

**Sistem de gestionare a unui cabinet medical**

***Document de analiză și proiectare***

*Student: Miklós Zsolt*

*Grupa: CSC*

Obiectivul lucrării.

Dezvolarea unei aplicații care poate fi utilizată într-un cabinet medical. Aplicația va avea 3 tipuri de utilizatori: medic, asistent și administrator.

Utilizatorii de tip medic pot efectua următoarele operații după autentificare: Vizualizarea și actualizarea (simptome, diagnostic, tratament) fișelor medicale ale propriilor pacienți; Filtrarea propriilor pacienți după următoarele criterii: diagnostic, tratament.

Utilizatorii de tip asistent pot efectua următoarele operații după autentificare: Operații CRUD în ceea ce privește persistența pacienților (mai puțin actualizari legate de simptome, diagnostic, tratament); Filtrarea pacienților după următoarele criterii: medic, diagnostic, varstă.

Utilizatorii de tip administrator pot efectua următoarele operații după autentificare: Operații CRUD pentru informațiile legate de utilizatori; Vizualizarea listei tuturor utilizatorilor.

Instrumente utilizate:

* Limbaj de programare: Java
* Baza de date: MySQL

Instrumente de dezvoltare: Intellij, XAMPP.

Justificarea limbajului de programare ales.

Am ales Java pentru dezvoltarea aplicației datorită beneficiilor sale precum portabilitatea, securitatea și simplitatea. Java este un limbaj de programare popular și puternic care oferă suport pentru programarea orientată pe obiecte, ceea ce face mai ușoară dezvoltarea și menținerea unei aplicații complexe precum cea pentru cabinetul medical. În plus, Java are o comunitate puternică și o documentație bogată, ceea ce face mai ușoară depistarea și rezolvarea problemelor.

Descrierea diagramelor UML:

Daigrama cazurilor de utilizare cuprinde

* Conține cei trei actori (utilizatorii aplicatiei) si acțiunile acestora.
* Un utilizator poate fii un medic, asistent sau administrator.

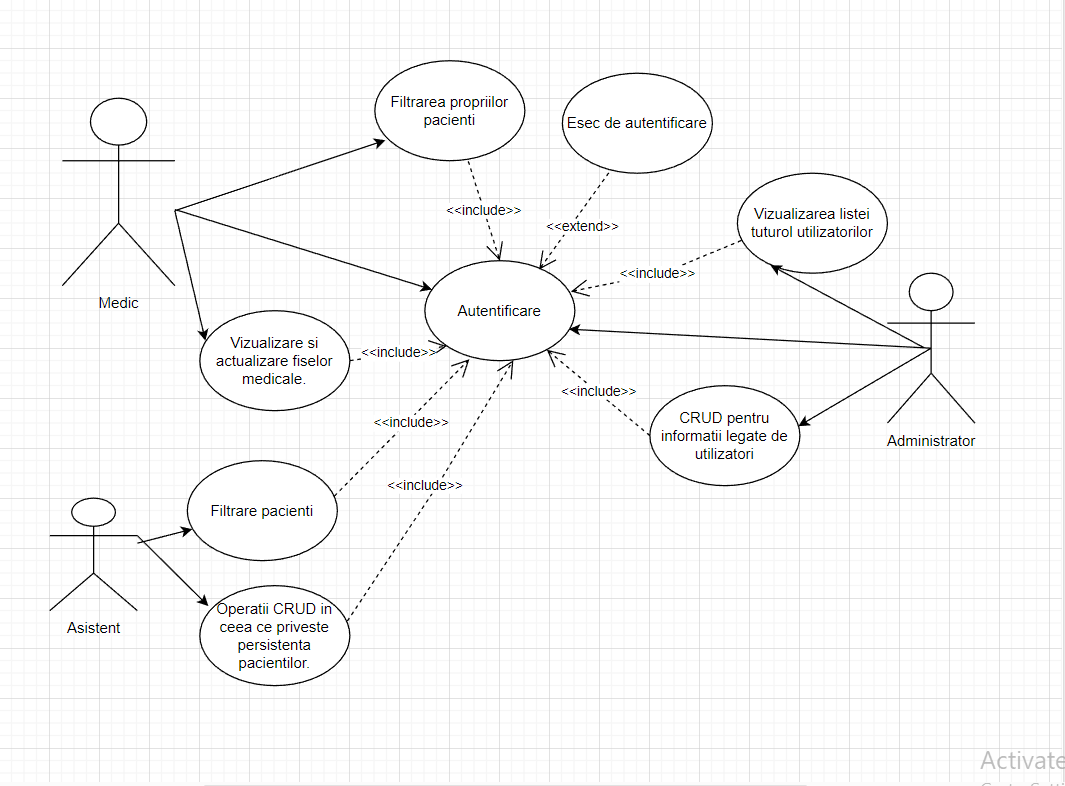


Diagrama de clase.

