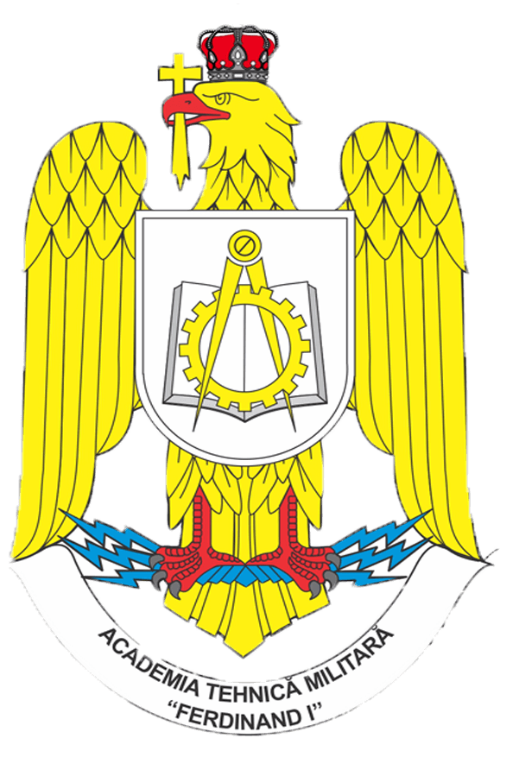
**Academia Tehnică Militară**

**“Ferdinand I”**



**Proiect Programare Orientată pe Obiecte**

**Livrabil nr. 1**

Profesor coordonator: Studenți:

Col. Prof. Univ. Dr. Ing. Mihai TOGAN Std. Cap. Duminica Alexandru

Std. Cap. Gugu Alexandra

Cuprins

[Capitolul 1 3](#_Toc130734714)

[1.2 Scopul documentului: 3](#_Toc130734715)

[1.2 Structura documentului: 3](#_Toc130734716)

[Capitolul 2 4](#_Toc130734717)

[2.1. Descrierea produsului 4](#_Toc130734718)

[2.2. Descrierea platformei 4](#_Toc130734719)

[Capitolul 3 5](#_Toc130734720)

[3.1. Cerințe funcționale: 5](#_Toc130734721)

[3.2. Pagina principală 5](#_Toc130734722)

[3.2 Cerințe nefuncționale: 6](#_Toc130734723)

# Capitolul 1

## 1.2 Scopul documentului:

Acest document are ca scop descrierea funcționalităților aplicației, a mecanismelor utilizate spre realizarea ei, cât și prezentarea punctuală și clară a cerințelor. De asemenea, acest document conține repere, precum și obiectivele intermediare pe care vrem să le atingem.

## 1.2 **Structura documentului:**

Aceste document este organizat în trei capitole după cum urmează:

* ***Capitolul 1,*** cel curent, este o prezentare succintă a livrabilului și a obiectivelor acestuia, menit să faciliteze parcurgerea cât mai ușoară.
* ***Capitolul 2*** prepupune o prezentarea a aplicație și a mecanismelor utilizate spre realizarea sa.
* ***Capitolul 3*** prezintă cerințele punctuale ale aplicației.

# Capitolul 2

## 2.1. **Descrierea produsului**

Scopul proiectului este acela de a realiza o aplicație de tip Version Control System, un soft ce se ocupa cu gestionarea mai multor versiuni ale unor fișiere. Aceasta va presupune configurarea, monitorizarea și administrarea unui depozit de fișiere, publicate de către clienți diferiți, fiecare având restricții de acces. Folosind un sistem de autentificare de tip Loggin/Loggout anumiți utilizatori își pot crea spații de lucru comune, pot vizualiza, accesa sau modifica sursele grupurilor de lucru din care fac parte sau pot oferi accesul altor clienți la fișierele de care dispun.

Opțiunile puse la dispoziția clientului sunt **Commit**, **Branch**, **Delete, Propertie** și **View**. Aceste au rolul de a crea un mediu de lucru de tip open-source, cât mai facil și ușor de utilizat, comeziile fiind intuitive. În contextul acestei aplicatii se încearcă o replicare a modului de funcționalitate are unor soft-uri cunoscute precum SVN și Git.

Aplicația va fi dezvoltată în limbajul de programare C++, iar interfata grafica va fi realizata cu ajutorul framewok-ului Qt. Din punct de vedere architectural, aceasta va avea două mari componente: aplicația client, o interfață grafică folosită de utilizator și o aplicație server. Conexiune dintre cele două vor fi realizate pe baza protocolului TCP/IP, oferindu-i astfel utilizatorului posibilitatea de a-și accesa și manipula datele de la distanță. Comunicarea dintre server și client se doreste a fi asincrona pentru un flux cat mai rapid și pentru a permite mai multor utilizatori să folosească aplicația simultan.

## 2.2. **Descrierea platformei**

Produsul software este dezvoltat pentru Windows 10, respective Windows 11. Vom utiliza platforma Qt Creator pentru realizarea unei interfețe prietenoase, cât și modulul Qt Network care pune la dispoziție un set de protocoale pentru programarea în rețea ce folosesc TCP/IP. Limbajul de programare utilizat va fi C++ și se vor utiliza conceptele programării orientate pe obiecte. Serverul de baze de date îl vom implementa cu ajutorul SQL Server Management Studio.

