**Aplicatie Client-Server pentru criptarea/decriptarea fisierelor text**

**1.Prezentarea cerintelor functionale si non-functionale si identificarea acelor cerinte care influenteaza arhitectura**

**1.1.Cerinte functionale**

Aplicatia va fi o aplicatie Desktop Client-Server unde un utilizator se inregistra sau loga pentru a cripta/decripta fisiere text.

Utilizatorul va incarca fisierul ce doreste sa-l cripteze/decripteze iar dupa va primi inapoi fisierul decriptat/criptat si il poate salva pe propriul PC.

Dupa ce fisierul a fost incarcat pe server, intr-o baza de date vor fi salvate numele fisierului, user-ul care a criptat acel fisier si parametrii criptosistemului.

Informatiile unui utilizator adica numele, username-ul si parola vor fi de asemenea criptate si stocate in baza de date.

**1.2.Cerinte non-functionale**

* Constrangeri tehnologice

Pentru dezvoltarea aplicatiei se va folosi limbajul de programare C#, baza de date va fi stocata in cloud.

* Constrangeri impuse de politica firmei

Un utilizator nu poate cripta un fisier cu acelasi nume de doua ori si nici nu poate decripta un fisier daca la criptare a avut un nume diferit.

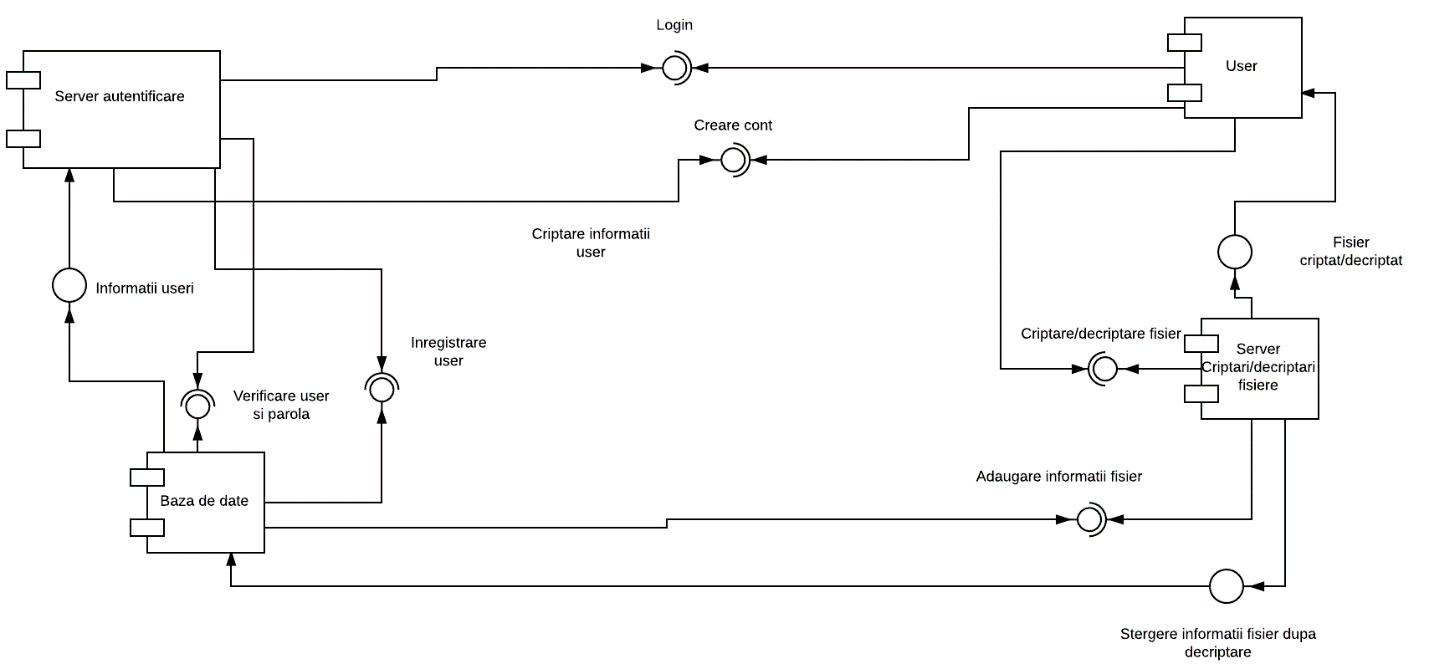
* Cerinte care influenteaza arhitectura

Toate schimburile de date intre client si server trebuie transmise criptate.

Nu este acceptata pierderea mesajelor/informatiilor, fisierele nu vor mai fi criptate/decriptate corespunzator.

Programarea trebuie facuta in C# pentru a se folosi staful de dezvoltare existent si pentru a nu mai fi nevoie de documentarea si instruirea staff-ului, lucru ce ar amana lansarea aplicatiei. Criptarea se va face cu RSA.

**2. Descompunere in componente, definirea responsabilitatilor componentelor si a relatiilor dintre ele**

****

Cele 4 componente ale arhitecturii sunt: serverul de autentificare,user,baza de date si serverul pentru criptari/decriptari de fisiere.