Diagrama de clase este un instrument folosit în cadrul procesului de dezvoltare a software-ului și face parte din limbajul unificat de modelare UML (Unified Modeling Language). Scopul acestei diagrame este de a reprezenta structura și organizarea entităților dintr-un proiect software, precum și relațiile dintre acestea.

Fiecare entitate este reprezentată în diagrama de clasă printr-o clasă, cu atribute și metode specifice.

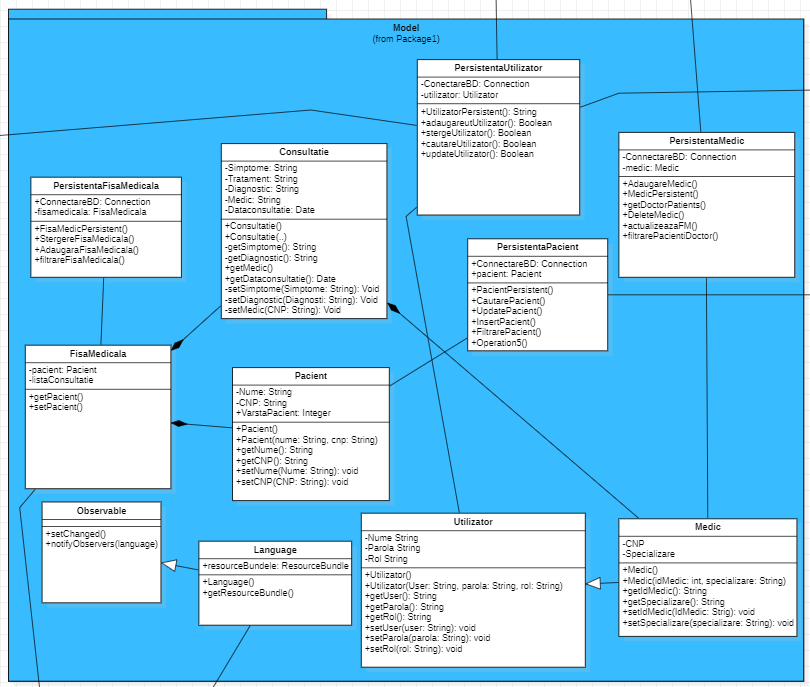
Diagrama de clase pentru arhitectura MVC (Model-View-Controller) descrie modul în care entitățile și interacțiunile dintre acestea sunt organizate într-o aplicație software. În cazul acestei aplicații, diagrama este realizată folosind arhitectura MVC și este împărțită în trei pachete principale: Model, View și Controller.

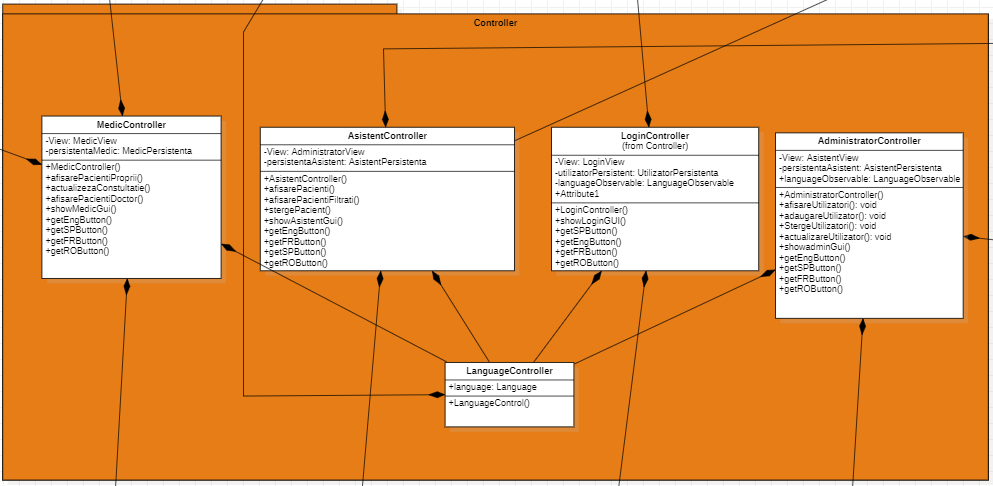
• Pachetul Model conține entitățile aplicației și se ocupă de logica de acces și manipulare a datelor. În acest pachet se găsesc clase precum Utilizator, Pacient, Medic și FisaMedicala, iar pentru persistența datelor, clase separate numite PersistentaMedic, PersistentaFisamedicala, UtilizatorPersistența, PacientPersistenta.

