**Aplicație pentru acces diferențiat la zona de stocare**

**-Proiect POO-**

Prof. Coord.: Arseni Ștefan-Ciprian

Grupa: C112 A

Studenți: Bălăcescu Anamaria

Clima Tudor Traian



Cuprins

Contents

[**Tabel versiuni** 3](#_Toc129805615)

[**Capitolul 1 – Introducere** 4](#_Toc129805616)

[1.1. Scopul proiectului 4](#_Toc129805617)

[1.2. Lista definițiilor 4](#_Toc129805618)

[1.3. Structura DSC 4](#_Toc129805619)

[**Capitolul 2 – Descrierea generală a produsului software** 4](#_Toc129805620)

[2.1. Descrierea produsului software 4](#_Toc129805621)

[2.2. Descrierea platformei HW/SW 5](#_Toc129805622)

[**Capitolul 3 – Descrierea cerințelor** 6](#_Toc129805623)

[3.1. Cerințele funcționale 6](#_Toc129805624)

[3.2. Cerințele non-funcționale 6](#_Toc129805625)

# **Tabel versiuni**

|  |  |  |
| --- | --- | --- |
| Versiune | Client – specificații | Server - specificații |
| Versiune 1.0 | Posibilități de înregistrare, autentificare, vizualizare și modificare fișiere | Stocare informații de autentificare, memorare și criptare de fișiere, istoric autentificări și actualizare baza de date. |

# **Capitolul 1 – Introducere**

## 1.1. Scopul proiectului

Programul are ca scop stocarea și diferențierea accesului la datele din zona de stocare.

Fiecare client își alege la crearea contului nivelul de acces. În funcție de acest nivel poate accesa datele stocate sub același cod de identificare și să le modifice în funcție de acțiunile permise la acest nivel. De asemenea, în funcție de preferințe, accesul la spațiul de stocare poate fi facilitat prin comenzi rapide de remember me și/sau keep me.

## 1.2. Lista definițiilor

## 

|  |  |
| --- | --- |
| Abreviere | Definiție |
| App | Aplicație |
| DSC | Document Cerințe Software |
| HW | Hardware |
| SW | Software |
| Owner | Proprietar al fișierului în cauză |

## 1.3. Structura DSC

Documentul este împărțit în 3 capitole: capitolul 1 reprezintă Introducerea în tema proiectului (scopul proiectului, lista definițiilor), capitolul 2 prezintă descrierea generală a cerințelor software cu particularitățile stabilite inițial pentru funcționalitățile software, elementele hardware și constrângerile necesare realizării aplicației propriu-zise, urmând o detaliere a cerințelor, funcționale și non-funcționale, în capitolul 3.

# **Capitolul 2 – Descrierea generală a produsului software**

## 2.1. Descrierea produsului software

În prezent, majoritatea domeniilor au fost supuse unei transformări de regim tehnologic. Acest lucru presupune automatizarea sistemelor și gestiunea datelor printr-o platformă universală cu acces la o unică zona de stocare cu acces diferențiat la fișiere astfel încât, fiecare utilizator să aibă intimitatea personală de lucru, dar progresul în muncă să poată fi supravegheat de administrator. Problemele principale pe care le vizează aplicația noastră sunt diferențierea accesului și securitatea infomațiilor stocate.

Contextul actual, care permite regimul de lucru remote, impune totodată posibilitatea de conectare la un mediu de colaborare universal. Prin urmare, aplicația noastră oferă utilizatorilor o interfață prietenoasă, ușor de folosit prin intermediul căreia aceștia pot acesa zona de stocare a datelor.

Din motive de securitate, la crearea fiecărui cont parola utilizatorului va fi criptată și memorată în baza de date pentru evitarea „furtului de informație” în cazul în care o persoană neautorizată ar dobândi acces la baza de date.

După înregistrarea și onectarea la rețea prin intermediul internetului, utilizatorul intră în mediul de lucru unde poate crea, modifica, descărca sau șterge fișiere în funcție de necesitățile sale. Pentru a evita acțiunile malițioase la adresa unuia dintre angajați și pierderea sau sustragerea de informații de către alți utilizatori, fiecărui cont i se va atribui o zonă proprie din spațiul de stocare (acest acces diferențiat se va face pe baza unei chei de criptare aferente fiecărui owner) care va putea fi accesată doar de owner sau de administratorul care va avea acces la întreg spațiul de stocare.

Pentru a facilita descoperirea unor eventuale acțiuni malițioase sau a persoanelor responsabile de anumite defecțiuni aduse sistemului, fiecare acțiune de conectare va fi memorată la nivelul bazei de date într-un Histroy log, iar cele considerate malițioase de către app vor genera un mesaj de eroare. De asemenea, pentru a evita pierderea de informații, baza de date se va actualiza automat după fiecare modificare adusă.

Luând o situație ipotetică, vom considera o firmă multinațională cu angajați din diferite țări care lucrează în regim remote. Alina, Albert și John sunt angajați ai acestei firme cu roluri de useri normali și respectiv administrator la nivelul platformei.

