



LCHAT

Loris Zannini N86003744
Mattia Marucci N86003853

Anno Accademico 2022/2023

Docente:
Alessandra Rossi

Indice

LhChat

- Descrizione e requisiti del progetto 3
- Requisiti identificati 3
- Guida al Client..... 3
- Guida al Server 4
- Test effettuati 5
- Guida all’uso 5
 - Finestra di accesso 5
 - Finestra home 6
 - Finestra profilo..... 7
 - Finestra ricerca stanza 8
 - Finestra chat 9
- Conclusioni 9

LhChat

Descrizione e requisiti del progetto

Il progetto prevede l'implementazione di un sistema Multi-Chat in cui gli utenti possono scambiare messaggi all'interno di una stanza con gli altri membri, creare nuove stanze e richiedere accesso ad una stanza già presente. Ogni utente per entrare nel sistema dovrà registrarsi tramite username (univoco) e password. Le richieste di accesso ad una stanza verranno inviate all'admin che potrà rifiutarle o accettarle. L'applicazione richiede un'implementazione mediante il paradigma client-server.

Requisiti identificati

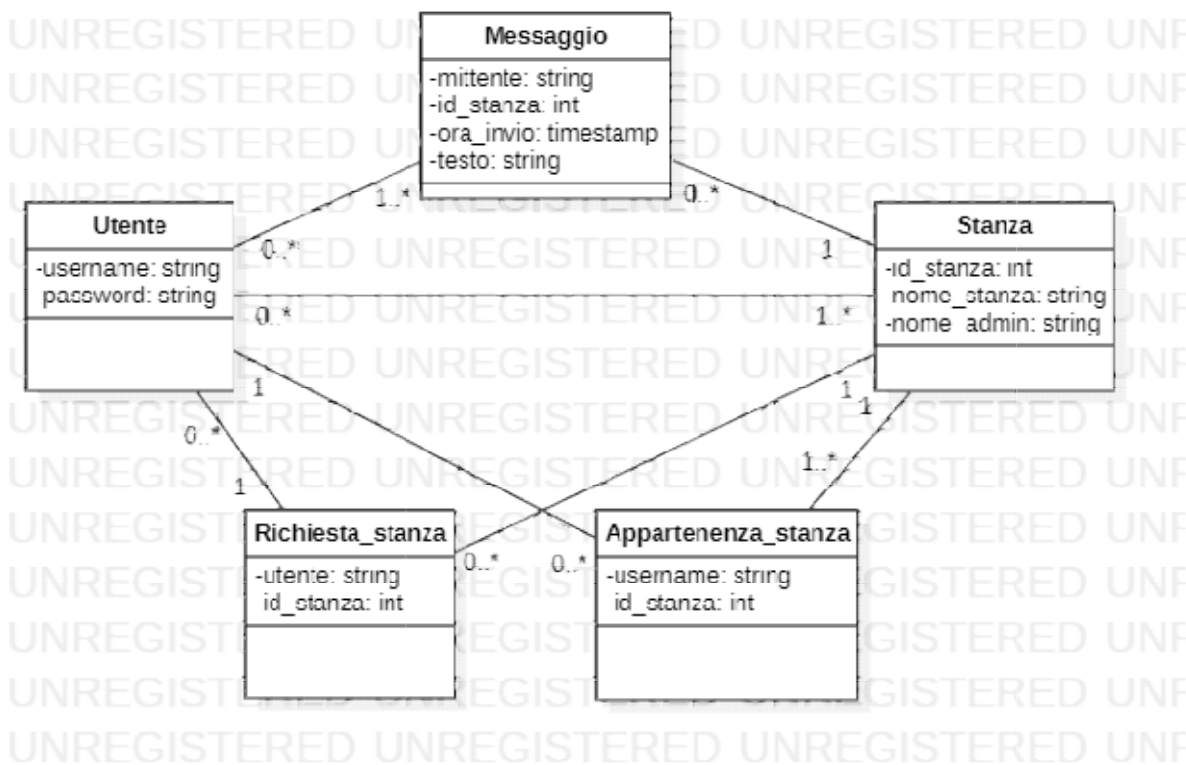
I requisiti identificati riguardano l'implementazione della comunicazione client-server, con l'utilizzo di un database per salvare i dati e la gestione del sistema informativo che riguarda l'interazione di n utenti che scambiano messaggi all'interno delle stanze. L'utente navigherà nel sistema tramite interfacce grafiche; all'apertura, si troverà nella schermata di avvio in cui verrà richiesto il login per entrare o, nel caso di primo avvio, potrà registrarsi in una apposita finestra. All'interno dell'applicazione sono presenti le seguenti funzionalità: creazione e ricerca di una stanza, richiesta di accesso ad una stanza, modifica della propria password, invio e visione dei messaggi di una stanza e visualizzazione dei partecipanti di ogni stanza. L'admin, in più, può gestire le richieste di accesso effettuate dagli utenti all'interno di una finestra accessibile solo ad egli. Il server, che resta in ascolto per i vari client, gestisce i dati del database mediante le richieste effettuate. In particolare, un client invia una query al server, che risponderà tramite le informazioni richieste in caso di successo o mediante messaggi di errore in caso di insuccesso. La comunicazione tra i due enti avviene tramite l'utilizzo di socket implementate in c.

