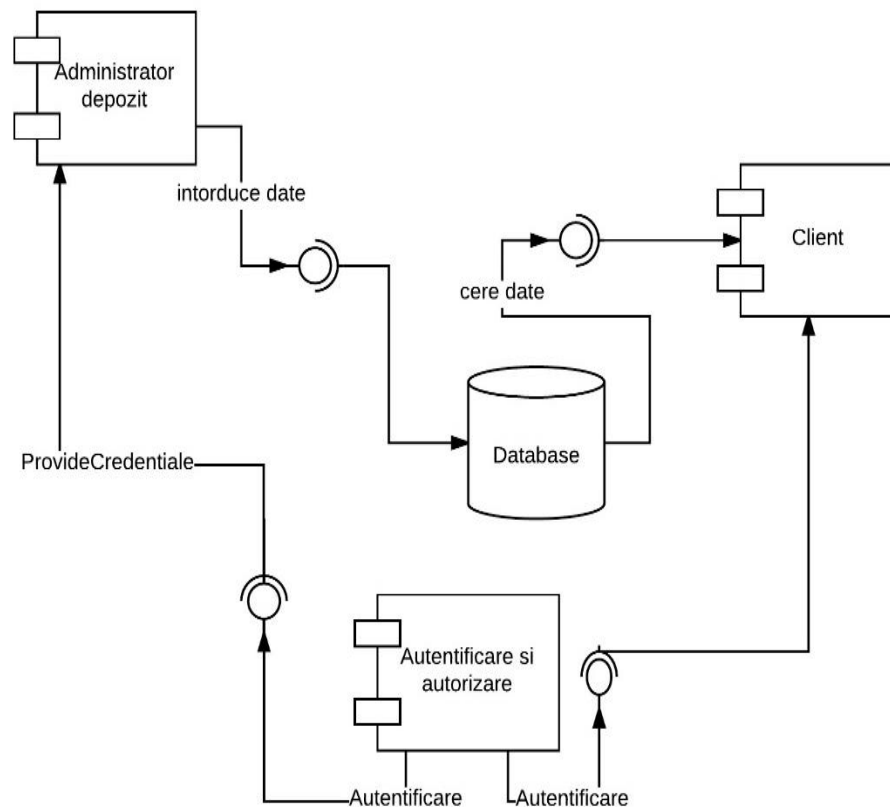


Analiza unui system software la alegere

Aplicație pentru gestionarea stocurilor din depozite

1. Prezentarea cerințelor funcționale și non-funcționale și identificarea acelor cerințe care influențează arhitectura
 - a. Cerințe funcționale:
 - O aplicație pentru administratorul depozitului pentru introducerea noilor stocuri și modificarea datelor introduse
 - O aplicație pentru client care poate consulta stocurile și efectuarea unor comenzi
 - b. Cerințe non-funcționale:
 - Timpul de răspuns minim
 - Realizarea aplicației pentru administrator astfel încât să fie compatibilă cu versiunile de Windows 7,8,10
 - Livrarea softului într-un termen de 7 luni
 - c. Cerințe care influențează arhitectura:
 - Folosirea unei baze de date
 - Aplicația trebuie să autentifice fiecare utilizator și informația trebuie trimisă criptat

