Offline Messenger

Timofte Claudiu

1 Introducere

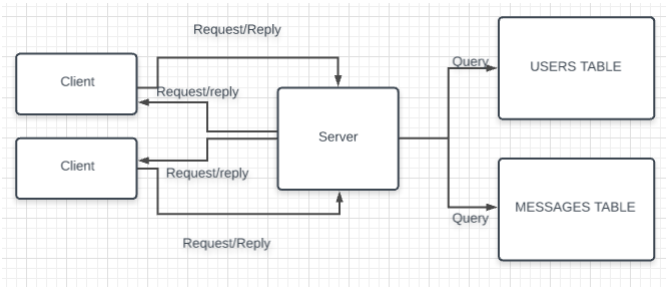
Offline Messenger este o aplicatie client/server permite schimbul de mesaje intre utilizatori care sunt conectati, pe baza unui username, si sa ofere functionalitatea trimiterii mesajelor si catre utilizatorii offline, dar inregistrati, acestora din urma aparandu-le mesajele atunci cand se vor conecta la server.Utilizatorii vor avea posibilitatea de a trimite un raspuns in mod specific la anumite mesaje primite, ei avand posibilitatea sa vizualizeze si istoricul conversatiilor pentru si cu fiecare utilizator in parte.

2 Tehnologii utilizate

Pentru stocarea utilizatorilor deja inregistrati, dar si pentru stocarea mesajelor am decis sa folosesc o baza de date de tip SQLITE3. Pentru comunicarea concurenta dintre server si clienti am ales sa folosesc protcolul TCP impotriva protocolului UDP pentru numeroase motive:
•.
•Ordine : Mesajele primite prin UDP nu respecta ordinea in care au fost trimise, acest lucru trebuind sa fie rezolvat mai tarziu in arhitectura aplicatiei.
•Duplicate : In cazul protocolului TCP duplicatele sunt eliminate.In cazul acestei aplicatii duplicatele ar incetini performanta, deoarece baza de date va fi nevoita sa stocheze mai multe date.
•Pierdere de date : Protocolul UDP nu are metoda de backup in cazul pachetelor pierdute.

3 Arhitectura programului

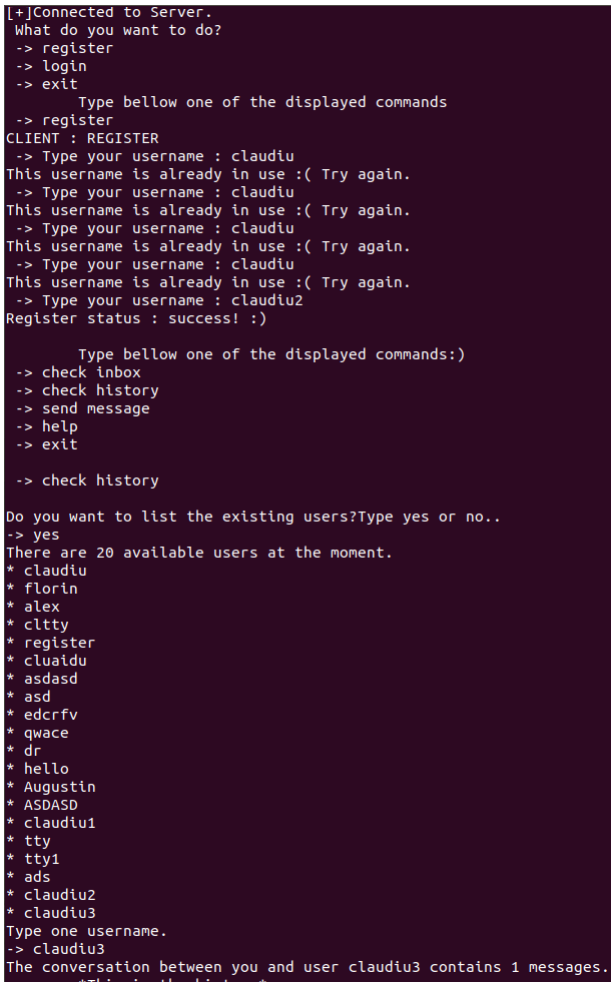
Aplicatia este de tip server-client concurent unde unul sau mai multi clienti se conecteaza la server cu scopul de a trimite anumite mesaje, pentru a verifica mesajele deja primite sau chiar istoricul conversatiilor.



3.0.1 Client Side

•Se conecteaza la server prin primitivele specifice.
•Initial are 3 comenzi la dispozitie : register, login si exit.

•Dupa ce se va loga, respectiv inregistra, utilizatorului ii va aparea un alt meniu compus din 5 comenzi:check inbox, check history, send message, help, exit.



•Clientul are posibilitatea de a introducere un numar nelimitat de comenzi pana la comanda *exit care il va deconecta de la server.

3.0.2 Server Side:

•Supravegheaza un port pentru a accepta o eventuala conexiune din partea unui client.
•Faptul ca este un server concurent poate accepta mai multi clienti in acelasi timp.
•Detine evidenta clientilor inregistrati, a mesajelor necitite si a istoricului conversatiei.
•Pentru obtinerea istoricului server-ul interogheaza baza de date intitulata CHAT.db compusa din doua tabele USERS si MESSAGES, care la randul lor contin campurile.

```
int callbackGetHistory(void *NotUsed, int argc, char **argv, char** azColName);
int getHistory(const char *msgID) {
    sqlite3* db;
    int exit = sqlite3_open("CHAT.db", &db);

    string sql = "SELECT SENDER_ID, RECEIVER_ID, TEXT, DATE FROM MESSAGES WHERE MESSAGE_ID = '";
    sql = sql + msgID;
    sql = sql + "' ORDER BY DATE ASC;";

    cout << "SQL COMMAND IS : " << sql << '\n';

    sqlite3_exec(db, sql.c_str(), callbackGetHistory, NULL, NULL);

    return 0;
}

int callbackGetHistory(void *NotUsed, int argc, char **argv, char** azColName) {
    cout << "INSIDE CALLBACk\n";
    senders[msgCounter] = (char *)malloc (100 * sizeof (char));
    receivers[msgCounter] = (char *)malloc (100 * sizeof (char));
    mssg[msgCounter] = (char *)malloc (100 * sizeof (char));
    date[msgCounter] = (char *)malloc (100 * sizeof (char));


    getUSERNAME(argv[0]);
    strcpy(senders[msgCounter], user_name);

    getUSERNAME(argv[1]);
    strcpy(receivers[msgCounter], user_name);

    strcpy(mssg[msgCounter], argv[2]);
    strcpy(date[msgCounter], argv[3]);

    cout << "SENDER : " ;
    printf("%s <-> %s\n", argv[0], senders[msgCounter]
    cout << "RECEIVER : ";
    printf("%s  <-> %s\n", argv[1], receivers[msgCount
    cout << "MSSG : ";
    printf("%s\n", mssg[msgCounter]);

    printf("DATE : %s\n", date[msgCounter]);

    msgCounter++;

    return 0;
}
```

4 Concluzie

Programul va fi o aplicatie client/server concurenta ce permite schimbul de mesaje in timp real intre utilizatorii conectati sau inregistrati in timp real.

5 Bibliografie

Principles of Concurrent and Distributed Programming by Addison-Wesley.
Click here
SQLite3 documentation.
Click here