
La comunicazione avviene tramite socket TCP bloccanti. L'applicativo Android inizializza la socket e tenta di connettersi al server all'avvio e, in caso affermativo, stabilisce la connessione, permettendo di effettuare le operazioni. Se la connessione non può essere stabilita, allora qualsiasi altra operazione non andrà a buon fine, e quindi bisogna attendere per una connessione stabile. Il server tenta di accedere al Database Postgres all'avvio e, in caso di successo, è in ascolto per nuovi client.

1.3 Implementazione generale

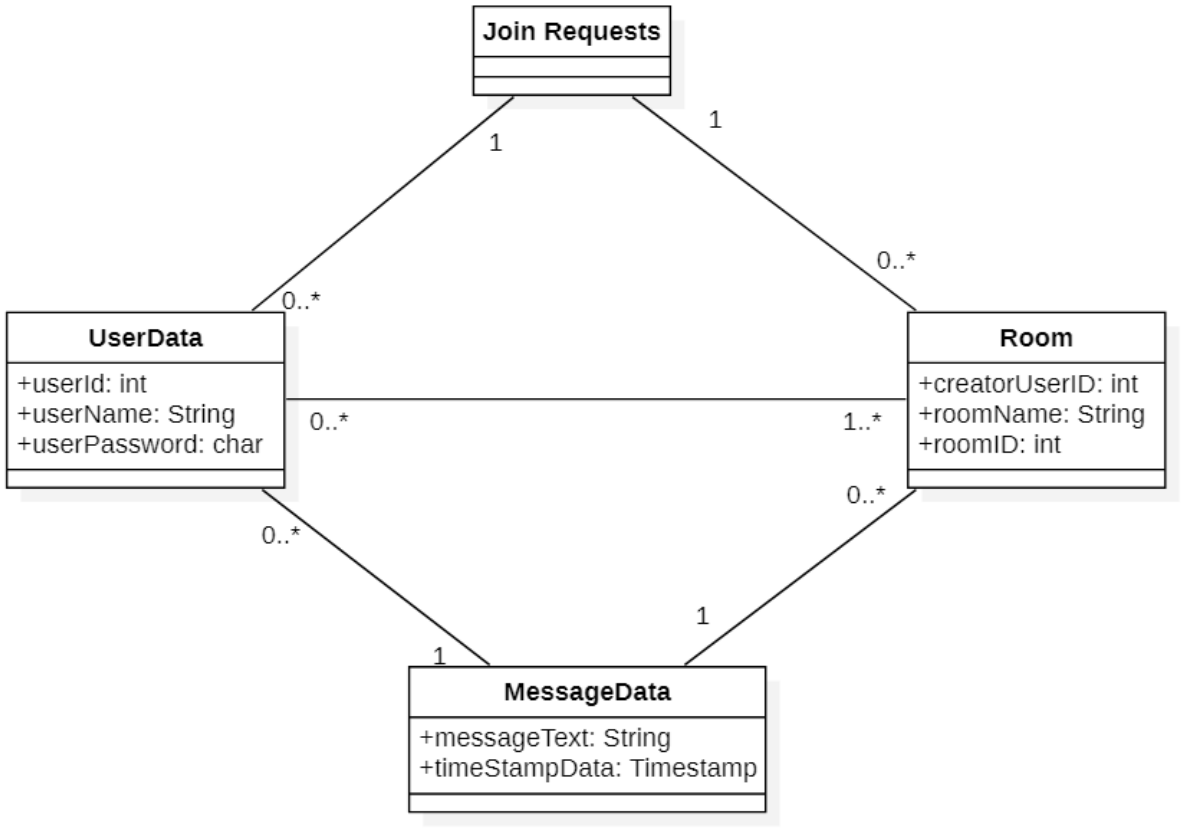
Per gestire gli utenti simultanei, il server crea un thread apposito per ogni client, ognuno di essi con una socket dedicata. Inoltre ogni socket viene inserito in un array globale in modo tale da tener traccia di tutti gli utenti connessi.
Gli utenti, effettuando un accesso, inviando al server le credenziali inserite e, in caso le credenziali siano corrette, allora si procede. Altrimenti, viene mostrato a schermo un messaggio di errore che invita l'utente a rivedere le credenziali inserite e di riprovare. Un utente creato equivale a inserire l'utente nel Database SQL. La creazione delle stanze funzion in modo analogo: al server vengono inviati nome e creatore, dopodiché il server verifica i dati ottenuti ed eventualmente fa la INSERT nel database.
Quando un utente apre un gruppo a cui appartiene, viene mostrata la chat di quel gruppo e, simultaneamente, comunica al server che é pronto per l'invio dei messaggi ed è in attesa per i messaggi di altri utenti. Il server inserisce la socket in un nuovo array e appena vengono inviati messaggi scrive su tutte le socket presente nell'array quel messaggio, così da poter visualizzare nella chat il nuovo messaggio.
Ogni operazione richiesta dal client ha un identificativo che poi viene smistato da un switch-case presente in un pthread.

