Universitatea Alexandru Ioan Cuza

Facultatea de informatică

Tema: VirtualSoc

Scortanu Stefana-Raluca

Anul II, Grupa A7

Anul universitar 2016-2017

Cuprins

Enuntul temei

Sa se realizeze o aplicatie client/server ce va simula functionalitatea unei retele sociale. Sistemul ofera urmatoarele: inregistrarea utilizatorilor (diferite tipuri de utilizatori: obisnuiti, administratori). Utilizatorii isi pot seta profilul si postruile ca fiind publice sau private. Utilizatorii pot vizualiza postarile publice ale celorlalti utilizatori fara sa fie autentificati.Toate celelalte comenzi vor fi restrictionate de autentificare. Sistemul va suporta minim urmatoarele operatii: adaugarea unui prieten in lista de prieteni(specificarea tipului de prieten: apropiat, cunostiinta, etc.), postarea unei stiri (stirea va putea fi: trimisa unui grup special: prieteni apropiati, prieteni sau poate fi publica). Sistemul permite un mecanism de comunicare privata intre 2 sau mai multe persoane.

Introducere

Conceptele din spatele retelelor de social media nu sunt noi – inca de la inceputul existentei oamenilor, am fost in continua cautare de moduri in care ne putem apropria, conecta si promova unul pe altul – doar ca au ajuns la cu totul alt nivel in era digitala. Unde obisnuiam sa avem strangeri de mana, comunicare verbala si scrisori timbrate, relatiile de astazi sunt de multe ori incepute si dezvoltate. Siteurile de socializare sunt aplicatii care permit utilizatorilor sa se conecteze prin crearea de profile cu informatii personale si mai apoi sa interactioneze cu prieteni si nu numai.

Tehnologii utilizate

Întrucât serverul nostru are drept scop furnizarea unei date unui anumit client care a cerut un anumit raspuns la comenzi (login, register, addfriend etc.), se va recurge la servicii orientate conexiune, adica la modelul TCP/IP (Transmission Control Protocol/Internet Protocol). TCP este un protocol orientat-conexiune, ceea ce înseamnă că o conexiune este realizată si menținută până când programele de aplicații implicate în procesul de comunicare termină schimbul de mesaje, fiind extrem de similar serviciului telefonic. TCP administrează mesajul sau fișierul ce se dorește a fi transmis, împărțindu-l în blocuri de dimensiuni mici (pachete), care apoi vor fi reasamblate în mesajul original, dorindu-și transmiterea de date fără eroare. IP tine evidența adresei spre care este îndreptat fiecare pachet astfel încât acesta să ajungă la destinația corectă. Chiar dacă fiecare pachet în care a fost împărțit un mesaj are aceeași adresă IP sursă și aceeași destinație, pachetele pot fi trimise pe mai multe căi de rutare. Fiecare pachet are un timp de viață, denumit TTL (Time-To-Live) care este setat de către destinatarul pachetului (de obicei are o valoare între 1 și 255) și care este decrementat la fiecare trecere printr-un router. Dacă acest timp expiră înainte ca pachetul să ajungă la destinație (devine zero), router-ul la care a ajuns în acel moment pachetul trimite serverului-destinatar un mesaj de tip Time Exceeded, utilizând protocolul ICMP. Astfel, scopul server-ului este de a mări treptat acest TTL, astfel încât să obțină mesaje de tip Time Exceeded de la

toate router-ele aflate pe traseu. Programul TCP de la nivelul clientului așteaptă să primească toate pachetele, apoi înștințează serverul de primirea lor (sau cere retransmiterea unui pachet neprimit), după care le asamblează în fișierul care se dorea primit. Modelul de comunicare TCP/IP este cel în care un client cere și primește date sau un serviciu de la un server.

Arhitectura aplicatiei

Arhitectura aplicației va fi una de tip Client/Server concurent TCP, iar serverul va comunica cu mai mulți client în același timp pentru a facilita accesul si viteza. La acceptarea conexiunii cu un nou client, în cadrul socketului se va crea un thread care să deservească acest client. În cadrul thread-ului creat serverul îi va da clientului posibilitatea de a se loga (client existent) pe baza unui username (unic) și a unei parole, stocate în baza de date a server-ului, sau de a se înregistra (client nou). Inregistrarea clienților se face într-o bază de date împreună cu profilele, prietenii si conversatiile lor. Dacă logarea clientului eșuează (username/parolă inexistentă), server-ul îi oferă posibilitatea de a mai încerca o data logarea (în caz că acesta a greșit la tastare) sau de a se inregistra ca si client nou. Înregistrarea unui nou client se face la fel, pe baza unui username, a unei parole, dar va cuprinde și introducerea datelor personale (nume, prenume, tipul profilului). Dupa inregistrare clientul poate sa se logheze, dupa care acestuia i se ofera mai multe posibilitati:

- poate sa vizualizeze newsfeed-ul cu postarile prietenilor (exista si un newsfeed public, ce poate fi vazut inainte de logare, unde sunt vizibile postarile tuturor clientilor);
- poate sa isi editeze profilul;
- poate sa isi adauge noi prieteni;
- poate sa comunice cu oricare prieten din lista de prieteni pe care ii are existenti in baza de date sau cu orice chat format din mai multi prieteni.

Ieșirea din sesiunea de lucru se va realiza la cererea clientului. După aceasta, thread-ul care îl deservea va lua sfârșit.

Întrucât baza de date a server-ului va fi destul de complexă, cu ajutorul SQLite, implementam o bază de date care va conține tabele pentru:

- users – tabela va primi ca argumente ID-ul, numele, prenumele, username-ul, parola, tipul (user sau admin – poate sa stearga postari publice), profilul (public sau privat – nu poate fi adaugat ca prieten de alti useri);
- friends – UserID, FriendID, Type (normal sau close);
- chats – ChatID, UserID;
- messages – ID, chatID, UserID, Date, Message;
- posts – PostID, UserID, Date, Type (publica sau privata – care poata fi vazuta doar de anumiti useri), Message.

Detalii de implementare

Implementarea serverului la nivel de acceptare a cererilor de conexiune va fi apropiata de cea clasica a unui server TCP concurent. Odata ce se conecteaza un user, se creeaza un thread care va deservi user-ul dat pana la deconectarea lui. La conectare, user-ului i se ofera sa se logheze sau sa se inregistreze, fara aceasta fiind imposibil de a face orice actiune (cu exceptia vizionarii postarilor din newsfeed-ul public). La orice mesaj trimis thread-ul clientului face o interogare la baza de date care va putea fi utilizata datorita librariei <sqlite3.h>.

De menționat că fiecare funcție a serverului (login(), register(), addfriend() etc.) va lucra cu baze de date. La astfel de funcții se vor face interogari în tabelele aferente, pentru a verifica dacă în tabel se află un client cu username-ul și parola introduse de client-ul actual.

Concluzii

In afara de cerintele din enunt, imbunatatirea aplicatiei se poate face prin crearea unei interfete grafice atractiva si prin implementarea unui socket pentru chat-urile din client, care sa aiba un sistem de management in server.

Bibliografie

- http://profs.info.uaic.ro/~adria/teach/courses/net/cursullaboratorul.php
- http://en.wikipedia.org/wiki/Transmission_Control_Protocol;
- http://stackoverflow.com/
- http://searchnetworking.techtarget.com/
- Sabin Buraga, Gabriel Ciobanu - Atelier de programare în rețelele de calculatoare, Ed. Polirom, 2001