Guida al Client

Il client Android è scritto in Java ed è costituito dalle seguenti Activity: LoginActivity, che rappresenta l'activity iniziale in cui l'utente può eseguire il login, RegistrazioneActivity, che rappresenta l'activity in cui l'utente può registrarsi nel sistema, HomeActivity, nella quale sono presenti le stanze di cui si fa parte ed un menu' per navigare nell'applicazione, ProfiloActivity, cioè la pagina profilo in cui è possibile visionare le proprie credenziali, effettuare il logout oppure cambiare password entrando in ModificaPasswordActivity, CreaStanzaActivity, in cui un utente può creare la propria stanza, CercaStanzaActivity, dove è possibile visualizzare tutte le stanze di cui non si fa parte e richiederne l'accesso oppure cercare una stanza digitando il nome, aprendo così StanzeTrovateActivity, ChatActivity, che rappresenta la schermata di chat di una determinata stanza, in cui è anche presente un menu' per abbandonare la stanza, visualizzare i partecipanti entrando in PartecipantiActivity, oppure vedere le richieste di accesso a quella stanza mediante VediRichiesteActivity. Quest'ultima Activity può essere acceduta soltanto dall'admin della stanza in questione. Oltre alle Activity il codice Java presenta anche 4 adapters ovvero: MessageAdapter, per permettere la visione dei messaggi in una stanza, PartecipantiAdapter, RichiesteAdapter e RoomsAdapter, che permettono di inglobare i rispettivi dati all'interno di collezioni che vengono visualizzate mediante l'uso di RecyclerViews. Abbiamo la cartella Model, in cui ci sono classi che rappresentano le entità all'interno del database, ovvero: Appartenenza_stanza, Messaggio, Richiesta_stanza, Stanza, Utente. Infine è presente la classe Controller che contiene tutti i metodi che permettono di eseguire richieste al Server per ogni singola funzionalità che richiediamo attraverso l'interfaccia grafica; in particolare ognuno di questi metodi crea un thread per mandare al server la richiesta effettuata mediante l'utilizzo di alcuni codici di 3 cifre che rappresentano le funzionalità. Ogni funzionalità avrà un codice di richiesta e due di risposta: uno di successo ed uno di errore.

Guida al Server

Il server, scritto in C, utilizza la libreria libpq-dev per gestire i dati presenti all'interno del database. I file c creati per l'implementazione del server sono: controller, che si occupa di gestire le richieste ricevute, database, in cui sono scritte le query per garantire tutte le funzionalità dell'applicativo, gestore_richieste, in cui vengono prodotte le risposte delle richieste effettuate dal client al server, e server, in cui viene instaurata la connessione tra end systems attraverso le socket (il server resterà in ascolto per accogliere nuove richieste). Viene creato un thread per ogni client in modo da gestire la simultaneità più utenti, a cui viene dedicata una socket per ognuno di essi. Il database è implementato in Postgres e contiene le tabelle utente, stanza, messaggio, appartenenza_stanza, richiesta_stanza. La sua struttura è raffigurata nel seguente diagramma UML:



La tabella Utente contiene i campi username (pk), che è univoco in tutto il sistema e serve per identificare ogni singolo utente, e password, che l'user digita in fase di registrazione/accesso. La tabella Stanza contiene i campi id_stanza (pk), che rappresenta l'identificativo della stanza, nome_stanza, che è univoco in tutto il sistema e nome_admin (fk), ovvero lo username dell'utente che ha creato quella stanza. La tabella messaggio contiene i campi mittente (fk), che rappresenta il mittente del messaggio, id_stanza (fk), che rappresenta l'identificativo della chat in cui si sta massaggiando, ora_invio, che rappresenta l'ora in cui è stato inviato il messaggio, e testo, ovvero il testo del messaggio.

All'interno del database sono presenti i seguenti trigger: `appartenenza_creatore()`, che inserisce il creatore della stanza all'interno di essa nel momento in cui viene creata e `delete_richiesta_on_insert()`, che inserisce in una determinata stanza l'utente che ha fatto richiesta e viene accettato dall'admin, togliendolo quindi dalla tabella richiesta_stanza.

