Top Music

Manolache Ecaterina Vasilica

Facultatea de Informatică, Universitatea „Alexandru Ioan Cuza”, Iaşi

[ecaterina.manolache@info.uaic.ro](mailto:ecaterina.manolache@info.uaic.ro)

**Abstract.** Lucrarea prezinta problema pentru managementul unui top muzical ce contine mai multe melodii cu diferite caracteristici.In aceasta documentatie se prezinta o solutie pentru a utiliza aplicatia Top Music.

1. Introducere

Este dezvoltata o aplicatie client/server in cadrul in care clientii sunt utilizatorii care dupa ce sunt inregistrati apoi logati au acces la anumite optiuni ale aplicatie. Asadar aplicati Top Music pune la dispozitie utilizatorilor de a avea acces la un clasament muzical.

1.1 Utilizatorii vor avea posibilitatea sa vada topul la un anumit moment, in functie de numarul de voturi sau selectia se face in functie de genurile muzica. Pe langa asta va avea posibilitatea de a insera chiar el o melodie cu anumite specificatii.De asemenea utilizatorii pot posta anumite comentarii la melodiile aflate deja in top.

1.2 Administratorii are posibilitatea de a vedea topul musica la un anumit moment,lista utilizatorilor inscrisi in aplicatie , poate sterge melodii din top,utilizatori si poate opri posibilitatea unui anumit utilizator de a vota.

1. Tehnologii utilizate
   1. TCP

Pentru a implementa acest proiect se utilizeaza protocolul TCP (Transmission Control Protocol) utilizat in special de aplicatiile care au nevoie de confirmarea datelor la primire.

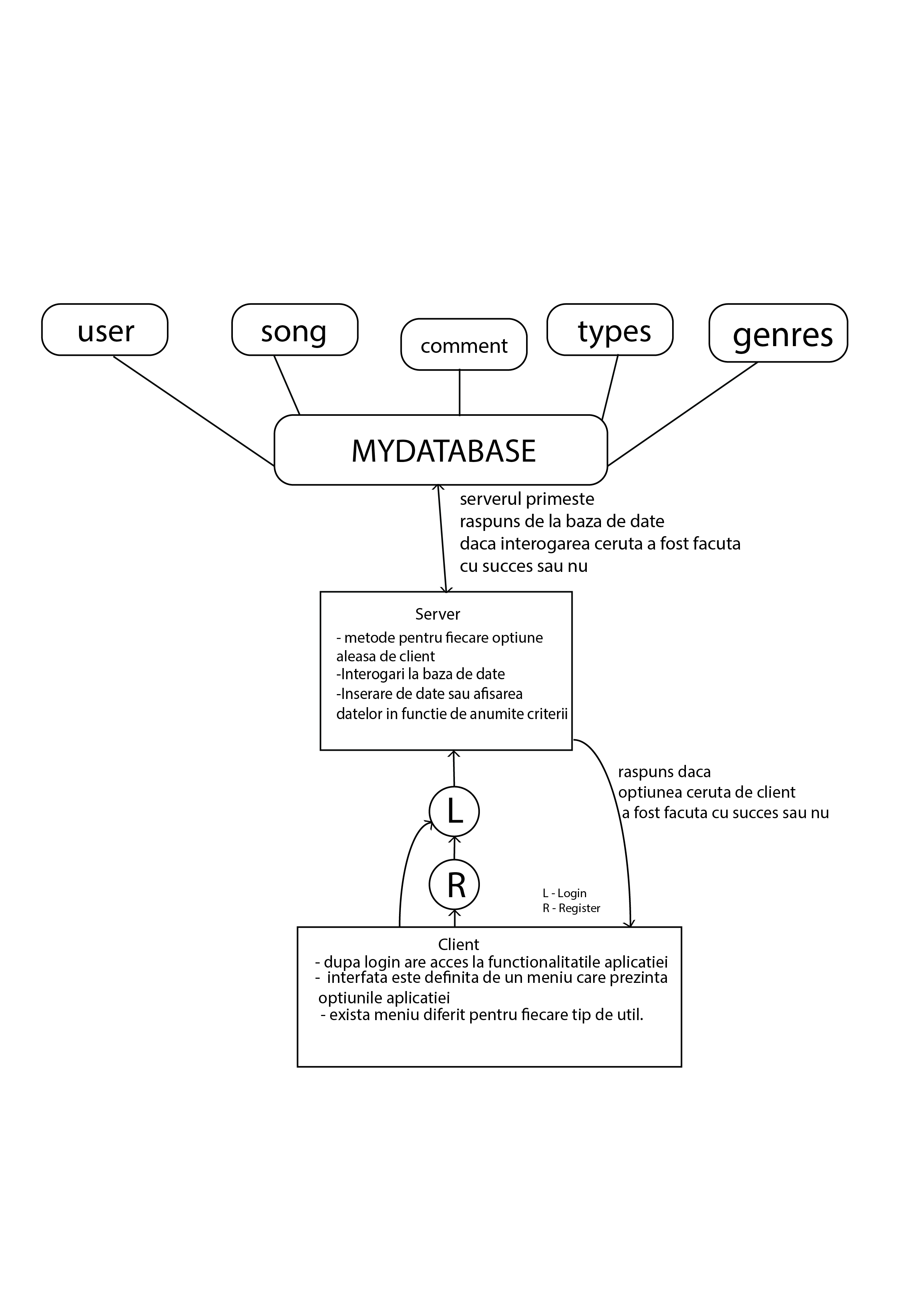
TCP imi garanteaza trimiterea si primirea datelor de la o gazda la alta fara duplicare sau pierdere de date. Receptorul raspunde intotdeauna cu un mesaj de conirmare daca primeste un anumit pachet de date. Expeditorul pastreaza pachetul pana cand vine confirmarea si daca aceasta e negative retrimite pachetul.

