

# Visit

## 1 Introducere

Proiectul Visit concepe o aplicație client/server care furnizează cărți de vizită în format VCARD utilizatorilor conectați. Utilizatorii sunt reprezentați de clienții aplicației. Un utilizator poate apela la următoarele servicii oferite de aplicație: căutarea unei cărți de vizită după numele și prenumele persoanei de pe ea, căutarea unei cărți de vizită după userul cărții de vizită, adăugarea unei noi cărți de vizită, ștergerea unei cărți de vizită.

## 2 Tehnologii utilizate

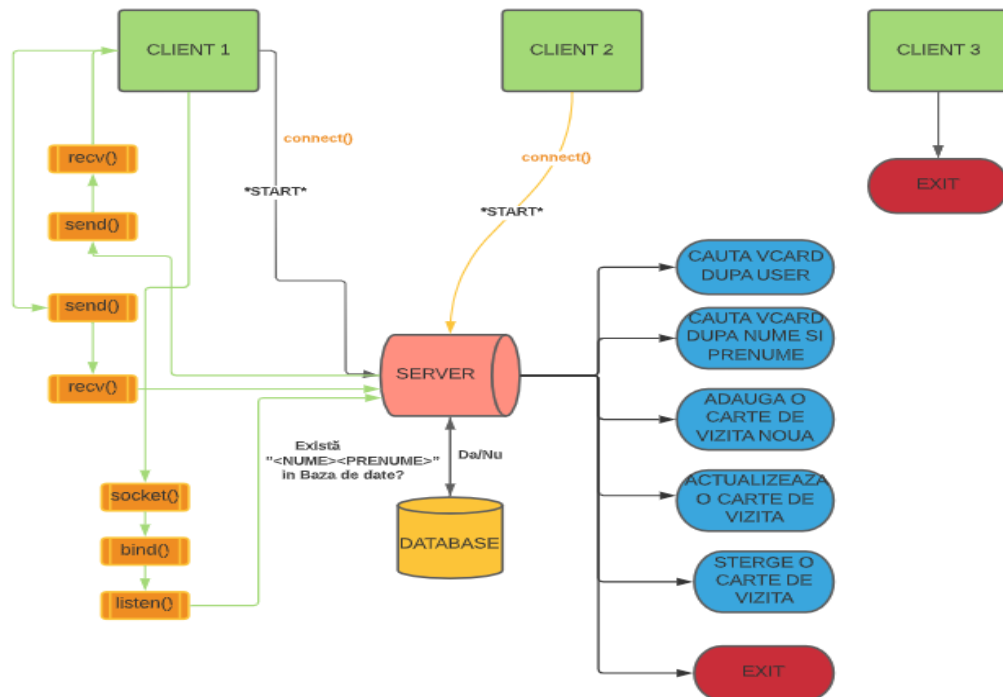
În primul rând am realizat necesitatea stocării datelor utilizate de aplicație într-o bază de date. Am ajuns la concluzia că este mult mai eficientă utilizarea unei baze de date SQL, creată cu ajutorul bibliotecii `sqlite3`. Această bază de date este gestionată mai ușor, mai sigur și mai eficient decât orice alt sistem de stocare a unor date, precum un fișier `.txt`. Prelucrarea datelor stocate folosind un fișier `.txt` este mai lentă comparativ cu prelucrarea datelor stocate dintr-o bază SQL, deoarece putem folosi mult mai rapid interogările predefinite în structura serverului aplicației.

În al doilea rând am ales să implementez un server TCP concurent deoarece acesta este mai sigur decât un server UDP, deoarece acesta transmite în ordine datele către clienți, evitându-se astfel situațiile în care datele sunt pierdute.

În ultimul rând, am folosit threaduri pentru servirea clienților aplicației, la intrarea fiecărui client în aplicație, se creează un thread care servește acest client până la deconectarea lui din aplicație.

## 3 Arhitectura aplicației

Conceptele implicate sunt în număr de 3: serverul, clientul și baza de date. Serverul poate accepta simultan (deoarece este un server TCP concurent) mai mulți clienți. Tot serverul comunică cu baza de date prin interogările deja concepute în structura sa pentru a verifica autenticitatea cererilor pe care le au clienții, și bineînțeles dacă cererile lor pot avea un răspuns.



## 4 Detalii de implementare

### 4.1 Scenarii de utilizare

Utilizatorii care aleg să folosească aplicația *VISIT* trebuie să se conecteze la server prin scrierea mesajului start. Din acest moment, un utilizator are mai multe comenzi pe care le poate apela. Odată intrați în "meniul" principal, utilizatorii pot folosi comenzile disponibile pentru ei:

- Comanda de a cauta o carte de vizită după user, le cere clienților să introducă userul cărții de vizită cerute, aceasta fiind afișată pe ecran.
- Comanda de a cauta o carte de vizită după numele și prenumele inscripționat pe ea, le cere clienților să introducă numele și prenumele cărții de vizită căutate, aceasta fiind afișată pe ecran.
- Comanda de a adăuga o carte de vizită nouă, le cere utilizatorilor să introducă datele necesare pentru o carte de vizită nouă.
- Comanda de ștergere a unei cărți de vizită, șterge o carte de vizită existentă după userul acesteia.
- Comanda de actualizare a unei cărți de vizită, le permite utilizatorilor să rescrie o carte de vizită.

În afară de aceste comenzi, mai există o ultimă comandă care poate fi folosită chiar de la deschiderea aplicației, comanda *exit*, care închide aplicația.

### 4.2 Cod relevant proiectului

Serverul folosește thread-uri deoarece acestea sunt mult mai eficiente decât utilizarea funcțiilor *fork()*. Fiecare thread servește câte un client în parte, economisind memorie. Acestea sunt implementate în server.

Serverul și clientul comunică prin intermediul unui *socket*. *Socketul* este unul dintre cele mai eficiente moduri de comunicare, deoarece aceasta realizează transmiterea datelor în mod bidirecțional, contrar pipe-urilor și fifo-urilor.

## 5 Concluzii

Aplicația implementată poate fi îmbunătățită din mai multe puncte de vedere, ca de exemplu:

-Din perspectiva securității, aplicația implementată nu oferă securizarea cărților de vizită create de utilizatori. Orice utilizator poate șterge informații din baza de date. Ca soluție pentru această problemă, s-ar putea cere autentificarea în niste conturi pentru clienți, și un cont pentru administrator, iar acesta din urmă poate accepta ștergerea sau modificarea datelor în baza de date.

-Din perspectiva utilizării, pentru a ușura folosirea aplicației de către utilizatori, clientului îi lipsește o interfață grafică.

## 6 Bibliografie

-Computer Networks - Faculty of Computer Science - Course and Laboratory: Computer Networks page;

-An Introduction To The SQLite C/C++ Interface: SQLite Home Page;

-Tutorialspoint.com:SQLite - Commands;

-Binarytides.com: Code Examples;

-App for drawing diagrams:LucidchartApp;

-Different questions:Stackoverflow;

-TCP Server-Client implementation in C:geeksforgeeks;

-VCARD FormatVCARD;