1.4 Implentazione Android

L'applicativo Android consiste in 3 Activity, ovvero: LoginActivity, la quale è deputata nella gestione dell'inserimento delle credenziali, MainActivity, la quale è composta da tre fragment che permettono di visualizzare rispettivamente i gruppi di appartenenza, i gruppi a cui è possibile inviare una richiesta e le richieste in pendenza; Inoltre è presente un pulsante tramite il quale si può effettuare il logout, modificare le proprie credenziali o creare un nuovo gruppo. Infine ChatActivity, ovvero la visualizzazione dei messaggi inviati di una stanza e un apposito riquadro per l'inserimento dei nuovi messaggi. Inoltre è presente la classe Connessione.java, la quale gestisce ogni operazione di comunicazione: apre e chiude la socket, riceve e invia i dati dal server (operazioni effettuate tramite Thread). La classe è statica ed è accessibile a tutte le altre classi affinché si possano effettuare le operazioni di comunicazioni senza dover passare ogni volta la variabile.

1.5 Implementazione Database

Il Database è stato implementato tramite PostgreSQL. Il database è già popolato e contiene già stanze, utenti, richieste di accesso, messaggi. Il database è stato strutturato seguendo questo diagramma UML:



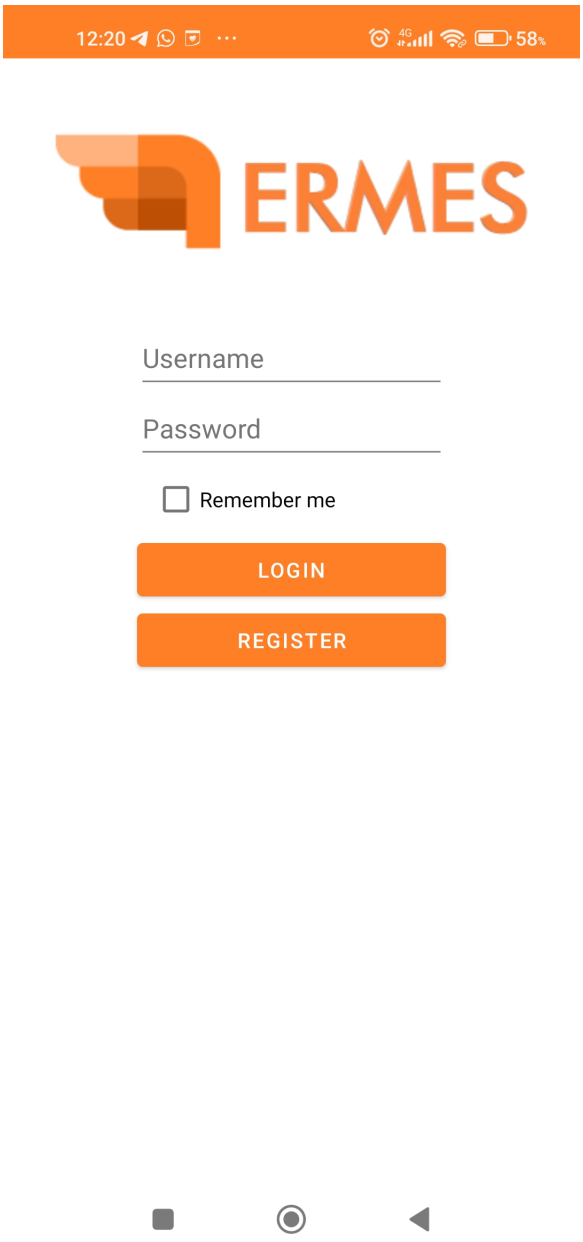
La tabella Userdata contiene tutte le informazioni necessarie degli utenti: username, password e un numero identificativo. La tabella Room contiene tutte le informazioni necessarie di una stanza: l'id del suo creatore, il nome della stanza, e un proprio identificativo. La tabella MessageData contiene tutte le informazioni necessarie dei messaggi, ovvero il testo del messaggio e il TimeStamp dell'invio. La tabella JoinRequest contiene le richieste in attesa.
Il DBMS contiene anche dei trigger con l'intento di mantenere l'integrità complessiva dei dati. Si possono consultare tutti i trigger implementati qui.

1.6 Test effettuati

I test sono stati effettuati su un pc con sistema operativo Linux (Ubuntu e Debian), e quatttro dispositivi android. Abbiamo connesso i dispositivi su una rete locale in modo tale da poterli connetterli fra loro. La connessione risulta stabile, mentre le operazioni I/O sulle socket risultano leggermente lente.

1.7 Screenshots

1.7.1 Schermata di accesso



1.7.2 Schermata visualizzazione propri gruppi

Vengono mostrati i propri gruppi, se cliccati viene mostrata la chat

1.7.3 Schermata visualizzazione gruppi estranei

Vengono mostrati i gruppi estranei e i loro ultimi messaggi, se premuti si invia una richiesta al gruppo

1.7.4 Schermata visualizzione Chat

La chat mostra l'username dell'utente, il temstamp e il messaggio stesso, oltre al nome del gruppo di appartenenza.

1.8 Conclusione

La versione di postgresql usata è la 14 e abbiamo usato github come software per il controllo di versione distribuito, la repository è consultabile qui

Giorgio Longobardo
Giuseppe Francione
Claudio Simonelli