Aplicatia utilizeaza un model de TCP concurrent.

* 1. Sqlite3

Este utilizata biblioteca <sqlite3> pentru a introduce datele in baza de date myDataBase.db. SQLite este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat si cu ajutorul careia sunt salvata informatiile pentru aplicatia Top Music. In cadrul acestei aplicatii sunt dezvoltate 5 tabele care ajuta la functionarea aplicatiei.

1. Arhitectura
   1. Arhitectura client-server



Este folosita o arhitectura de tipul client/server TCP concurent. Serverul preia comenzile de conectare de la clienti si le ofera posibilitatea de inregistrare sau logare.

La inregistrare dupa ce clientul trimite username si parola, serverul verifica cazul in care exista vreun utilizator cu aceleasi date si in acest caz il anunta pe utilizatorul curent imediat oferindu-i posibilitatea de a se inregistra cu o noua

combinatie de username.

Dupa ce reuseste sa se inregistreze clientul se poate loga in aplicatie si imediat ii sunt puse la dispozitie comenzile de baza ale aplicatiei de management al melodiilor. Atat timp cat este logat si nu acceseaza optiunea de delogare utilizatorul poate apela optiunile aplicatiei.

3.3 Stocarea Datelor ( arhitectura bazei de date)

Stocarea datelor este facuta intr-o baza de date dataBase.db ce contine 5 tabele.

**User** - Aici sunt salvati toti utilizatorii aplicatiei ( usernameuri, parole si un flag care arata daca utilizatorul are drept de vot sau nu)