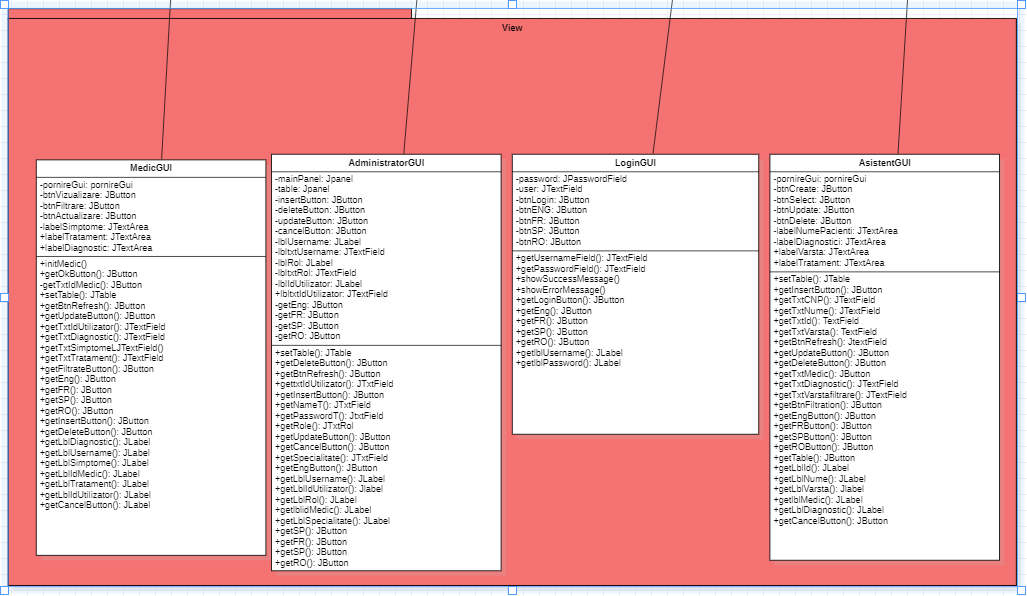
• Pachetul View conține interfețele și clasele responsabile de prezentarea datelor utilizatorului, folosind informațiile furnizate de Model și Controller. În acest pachet se găsesc clase precum AdministratorView, MedicView, AsistentView și LoginView.

• Pachetul Controller reprezintă componenta care gestionează interacțiunea dintre Model și View. Acesta se ocupă de preluarea evenimentelor și de gestionarea logicii de afișare și manipulare a datelor. În acest pachet se găsesc clase precum LoginController, AdministratorController, MedicController și AsistentController.

În esență, arhitectura MVC separă aplicația în trei componente distincte, cu roluri clare și bine definite. Această separare permite dezvoltatorilor să modifice sau să extindă aplicația într-un mod mai eficient, fără a afecta celelalte componente ale aplicației.

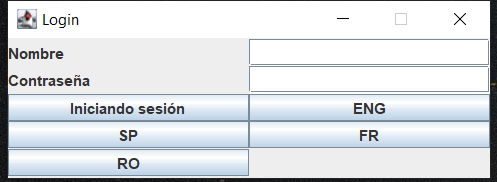






Descrierea Aplicatiei.

