

**Programarea Calculatoarelor si Limbaje de Programare II**

**Profesor  Coordonator: Studenți:**

**Cosmin Fudulu Florentin              Gurau Andrei Eduard**

**Gabriela Olteanu**

A close up of a sign

AI-generated content may be incorrect.

**Notepad+**

**Profesor  Coordonator: Studenți:**

**Cosmin Fudulu Florentin              Gurau Andrei Eduard**

**Gabriela Olteanu**

**CUPRINS**

[**1 - Prezentare Generala:** 4](#_Toc199408409)

[**2 - Funcționalități principale:** 4](#_Toc199408410)

[**2.1 - Ecranul principal:** 4](#_Toc199408411)

[**2.2 – Crearea si Deschiderea Fisierelor** 5](#_Toc199408412)

[**2.3 – Vizualizarea si Navigarea Continutului** 6](#_Toc199408413)

[**2.4 – Editarea Textului si Formatarea de Baza** 7](#_Toc199408414)

[**2.5 – Salvarea, Inchiderea si Gestionarii Starii Fisierelor** 7](#_Toc199408415)

[**3 - Tehnologii Utilizate** 8](#_Toc199408416)

[**3.1 - Framework-ul Qt** 8](#_Toc199408417)

[**3.2 - Limbajul de Programare C++** 9](#_Toc199408418)

[**3.3 -VS Code IDE** 9](#_Toc199408419)

[**3.4 – Sistemul de gestiune al buildului (qmake)** 10](#_Toc199408420)

[**3.5 – Interactiunea cu sistemul de operare nativ** 10](#_Toc199408421)

[**4 – Explicarea Codului** 11](#_Toc199408422)

[**4.1 - Configurarea si Interfata Temei** 11](#_Toc199408423)

[**4.2 - Fereastra Principală** 12](#_Toc199408424)

[**4.3 - Interacțiunile Utilizatorului** 14](#_Toc199408425)

[**4.4 – Gestionarea fisierelor** 14](#_Toc199408426)

[**5 – Concluzie** 15](#_Toc199408427)

[**6 – Bibliografie** 16](#_Toc199408428)

# **1 - Prezentare Generala:**

Proiectul reprezintă un editor de text multi-platformă, dezvoltat în C++ utilizând framework-ul Qt. Denumit "Notepad+" de către autor, aplicația își propune să ofere o interfață grafică intuitivă și funcționalități esențiale pentru crearea, editarea și gestionarea fișierelor text. Un accent deosebit este pus pe personalizarea experienței utilizatorului prin implementarea unui sistem de teme vizuale (incluzând detectarea automată a temei sistemului de operare și opțiuni predefinite precum Dark Contrast, Dark Gray, Light și Solarized) și ajustarea aspectului, cum ar fi raza marginilor componentelor UI. De asemenea, editorul include funcționalități pentru persistența sesiunii, permițând restaurarea fișierelor deschise la o pornire ulterioară.

# **2 - Funcționalități principale:**

Principala funcție a acestei aplicații este de a facilita crearea și modificarea documentelor text într-un mediu prietenos și personalizabil. Editorul oferă unelte esențiale pentru manipularea fișierelor (nou, deschidere, salvare, închidere), editarea textului (tăiere, copiere, lipire, căutare/înlocuire), și adaptarea interfeței la preferințele utilizatorului prin diverse teme vizuale și setări de aspect. De asemenea, gestionează sesiunile de lucru, permițând reluarea rapidă a activității prin restaurarea fișierelor deschise anterior.

## **2.1 - Ecranul principal:**

Ecranul principal este conceput pentru acces direct la funcțiile de editare și vizualizare a conținutului. În partea superioară se găsesc meniul principal și o bară de instrumente pentru acțiuni rapide. Zona centrală este dedicată editării textului, cu fișierele organizate în tab-uri pentru navigare ușoară și un afișaj al numerelor de linie. Partea inferioară conține o bară de stare ce oferă informații contextuale despre fișierul activ și poziția cursorului.

A screenshot of a computer

AI-generated content may be incorrect.

Fig 1 – Ecranul principal

## **2.2 – Crearea si Deschiderea Fisierelor**

Utilizatorul poate iniția lucrul în editor fie prin crearea unui nou fișier text gol, fie prin deschiderea unui fișier existent de pe disc. La crearea unui nou fișier, prin buton , dublu click pe bara de taburi , sau pe butonul din meniul dropdown File, un tab nou, marcat ca "New File", este adăugat în interfață, gata pentru introducerea conținutului. Deschiderea unui fișier existent se realizează printr-un dialog standard de selecție, iar conținutul acestuia este încărcat într-un nou tab, numele fișierului fiind afișat corespunzător. Aplicația suportă și restaurarea sesiunilor anterioare, redeschizând automat fișierele care erau active la ultima închidere.

A screenshot of a computer

AI-generated content may be incorrect.

Fig 2 – Butoanele pentru crearea sau deschiderea unui fisier

A screenshot of a computer

AI-generated content may be incorrect.

Fig 2.1 –Ecranul de tip dialog pentru selectarea fisierului care trebuie deschis.

## **2.3 – Vizualizarea si Navigarea Continutului**

După ce un fișier este deschis sau creat, conținutul său este afișat în zona centrală de editare. Această zonă este echipată cu un sistem de numere de linie în partea stângă, pentru o mai bună orientare în document, iar linia curentă pe care se află cursorul este evidențiată distinct. Bara de stare din partea inferioară a ferestrei oferă informații contextuale suplimentare, precum numărul total de linii și caractere, poziția exactă a cursorului și tipul de fișier. Navigarea între multiple fișiere deschise se face simplu, prin selectarea tab-ului corespunzător.

A screenshot of a computer

AI-generated content may be incorrect.