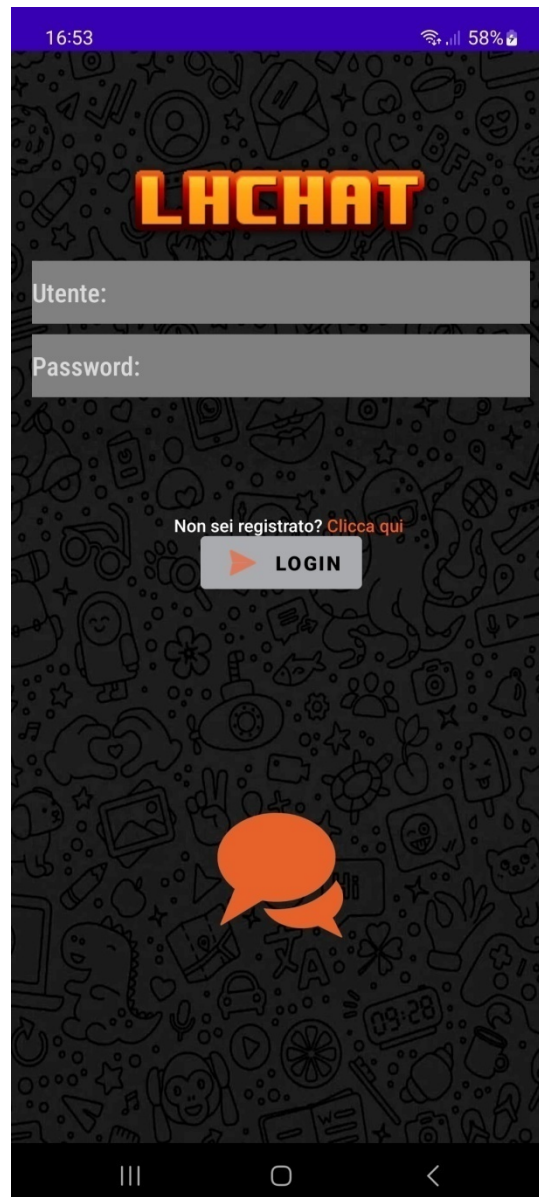
Come per il client, anche nel server sono presenti codici a tre cifre che rappresentano i comandi per l'esecuzione delle funzionalità. Nello specifico sarà il server ad occuparsi di decodificare i codici ricevuti e produrre una risposta, che sia positiva o negativa. Oltre ai codici di risposta con successo o con errore, sono presenti anche diversi codici "speciali", per gestire in modo specifico casi particolari, come ad esempio un utente che vuole registrarsi con uno username già presente nel sistema.

Test effettuati

I test sono stati effettuati mediante un dispositivo con sistema operativo Linux (Ubuntu), per quanto riguarda il server. Lato client abbiamo utilizzato 3 dispositivi android (di cui uno virtuale), per fare una prova multi client ed usare più di dispositivi simultaneamente. La connessione e le operazioni di comunicazione tramite socket sono risultate stabili.

Guida all'uso

Finestra di accesso



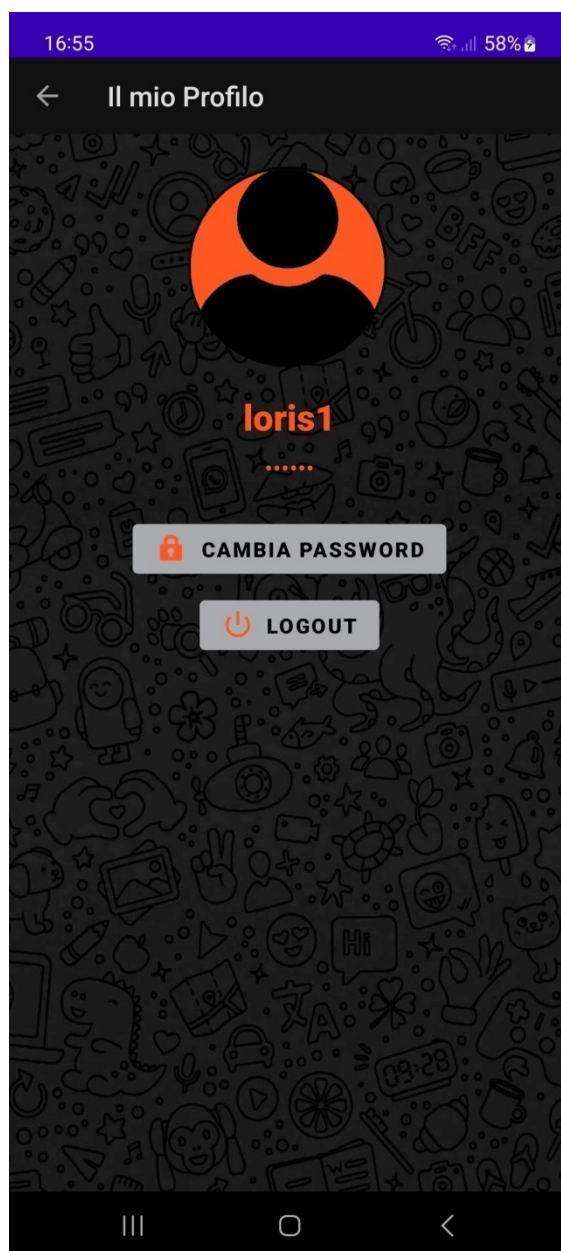
Nella schermata di accesso è possibile inserire le proprie credenziali per effettuare il login ed entrare nel sistema. Se un utente non è registrato potrà farlo cliccando sull'etichetta sopra il bottone, venendo indirizzato alla pagina di registrazione.