**Serverul de autentificare** are rolul de a valida identitatea unui user, serverul verifica daca in baza de date exista userul respectiv, daca exista extrage din baza de date parola, si parametrii RSA, dupa care cripteaza parola introdusa de user si o compara cu cea din baza de date, daca cele doua parole corespund userul va primi acceptul.

Daca un user doreste sa-si creeze un cont nou el va trebui sa introduca un username si o parola, serverul verifica daca exista acest nume deja in baza de date, daca nu exista se va salva in baza de date numele,parola si parametrii RSA.

**Userul** isi se poate loga sau isi poate crea un cont. Dupa ce s-a el va putea incarca fisiere pentru a le cripta/decripta, si va primi inapoi fisierul criptat/decriptat si il va putea salva pe propriul PC.

**Serverul pentru criptari/decriptari** de fisiere va cripta/decripta fisierele incarcate de user.

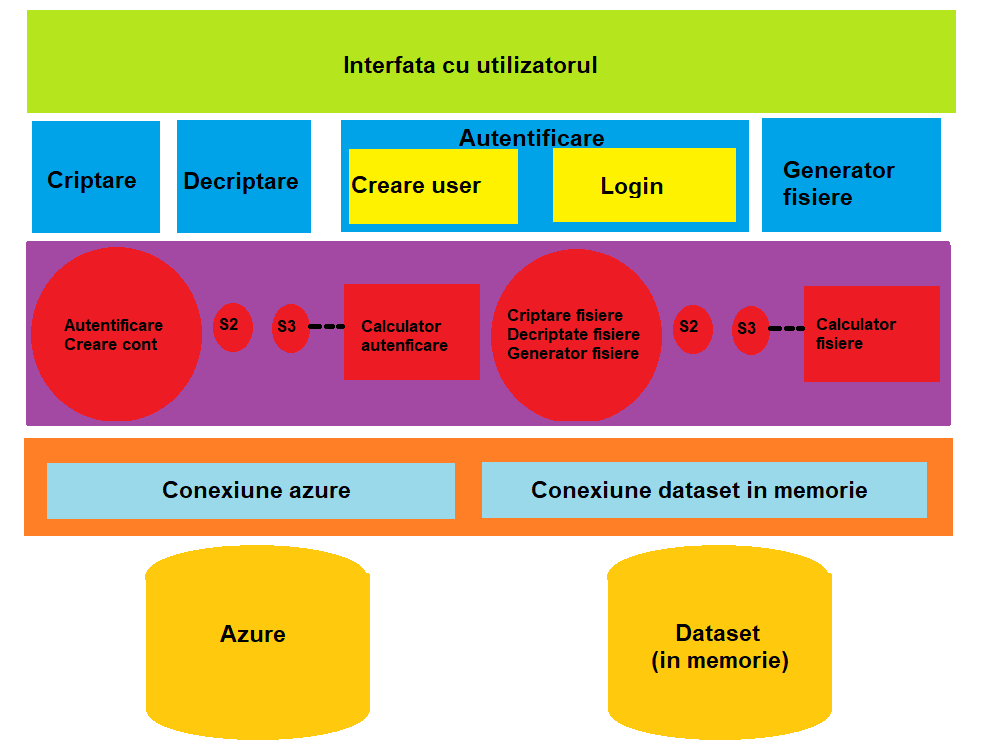
La criptare se va salva numele fisierului incarcat de user, userul care l-a incarcat si parametrii de criptare in baza de date si se va genera un fisier nou ce va fi descarcat de catre utilizator.

La decriptare se serverul va cauta in baza de date numele fisierului incarcat de utilizator, verifica daca exista acel fisier in baza de date si daca tot acel user l-a criptat, extrage paramaterii de criptare din baza de date si genereaza fisierul decriptat pe care userul il va descarca.

