

Online Contacts

Iftimie Gabriel Codrut

Pester Leonard

Facultatea de Informatica Iasi

Rezumat

Sa se dezvolte o aplicatie Web care sa reprezinte un manager de contacte personale. Utilizatorii vor avea acces la functionalitati pe baza unui cont. Fiecare persoana de contact va avea asociate informatii precum nume, prenume, adresa, data nasterii, minim o fotografie, numere de telefon, adrese de e-mail, adrese Web, descriere, interese etc.

Aplicatia va permite gruparea contactelor, cautarea pe baza unor criterii complexe (de pilda, obtinerea persoanelor mai tinere de 20 de ani localizate in Iasi, actualmente la FII si care sunt interesate de tehnologii Web), precum si exportarea acestor informatii in diferite formate (cel putin vCard, CSV, Atom).

1. Tipuri de utilizatori

Utilizator neautentificat

Acest tip de utilizator are de ales între două optiuni, se poate autentifica dacă deține un cont sau în caz contrar se poate înregistra.

Utilizator autentificat

Acest tip de utilizator poate adăuga contacte în agenda personala, poate edita contactele adaugate cât și profilul acestuia, are posibilitatea de a forma grupuri din contactele adăugate.

2. Tehnologii utilizate

Apache 2.4.38

Se va folosi web serverul Apache deoarece este unul foarte puternic, ce oferă suport pentru limbajul de programare PHP și de asemenea permite modificarea web serverului după bunul plac.

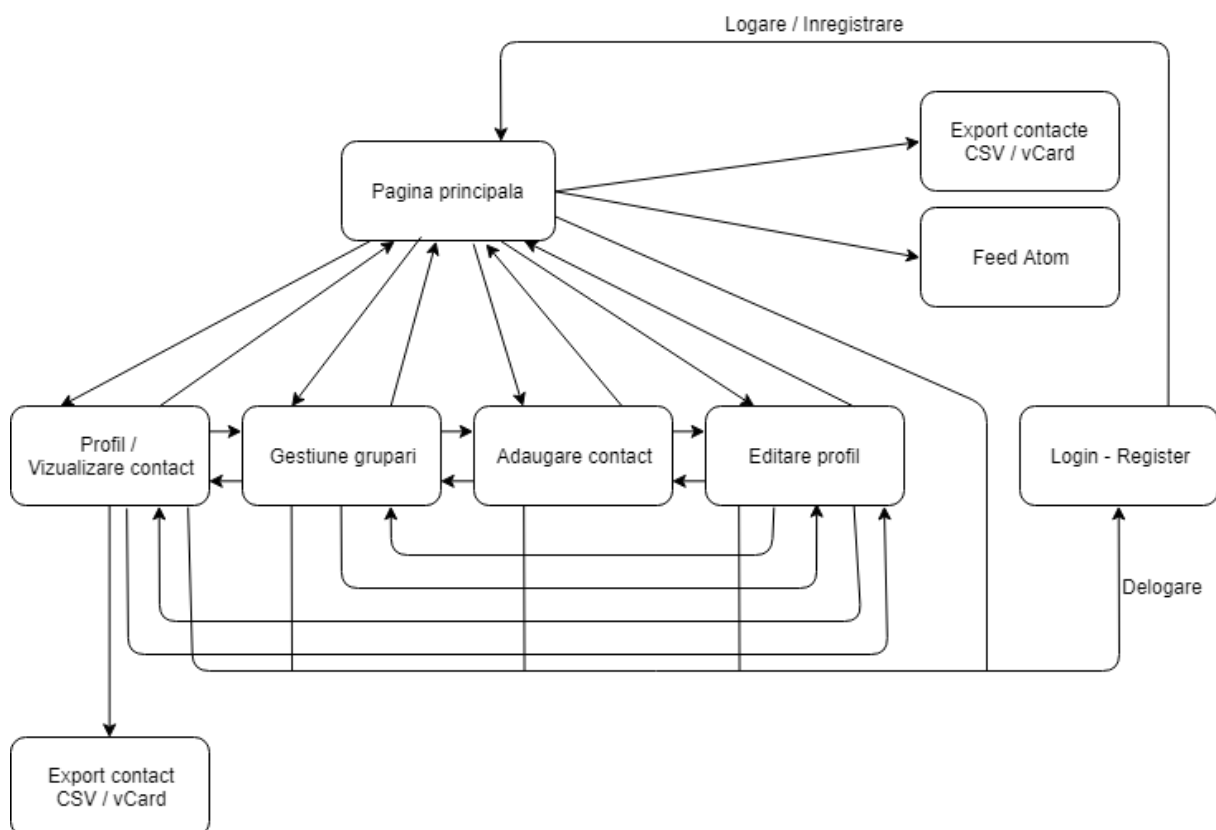
MySQL 5.7.21

Se va folosi managerul pentru baze de date relaționale MySQL deoarece oferă performanțe foarte ridicate chiar și când este foarte solicitat. De asemenea este ușor de gestionat atunci când este nevoie.

PHP 7.2.4

Ca și tehnologie backend se va folosi limbajul PHP deoarece este foarte ușor de integrat într-o pagină web HTML. Totodată este foarte ușor de folosit împreună cu MySQL, o altă tehnologie utilizată, iar acest limbaj este foarte bine documentat.