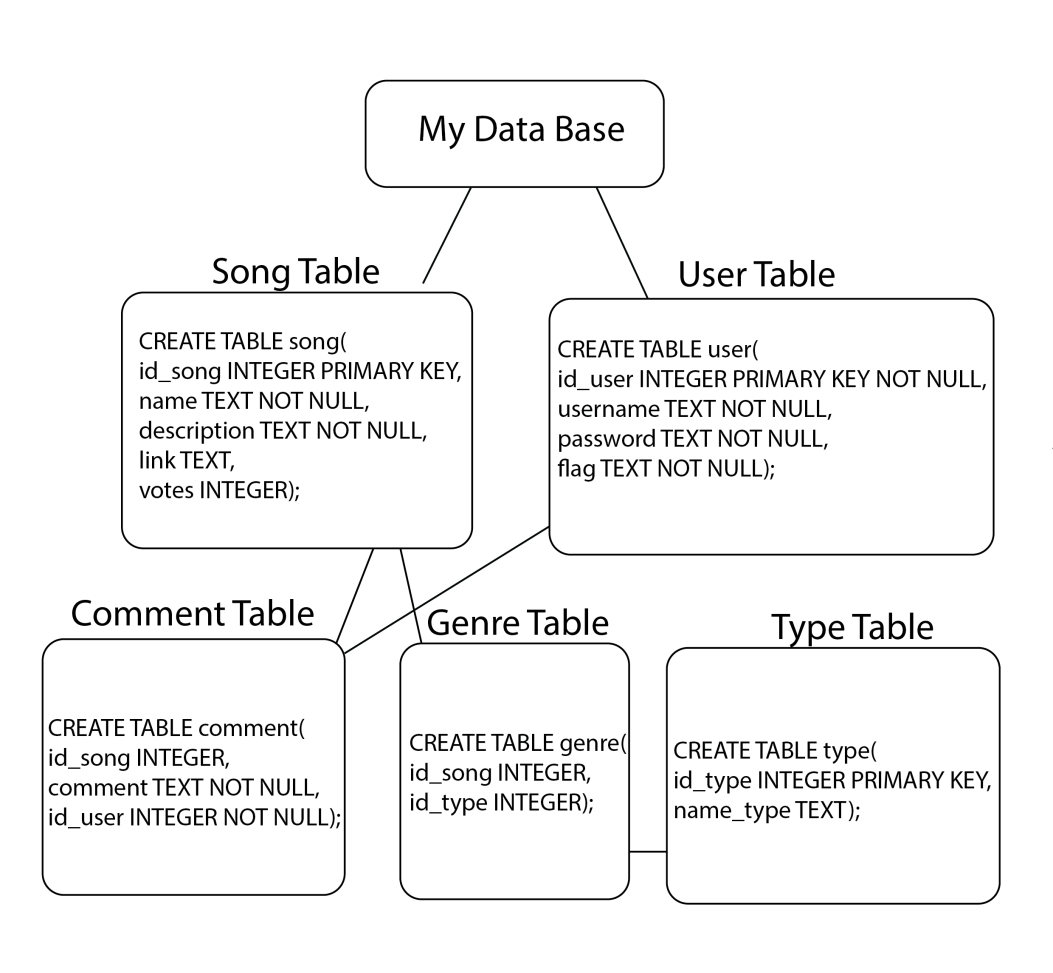
**Song** - Lista de melodii pe baza careia aplicatia functioneaza

**Types** – O lista unde sunt atribuite pentru diferite tipuri muzica un ID unic pentru a putea determina cu usurinta tipul melodiei in cazul in care utilizatorul doreste acest lucru

**Genre** –aici este creata legatura intre ID melodiei si ID tip(o melodie poate avea mai multe genuri muzicale)

**Comment**

-Introducem in tabela pentru un anumit id\_song un comentariu sau mai multe



1. Detalii de implementare

Aplicatia utilizeaza structura clasica de client-serverul. Dupa ce este pornit, serverul intra in modul de asteptarea pana la conectarea unui client. Pe de cealalta parte, clientul este cel care face cererea sa se conecteze la server si imediat dupa conecatare ii sunt puse la dispozitiile functionalitatile aplicatiei. Comunicarea aceaste intre client si server se face prin socketi ( apelul socket() ). Atunci cand clientul nu mai doreste comunicarea cu serverul, utilizatorul paraseste aplicatia iar la randul sau serverul inchide socketul acelui client acesta intrand din nou in faza de asteptare pentru alti clienti.

Exemplu cod server :

struct sockaddr\_in server;

struct sockaddr\_in client;

int socketDesc;

socketDesc = socket(AF\_INET,SOCK\_STREAM,0);

if( socketDesc == -1)

{

perror("Unable to make soket");

return -1;

}

bzero(&server, sizeof(server));

bzero(&client, sizeof(client));

server.sin\_family = AF\_INET;

server.sin\_addr.s\_addr = htonl(INADDR\_ANY);

server.sin\_port = htons(PORT);

if( bind(socketDesc,(struct sockaddr\*) &server,

sizeof(struct sockaddr)) == -1)

{

perror("Error at bind");

return -1;

}

if(listen( socketDesc,4) == -1)

{

perror("Error at listen");

return -1;

}

**Pentru concurenta este utilizat conceptul de thread.**

typedef struct thData

{

int id;//id thread<=>id\_client

int desc; //descriptorul

}thData;

… … … …

while(1)

{

int cliDesc;

thData\* thread;

socklen\_t length = sizeof(client);

printf("Server at port %d\n", PORT);

if( (cliDesc = accept(socketDesc, (struct sockaddr\*)&client,

&length)) <0)

{

perror("Error at accept() this client..");

continue;

}

int id;

int desc;

thread = (struct thData\*) malloc(sizeof(struct thData));

thread->id = i++;

thread->desc = cliDesc;

th = (pthread\_t\*) realloc(th, i \* sizeof(pthread\_t));

pthread\_create(&th[i-1],NULL,&connection,thread);

}

Pentru fiecare client conectat se face un thread si cat timp acesta ramane logat are propriul lui thread unde poate face operatiunile pe care le doreste.