# Capitolul 3

## 3.1. **Cerințe funcționale:**

**La nivelul aplicației client:**

La momentul accesării aplicației de către client, acesta va avea de introdus un nume de utilizator și o parolă pentru a se putea conecta la server. Aplicația va oferi și posibilitatea de a crea un nou cont de utilizator. După verificarea datelor, clientul își va putea vizualiza și gestiona fișierele la care are acces. Dacă numele de utilizator și parola nu corespund cu cele salvate în baza de date, se va afișa un mesaj de eroare specific, cu posibilitatea de a reintroducere a datelor.

**La nivelul aplicației server:**

Pentru a gestiona toate conturile create, aplicatia stocheaza informațiile acestora (numele de utilizator, parola și id-urile fișierelor la care le este permis accesul) într-o bază de date.

La solicitarea unui client de a se conecta, server-ul va trebuie să verifice datele furnizate, comparându-le cu cele din baza de date și să întoarcă un mesaj de aprobare/respingere a conexiunii.

După aprobarea conexiunii, serverul va putea să trimită către client o lista a fișierelor disponibile și a tuturor versiounilor acestora.

## 3.2. **Pagina principală**

La nivelul aplicației client:

* Posibilitatea de autentificare printr-un username si o parola;
* Stocarea datelor de autentificare într-o bază de date, care va asigura integritatea și criptarea parolelor utilizatorilor;
* Posibilitate de a crea un cont de Administrator. Pe lângă funcționalitățile utilizatorilor obișnuiți acesta va dispune de următoarele:
* Accesul la toate fișierele și versiunile lor din sistem;
* Poate oferi sau restricționa accesul anumitor utilizatori la o sursa sau la mai multe;
* Poate numi la rândul său un alt Administrator;
* Un utilizator obișnuit dispune de următoarele comenzi:
  + **Commit**: Posibilitatea de a încărca o nouă versiune în spațiul de lucru;
  + **Delete**: Ștergerea unei versiuni sau a întregului spațiu de lucru (toate versiunile unui fișier);
  + **View**: Posibilitatea de a vizualiza conținutul unui fișier. În cazul în care se încearcă vizualizarea unui fișier la care accesul nu este permis, se va afișa conținutul criptat;
  + **Branch**: comanda utilizată pentru a bifurca setul de fișiere în căi de dezvoltare diferite;
  + **Properties**: comanda prin intermediul căreia se vor afișa numele utilizatorilor care au acces la un anumit fisier;
* Afișarea unui text explicativ în cazul în care sunt solicitate informații suplimentare (se apasă butonul de **HELP**);

La nivelul aplicației server:

* Adăugarea unui cont în baza de date. La momentul creării unui nou cont, se va adăuga o nouă linie în Tabela Conturi a bazei de date;
* Înainte de executarea oricărei comenzi, este verificat accesul utilizatorului la fișierul respective;
* Adăugarea unei noi versiuni, cât și a utilizatorului care a efectuat .modificările. Inainte de a adăuga o nouă versiune în sistem, se va declanșa detecția automată a modificărilor. În cazul în care se constată că nu au avut loc schimbări în fișier, comanda de **Commit** nu este executată;

## 3.2 Cerințe nefuncționale:

* Utilizarea limbajului C++ și a platformelor Qt Creator;
* Asigurarea securității, integrității și confidențialității datelor;
* Existența unui mecanism de excepții pentru a preveni utilizarea necorespunzatoare a aplicației, cu afișarea unor mesaje sugestive;
* Asigurarea realizării conexiunii cât mai eficiente atât pentru client și server, cât și pentru accesul la bazele de date;
* Asigurarea portabilității pe sisteme de operare bazate pe Windows;
* Datele utilizatorilor vor fi securizate și vor putea fi accesate doar pe baza credențialelor. La introducerea informațiilor în baza de date, acestea vor fi criptate;
* Aplicația nu va afișa niciodată parola utilizatorului, caracterele fiind înlocuite cu un caracter special;
* Toate acțiunile utilizatorilor, respectiv ale serverului vor fi jurnalizate într-un fisier de tip Logger;
* Se dorește extinderea aplicației și a funcționalităților acesteia în viitor. Astfel, vom utiliza concepte de programare orientate pe obiecte, entitățiile proiectului fiind independente, pentru a ușura procesulele de dezvoltarea ale posibilelor actualizări;
* Timpul de răspuns la orice acțiune să fie sub 10 secunde. (Se încearcă truncherea acestuia la un maxim de 5 secunde);
* Interfața trebuie să fie cât mai intuitivă, iar comenzile cât mai ușor de dat, în eventualitatea în care aplicația este folosită de un utilizator fără cunoștințe în domeniu;
* Datele stocate de aplicație nu vor fi modificate, decât la solicitarea utilizatorului sau a Administratorului;