La deschiderea aplicației, va fi prezentată fereastra de autentificare, unde utilizatorul va putea numele de utilizator și parola corespunzătoare contului său. După autentificare, rolul utilizatorului va fi determinat și va fi afișată fereastra corespunzătoare rolului respectiv. În cazul în care informațiile introduse de utilizator nu sunt corecte, o eroare de autentificare va fi semnalată și utilizatorul nu va avea acces la aplicație până la furnizarea informațiilor de autentificare corecte.



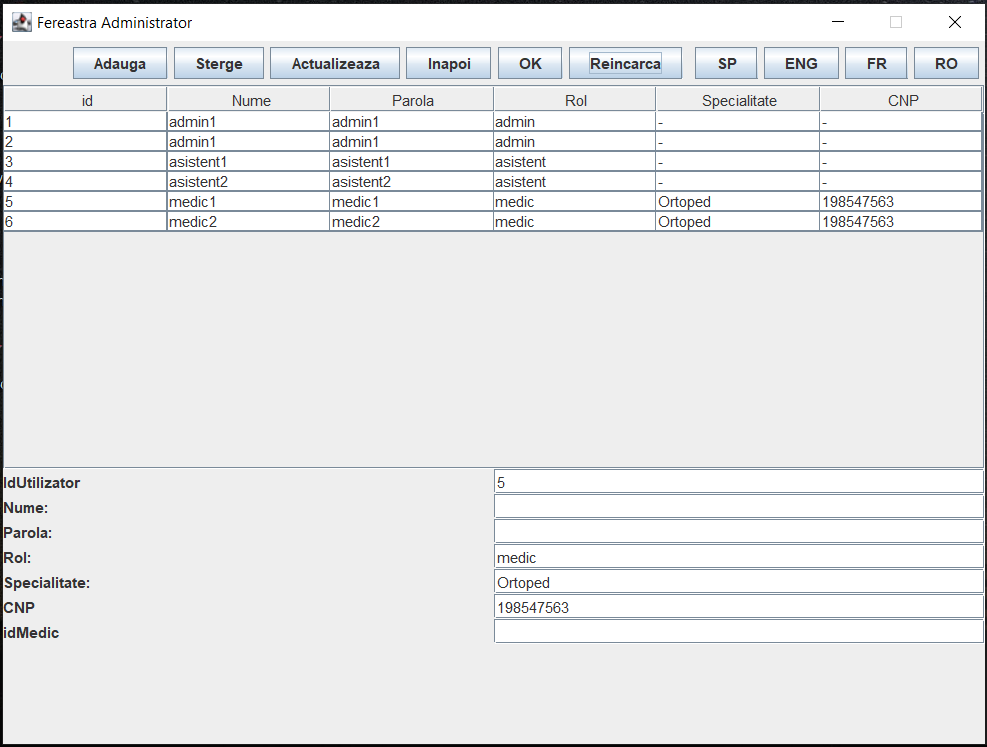
Dacă utilizatorul se autentifică ca un administrator poate efectua următoarele:

• Adăugarea unui utilizator

• Actualizarea utilizatorului

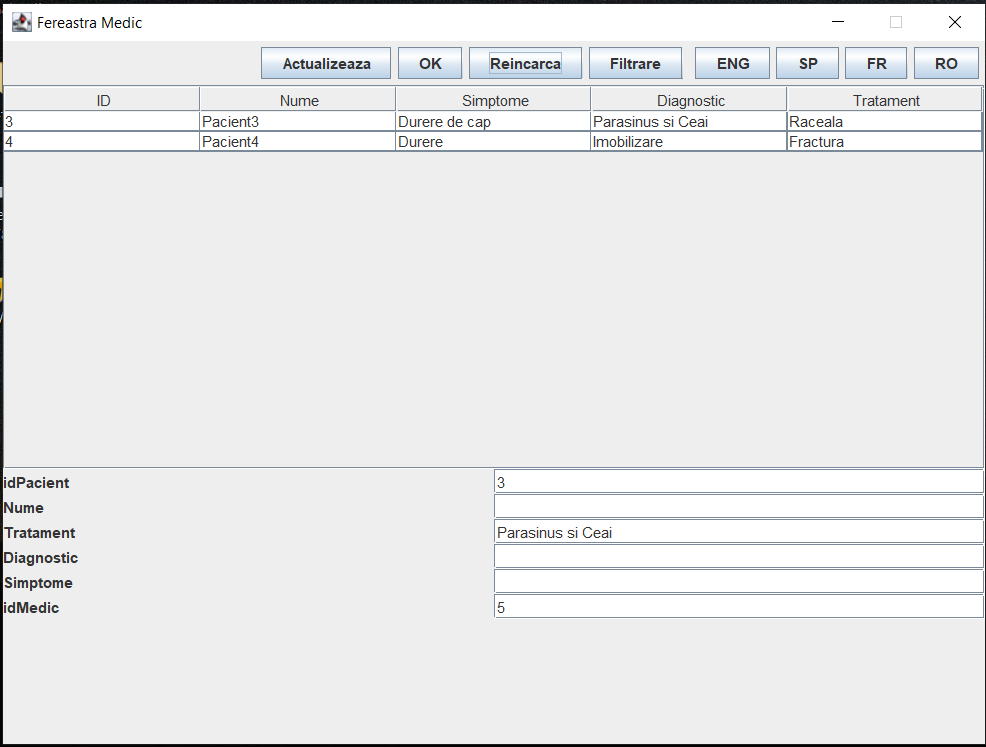
• Ștergerea utilizatorului

• Afișarea lista cu utilizatori.



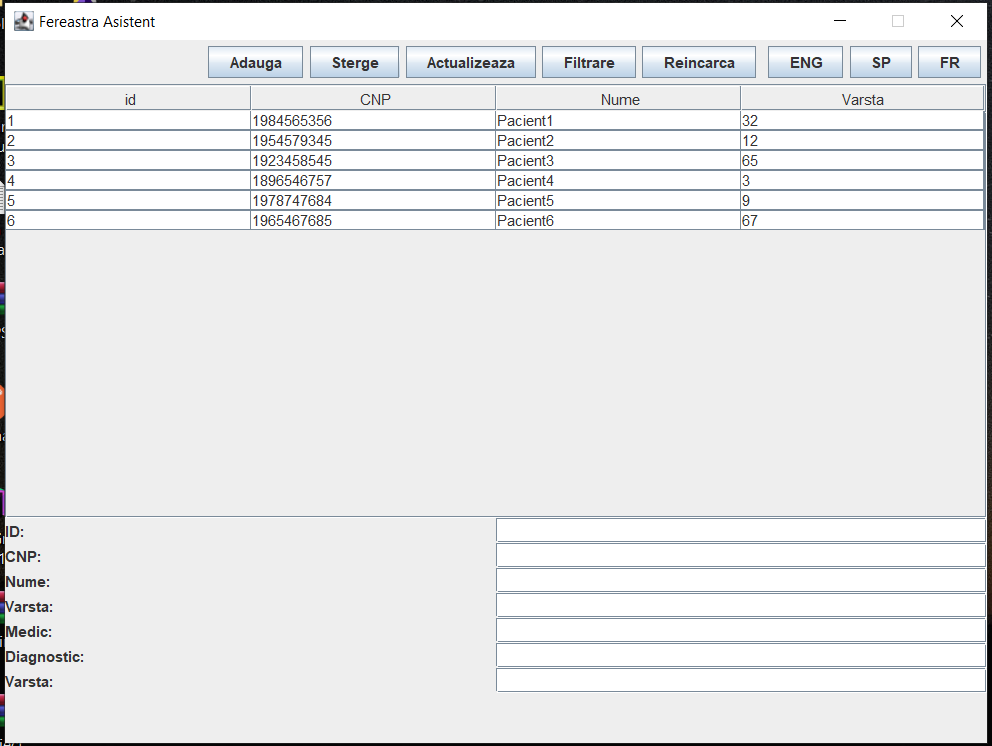
Dacă utilizatorul se autentifică ca un Medic poate efectua următoarele:

* Vizualizarea și actualizarea (simptome, diagnostic, tratament) fișelor medicale ale propriilor pacienți;
* Filtrarea propriilor pacienți după următoarele criterii: diagnostic, tratament.



Dacă utilizatorul se autentifică ca un Asistent poate efectua următoarele:

* Stergerea pacienților
* Adăugarea pacienților
* Actualizarea pacienților (mai puțin actualizari legate de simptome, diagnostic, tratament)
* Filtrarea pacienților după următoarele criterii: medic, diagnostic, varstă.



Aplicatia este disponibila in 3 limbi inafara de limba romana. Aceste 3 sunt: engleza, spaniola si franceza.