După prima autentificare Alina își încarcă pe platformă 4 documente aferente departamentului său. Realizează că unul dintre ele este personal și nu ar trebui să facă parte din baza de date a firmei. Îl șterge și se deconectează. Toate modificările sunt salvate automat.

Albert se conectează la platformă, își deschide mediul de lucru, descarcă 2 documente din folderul public cu elemente comune, necesare tuturor angajaților firmei, își termină task-urile și încarcă 1 document ce conține munca lui pe ziua respectivă, după care se deconectează.

John, fiind administrator, are printre task-urile sale verificarea îndeplinirii îndatoririlor celorlalți anagajați. El accesează prima dată folderul Alinei, observă și își notează într-un document personal modificările aduse documentului, îl închide și repetă procesul pentru fișierele lui Albert. În final încarcă documentele sale în fișierul personal și se deconectează.

La nivelul bazei de date vom reține atât datele celor 3, cât și acțiunile de conectare/deconectare. Un posibil scenariu de history log în acest caz ar putea fi:

1. Alina has connected to the platform. 12/03/2023 08:05
2. Alina has disconnected from the platform. 12/03/2023 14:55
3. Albert has connected to the platform. 12/03/2023 15:05
4. Albert has disconnected from the platform. 12/03/2023 19:55
5. John has connected to the platform. 12/03/2023 20:00
6. John has disconnected from the platform. 12/03/2023 23:57

## 2.2. Descrierea platformei HW/SW

Pentru testarea aplicației, atât serverul cât și utilizatorul au nevoie de o conexiune la internet și de platforme de tip Windows x32 sau mai recente.

# **Capitolul 3 – Descrierea cerințelor**

## 3.1. Cerințele funcționale

1. Autentificare utilizator (Login) – utilizatorul completează formularul de autentificare cu credențialele sale, acestea sunt preluate de frontend și transmise mai departe către backend ca request pentru verificare, se caută în baza de date și în cazul în care sunt găsite se redirecționează utilizatorul către ecranul de autentificare reușită.
2. Înregistrare utilizator – utilizatorul completează formularul de înregistrare care este preluat de frontend și trimis mai departe către backend care supune datele unor verificări de formatare și le verifică autenticitatea în sistem; dacă datele trec de acești pași, ele sunt memorate la nivelul bazei de date după criptarea aferentă și este transmis un răspuns pozitiv înapoi către backend.
3. Acces diferențiat la spațiul de stocare – utilizatorii de tip „regular” vor primi un ecran care va conține doar fișierele proprii, în timp ce un utilizator de tip „administrator” va primi un ecran care să conțină toate fișierele, atât cele personale, cât și pe cele ale utilizatorilor „regular” din spațiul de stocare.
4. Upload files – utilizatorul încarcă un fișier pe care frontend-ul îl trimite mai departe către backend pentru stocare în zona de memorie aferentă hash-ului owner-ului.
5. Descărcare fișiere – utilizatorul trimite o cerere de descărcare fișier către frontend, care îl trimite mai departe către backend; după ce corespondența hash-ului utilizatorului cu cel al fișierului este verificată, se preia conținutul din baza de date și este transmis ca răspuns utilizatorului.
6. Deconectare (Logout) – utilizatorul selectează opțiunea de logout (reprezentată de un buton) și trimite prin intermediul frontendului o cerere de deconectare spre backend și apoi spre baza de date unde se înregistrează acțiunea la nivelul istoricului de conectare și utilizatorul este retrimis la homepage ca și răspuns.

## 3.2. Cerințele non-funcționale

* Actualizare bază de date – după orice modificare (încarcare/ștergere de fișier) adusă zonei de stocare, backend-ul furnizeză un cod autocompilat de rememorare a zonei de memorie alterate peste locația inițială din baza de date.
* Sincronizarea automată a bazei de date - după orice modificare adusă zonei de stocare, sistemul trebuie să actualizeze automat baza de date pentru a reflecta aceste modificări.
* Verificarea integrității datelor - înainte de a actualiza baza de date, să se efectueze verificări pentru a asigura integritatea datelor și pentru a preveni coruperea bazei de date.
* Memorare date de logare – după primirea formularului de înregistrare și validarea datelor primite de backend, username-ul și hash-ul parolei sunt memorate automat la nivelul bazei de date.
* Securizarea informațiilor (Criptare parolă) – folosind metode criptografice pentru autenticitate, autentificare si constanta continutului datelor.
* Implementarea criptării parolelor - toate parolele trebuie să fie criptate astfel încât să nu fie accesibile persoanelor neautorizate.
* Asigurarea confidențialității datelor - în plus față de criptarea parolilor, să se cripteze și datele sensibile stocate în baza de date sau transferate între diferite sisteme.
* Înregistarea acțiunilor în cadrul aplicației (History log) – conectările și deconectările din cadrul platformei vor fi înregistrate automat la nivelul bazei de date din motive de securitate.