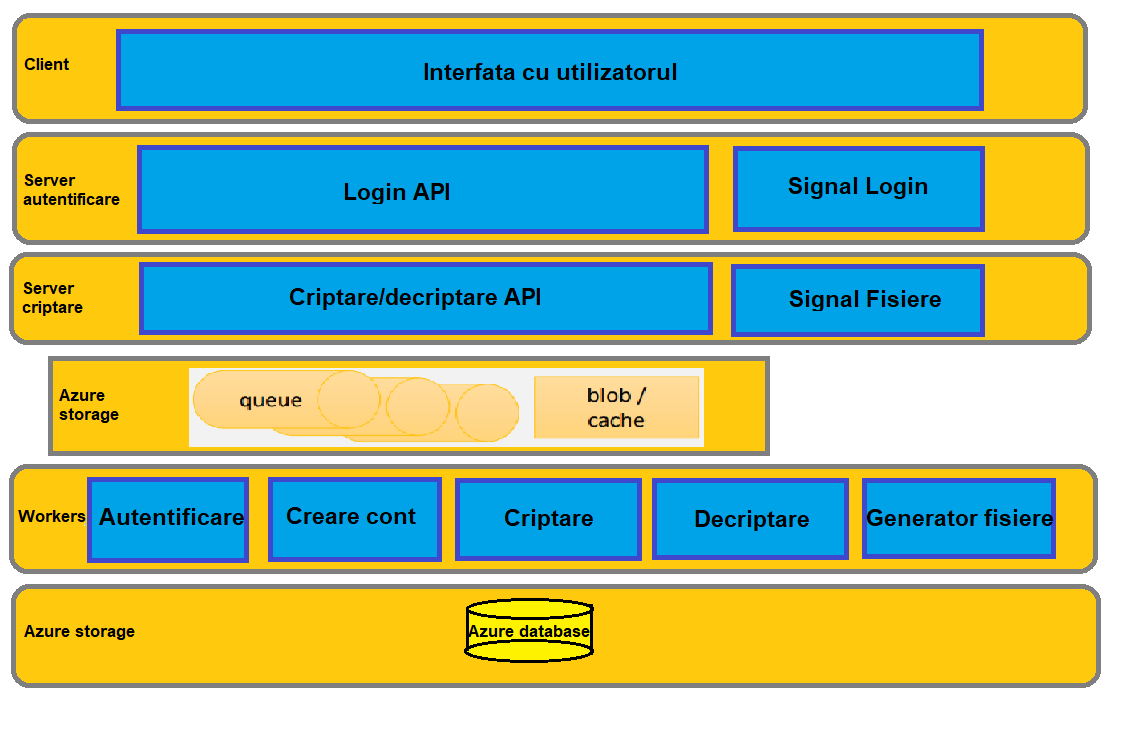
In **baza de date** vor fi doua tabele relationate prin numele userului. Intr-o tabela vor fi salvate informatiile userului, adica nume,parola si parametrii RSA iar in cealalta tabela numele userului, numele fisierului si parametrii de criptare.

**3. Prezentarea sistemului software din doua perspective**

**Perspectiva logica**

****

**Perspectiva process**

****

Interfata cu utilizatorul ofera posibilitatea creari unui cont sau autenficarii daca userul are deja un cont.

Un request pentru validarea autentificarii este trimis catre serverul de login iar daca va fi validat cu succes el va fi redirectionat

catre o alta fereastra de unde va putea cripta/decripta iar mai apoi va primi fisierul decriptat/criptat care il va putea salva pe propriul PC

Toate date vor fi salvate intr-o baza de date folosind cloud-ul Azure.

**4. Identitifcarea celor mai importanți 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate**

1.Securitate - Toate conexiunile trebuie autentificate iar informația transmisă criptata.

2.Fiabilitate - Pierderea mesajelor nu este acceptata, fisierele nu mai pot fi decriptate in mod corespunzator.

3.Peformanta – Aplicatia trebuie sa permita user-ului sa descarce fisierul nou creat in maxim 30 de secunde in 80% din cazuri.

**5. Identificarea tehnologiilor middleware folosite pentru a comunica intre componente**

Pentru partea de comunicare intre user si servere si intre server si Azure a fost realizata prin intermediuri mai multor form-uri realizate in limbajul C# folosind Microsoft Visual Studio 2015.

**6. Identificarea pincipalelor modele și stiluri arhitecturale folosite**

Sistemul software trebuie să răspundă unei varietati de parti interesate, cum ar fi, proprietarii, utilizatorii și developerii. Aceste părți interesate sunt preocupate în ceea ce privește sistemul. Echilibrarea acestor preocupări și demonstrarea modului în care acestea sunt abordate fac parte din proiectarea sistemului.

Documentația arhitecturii arată că toate preocupările părților interesate sunt abordate prin modelarea și descrierea arhitecturii din puncte de vedere separate, asociate diferitelor preocupări ale părților interesate

Arhitectura unui sistem software este mai strâns legata de atributele sale de calitate, cum ar fi toleranța la erori, compatibilitatea cu mai multe platforme, extensibilitatea, fiabilitatea, întreținerea, disponibilitatea, securitatea, gradul de utilizare și alte asemenea facilități. Întrebările părților interesate se traduc adesea în cerințe privind aceste atribute de calitate, numite în mod diferit cerințe nefuncționale, cerințe extrafuncționale, cerințe comportamentale sau cerințe privind atributele de calitate.

**7. Prezentarea scenariilor de validare a arhitecturii**

Validarea arhitecturii se va realiza prin urmatoarele scenarii:

Crearea unui user care deja exista pentru a verifica ca nu se adauga username-ul din nou in baza de date.

Logarea unui user cu o parola gresita si logarea fara introducera parolei sau userului pentru a verifica ca un utilizator nu se poate primi eronat accept pentru criptare/decriptare.

Un user sa decripteze un fisier criptat de alt user.

Verificarea standardelor de performanta, se va masura timpul de generare a fisierelor pentru 50 de useri, cel putin 40 trebuie sa-l primeasca in maximum 30 de secunde.

Verificarea corectitudinii informatiilor care sunt incarcate pe cloud, se verifica daca datele sunt introduse corect in baza de date.

Testarea programului de mai multi user simultan, se verifica daca aplicatia nu se va bloca daca mai multi user folosesc programul simultan.

Acoprerirea codului si majoritatea conditiilor folosind uniteste pentru detectarea altor eventuale erori nedectate anterior.