4 Meniul aplicatiei

* + 1. Meniul initial – meniul care apare prima data pentru orice utilizator

…..

int option;

printf("\n 1. Register");

printf("\n 2. Login");

printf("\n 3. Exit\n");

printf("%s\n", "Choose what you want to do : " );

scanf("%d",&option);

…..

* + 1. Utilizator normal

……..

int option;

printf("%s\n","1. Show Top" );

printf("%s\n","2. Show Top Order by Votes" );

printf("%s\n","3. Add Song" );

printf("%s\n","4. Vote Song" );

printf("%s\n","5. Comment a song" );

printf("%s\n","6. Show genre top" );

printf("%s\n","7. Exit" );

scanf("%d",&option);

……

* + 1. Admin

int option;

printf("%s\n","1. Show Top" );

printf("%s\n","2. Remove Song" );

printf("%s\n","3. Stop user" );

printf("%s\n","4. Show users" );

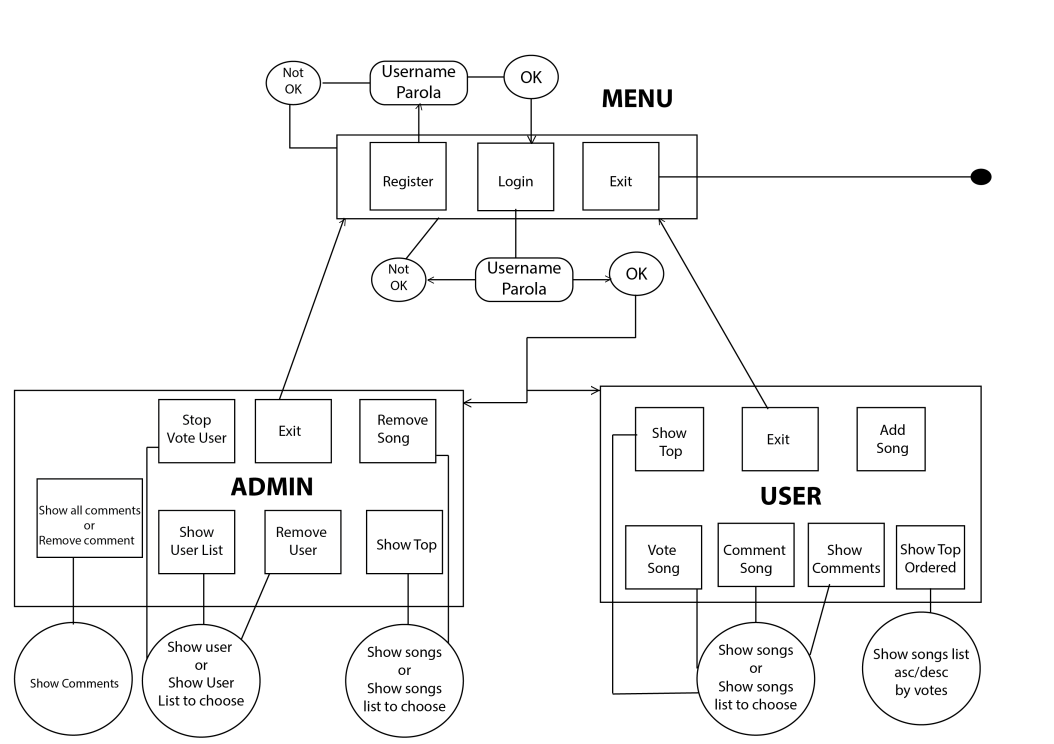
printf("%s\n","5. Exit" );

printf("%s\n", "Choose what you want to do : " );

scanf("%d",&option);

……

Schema Comunicare



1. Concluzii

O imbunatatire a aplicatiei poate fi facuta in primul rand printr-o arhitectura mai buna a bazei de date unde sunt pastrate toate informatiile aplicatiei( user, song,type,genre,comment).

In al doilea rand, o interfata grafica ar facilita interactiunea cu un posibil utilizator deoarece nu ar mai trebui facute cererile in format text.

1. Bibliografie

1. http://profs.info.uaic.ro/~adria/teach/courses/net/cursullaboratorul.php