Finestra home



Nella home un utente potrà avere visione delle stanze in cui è presente, oppure navigare utilizzando il menù presente nella barra sopra. Grazie al menù l'utente potrà cercare una stanza, creare una stanza, disconnettersi o entrare nella pagina profilo.

Finestra profilo



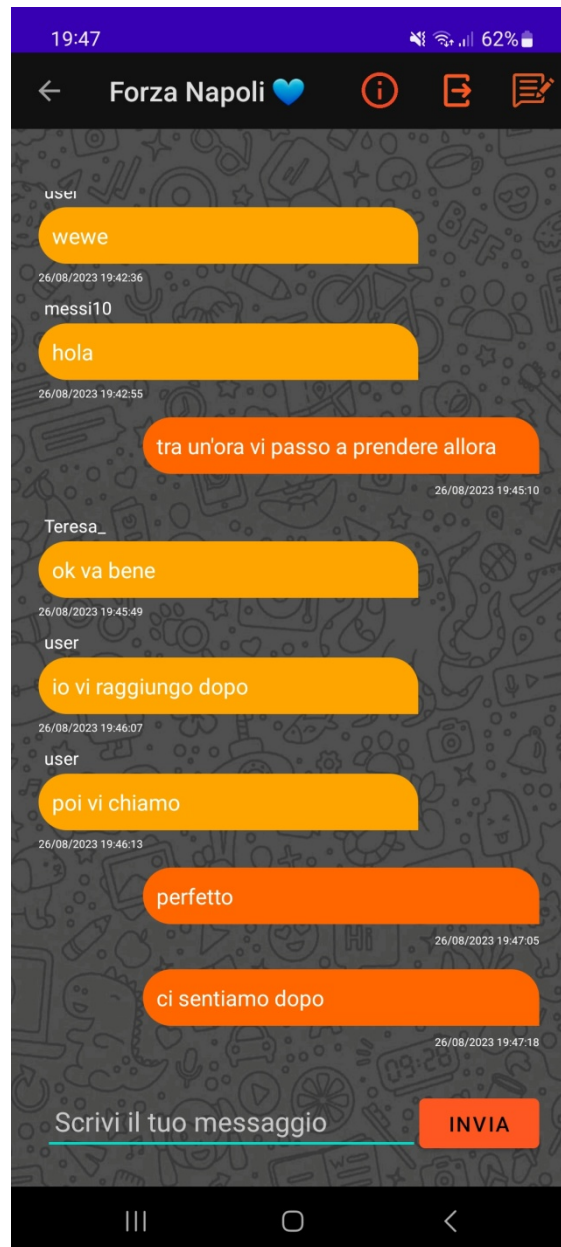
All'interno della schermata di profilo ogni utente potrà visualizzare le proprie credenziali (la password deve essere cliccata per visualizzarla), disconnettersi oppure cambiare la propria password.

Finestra ricerca stanza



In questa finestra l'utente potrà ricercare una stanza digitando il nome e cliccando il bottone cerca. Allo stesso tempo è presente una lista contenente tutte le stanze in cui l'utente non è presente, in caso voglia esplorarle una ad una.

Finestra chat



Nella schermata della chat i messaggi sono rappresentati mediante il testo e l'ora di invio (timestamp). Inoltre per i messaggi ricevuti abbiamo in alto il nome del mittente ed una colorazione più chiara per distinguere facilmente i messaggi inviati da quelli ricevuti.

Conclusioni

All'interno della directory principale è presente un file "README.md" nella quale sono elencate tutte le istruzioni per il funzionamento del Server. Il software utilizzato per la gestione delle versioni del sistema è GitHub, la repository è consultabile tramite questo [link](#). Grazie per l'attenzione.