2. Arhitectura aplicației

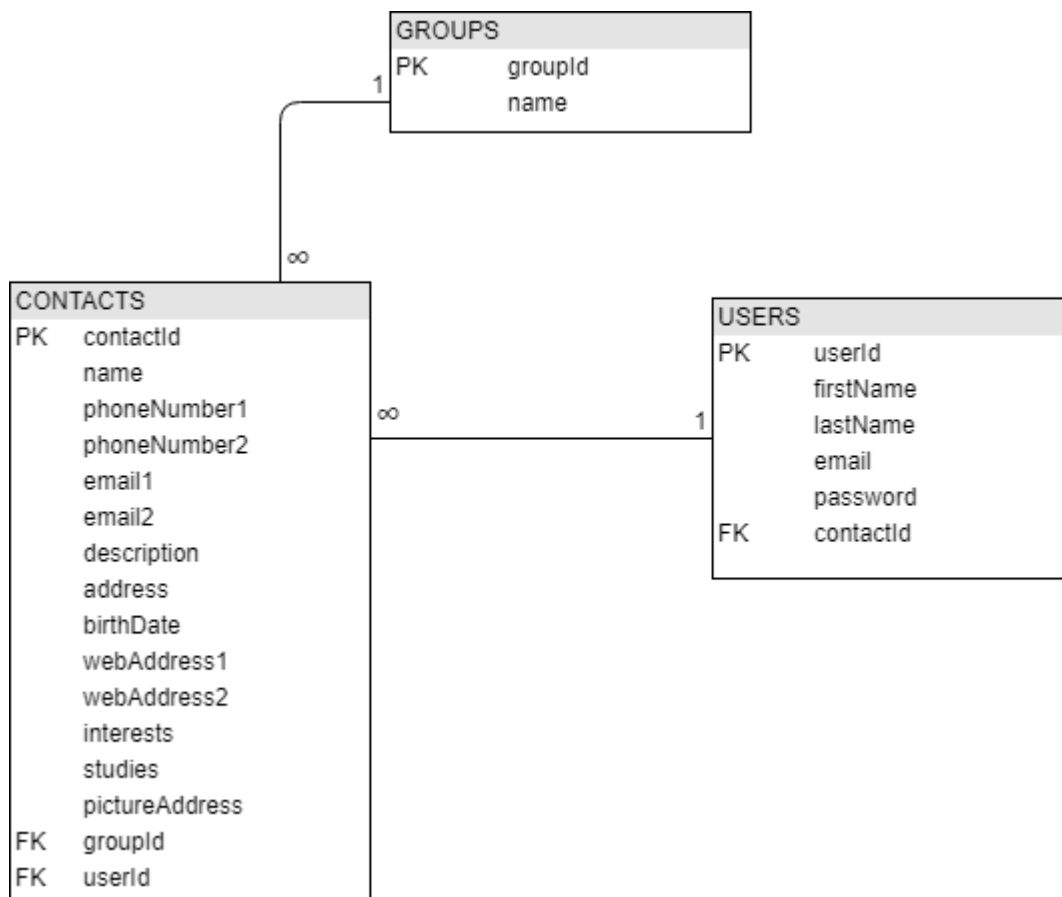


Aplicația, pentru stocarea datelor, se va baza pe o bază de date relațională ce va conține 3 tabele (Users, Contacts și Groups). Tabelul Users conține datele utilizatorului cerute la înregistrare și o referință (contactId) către un contact care va reprezenta datele suplimentare pe care utilizatorul le va adăuga eventual.

Tabelul Contacts conține datele unui contact adăugat de către un utilizator și două referințe. O referință către utilizatorul care a creat contactul (userId) și o referință către grupul căruia aparține contactul (groupId).

Tabelul Groups conține doar numele fiecărui grup.

Schema bazei de date este următoarea:



Aplicația se va baza pe design patternul MVC. Fiecare pagină va avea 3 componente.

1. View

Această componentă se va ocupa de afișarea paginii.

2. Model

Prin această componentă se vor face modificări asupra aplicației cum ar fi modificări asupra datelor din baza de date și diferite "calcul" ce vor schimba ce va afișa view-ul.

3. Controller

Controllerul este componenta care "leagă" cele 2 componente model și view ale unei pagini. Ea decide prin rezultatele de la componenta model ce va afișa prin componenta view.

| Pagina | Controller | Cerere | View | Model |
|--|-----------------------------|--|-----------------------|------------------------|
| Listarea contactelor | index-controller.php | Filtrarea contactelor | index-view.php | index-model.php |
| Vizualizarea profilului / unui contact | contact-controller.php | Vizualizarea unui contact sau a profilului | contact-view.php | contact-model.php |
| Logare / Inregistrare | auth-controller.php | Opțiunea de înregistrare sau logare | auth-view.php | auth-model.php |
| Gestionarea grupărilor | groups-controller.php | Actualizarea grupărilor sau crearea unor noi grupări | groups-view.php | groups-model.php |
| Editarea profilului / unui contact | edit-contact-controller.php | Actualizarea datelor | edit-contact-view.php | edit-contact-model.php |
| * | * | Afișarea meniului de navigare | nav-view.php | |

Luăm spre exemplu pagina principală în care dorim să facem o filtrare după Școala / Facultate a contactelor. După aplicarea filtrului se predă parametrul controllerului. Controllerul preia parametrul, apelează funcția pentru filtrare din model (care face o interogare la baza de date) și așteaptă răspunsul care va fi o listă cu toate contactele ce respectă condiția din filtru. Controllerul apoi apelează view-ul oferindu-i acea listă de contacte, acesta urmând să afișeze acele contacte.

Pentru a persista utilizatorii conectați se va folosi un cookie care va avea ca informație un JWT (JSON Web Token) care va conține 3 părți:

1. Header: ce va conține algoritmul de criptare și tipul tokenului (JWT).
2. Payload: conține id-ul utilizatorului
3. Semnatura: reprezintă headerul criptat + payloadul criptat + un secret, acest întreg mesaj fiind și el criptat.

Formatul acestui JWT va fi: hhhhh.ppppp.ssssss .

4. Diagrame use case