2. Descompunerea în componente, definirea responsabilităților componentelor și a relațiilor dintre ele; argumentare:

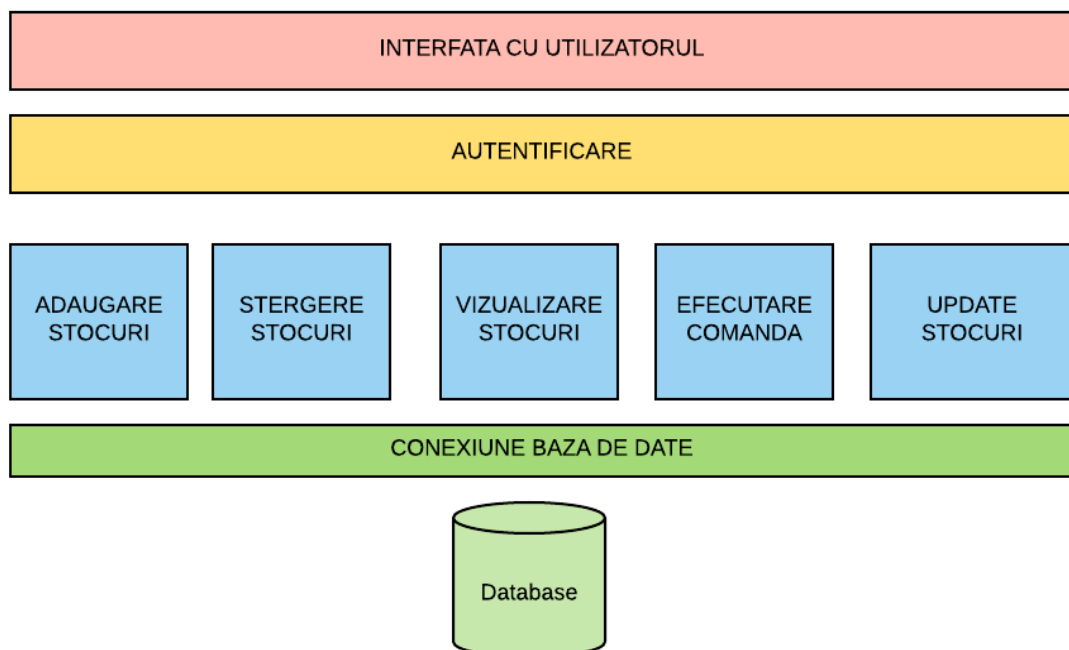


Vor exista doua componente, una pentru aplicatia destinata administratorului depozitului care va putea sa introduca stocuri sau sa le editeze si o componenta client, reprezentata prin aplicatia destinata clientului. Aplicatia va contine o baza de date care va putea fi utilizata de ambele componente, dar fiecare avand drepturi diferite, obtinute in urma autentificarii.

3. Prezentarea sistemului software din două perspective(o diagram pentru fiecare perspectivă + explicații):

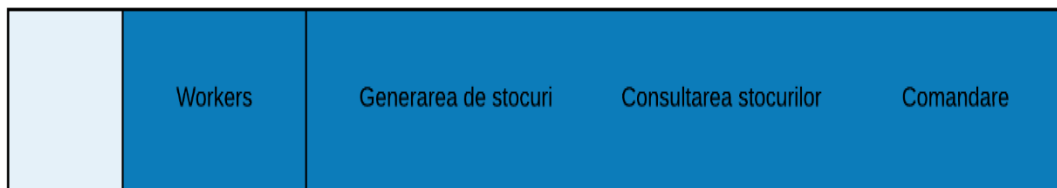
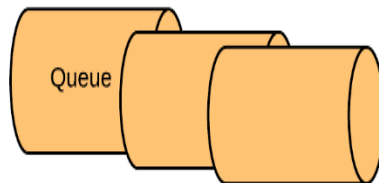
a. Perspectiva Logica

- Diagrama scoate in evidenta principalele componente ale aplicatiei, partea de autentificare are rolul de a acorda aprobari pentru efectuarea operatiunilor pe aplicatie. Aplicatia este impartita in mai multe module pe care le poate utiliza clientul sau administratorul si baza de date in care se vor salva toate informatiile referitoare la stocuri



b. Perspectiva Proces

- Cererile se vor pune intr-o coada de asteptare si vor fi mai apoi trimise ca si raspuns clientului



4. Identificarea celor mai importanți 3 indicatori de calitate, specificarea măsurii alese pentru fiecare indicator de calitate și argumentarea alegerilor:
 - a. Toleranta la modificari: folosind module , se pot modifica, scoate sau adauga functionalitati
 - b. Securitate: datele vor fi trimise criptat catre component de autentificare
 - c. Performanta: prin asigurarea unor componente hardware si a unora software adecvate

5. Identificarea tehnologiilor middleware folosite pentru a comunica între componente:
 - a. Modificarea datelor din baza de date se va face prin query
 - b. Pentru conexiunea cu baza de date se va folosi SQL Connection

6. Identificarea principalelor modele și stiluri arhitecturale folosite:
 - a. Client-server: folosind serverul pentru pastrarea datelor, astfel aplicatia client va consuma mai putine resurse si va exista posibilitate de backup a datelor
 - b. Coada de mesaje: tehnologiile pot procesa mii de mesaje pe secunda

7. Prezentarea scenariilor de validare a arhitecturii:
 - a. Mai multi clienti vor sa faca comenzi
 - b. Administratorul schimba datele in timp ce un client le vizualizeaza
 - c. Clientii incearca operatiuni la care nu au acces