Fig 3 – Partea principala a aplicatiei.

A screen shot of a computer

AI-generated content may be incorrect.

Fig 3.1 - Un GUI simplu dar eficient pentru schimbarea

tabului la fisierul dorit

## **2.4 – Editarea Textului si Formatarea de Baza**

Funcționalitatea de bază a editorului este modificarea conținutului text. Utilizatorii pot tasta liber, șterge, și utiliza operațiuni standard precum tăiere, copiere și lipire. Pentru modificări mai ample, sunt disponibile funcții de "Undo" (anulare) și "Redo" (refacere). Editorul oferă și unelte pentru căutarea și înlocuirea secvențelor de text în documentul activ. De asemenea, utilizatorul poate schimba modul de editare între inserare (INS) și suprascriere (OVR) și poate selecta tipul de sfârșit de linie (EOL) dorit (Windows CRLF, Unix LF, Mac CR), vizibil și configurabil din bara de stare.

A screenshot of a computer screen

AI-generated content may be incorrect.

Fig 4 – Optiuni ale editorului in ordine : New File , Open File , Save File,

Save All Files , Close File(inchide fisierul/tabul deschis) , Close All Files,

Print File , Cut , Copy , Paste , Undo , Redo , Search , Replace , Settings



Fig 4.1 – Status bar-ul din partea de jos a aplicatiei

Ofera informatii despre tabul curent – Ce tip de fisier este , lungimea lui

tipul de character encoding , de end-of-line characters , si un button pentru

a trece de la modul de insert la modul de overwrite

## **2.5 – Salvarea, Inchiderea si Gestionarii Starii Fisierelor**

Modificările aduse unui fișier pot fi salvate pe disc folosind opțiunile "Save" (pentru fișierele existente) sau "Save As" (pentru a specifica un nou nume/locație sau pentru prima salvare a unui fișier nou). Există și o opțiune de "Save All" pentru a salva toate fișierele modificate deschise. Starea unui fișier (modificat sau salvat) este indicată vizual în tab-ul său (de exemplu, cu un marcaj colorat 🔴(rosu) pentru modificat, 🔵(albastru) pentru salvat). Fișierele pot fi închise individual sau toate odată, editorul avertizând utilizatorul dacă există modificări nesalvate. Suplimentar, conținutul fișierului curent poate fi trimis către o imprimantă

A screenshot of a computer

AI-generated content may be incorrect.

Fig 5 –Aplicatia avertizeaza utlizatorul sa salveze modificarile inainte de a inchide fisierul./aplicatia.

# **3 - Tehnologii Utilizate**

Aplicatie de gestionare a utilizatorilor dintr-o baza de date

* Framework-ul Qt: Folosit pentru a crea o interfață grafică cross-platform pentru sistemul de gestionare a utilizatorilor, permițând o tabelă responsivă și interacțiuni bazate pe dialoguri pe Windows, macOS și Linux.
* C++: Limbajul de bază pentru implementarea logicii aplicației, asigurând performanță ridicată pentru operațiunile cu baza de date și actualizările interfeței.
* Qt Creator IDE: Servește ca mediu de dezvoltare, simplificând designul interfeței, codarea și depanarea pentru proiect.

## **3.1 - Framework-ul Qt**

Framework-ul Qt stă la baza interfeței grafice a editorului de text, oferind un set de unelte puternic și flexibil pentru construirea aplicațiilor desktop cross-platform. Acesta permite crearea unei ferestre principale bogate în funcționalități, cu o interfață cu taburi (folosind QTabBar și QStackedWidget), o bară de unelte personalizabilă (QToolBar) și o bară de stare (QStatusBar) care afișează metadate despre fișiere, cum ar fi numărul de linii și caractere. Sistemul de widgeturi al Qt, inclusiv QTextEdit și QDialog, asigură o experiență de utilizare intuitivă și receptivă, în timp ce mecanismul de semnale și sloturi leagă fără probleme acțiunile utilizatorului — cum ar fi clicurile pe butoane sau schimbările de taburi — de funcționalitățile de bază, precum operațiile pe fișiere și gestionarea setărilor. Natura cross-platform a framework-ului permite rularea consecventă a aplicației pe Windows, macOS și Linux, cu detecția temei și stilizarea dinamică adaptând interfața la mediul nativ al sistemului.

## **3.2 - Limbajul de Programare C++**

C++ este limbajul principal de programare folosit pentru editorul de text, valorificând performanța ridicată și controlul precis pe care îl oferă pentru a gestiona atât logica aplicației, cât și integrarea acesteia cu Qt. Limbajul se ocupă de funcționalitățile esențiale, precum operațiile de intrare/ieșire (de exemplu, salvarea și încărcarea fișierelor), manipularea textului și actualizările interfeței grafice, toate încapsulate în clase precum LineNumberTextEdit și FileTabBar. Designul orientat pe obiecte al C++ organizează codul în componente modulare, facilitând mentenanța și scalabilitatea. Eficiența sa asigură gestionarea fluentă a fișierelor de text mari și actualizări în timp real ale interfeței, cum ar fi urmărirea poziției cursorului și randarea numerelor de linie, ceea ce face din C++ o alegere ideală pentru o aplicație de editare de text robustă și receptivă.

## **3.3 -VS Code IDE**

Visual Studio Code (VS Code) a fost utilizat ca principal mediu de dezvoltare integrat (IDE) pentru realizarea editorului de text, oferind o platformă ușoară, dar puternică, pentru scrierea și depanarea codului C++. Suportul său extins pentru extensii dedicate C++, precum extensia C/C++ dezvoltată de Microsoft, pune la dispoziție funcționalități precum IntelliSense pentru completarea automată a codului, depanare integrată și evidențiere sintactică, facilitând semnificativ procesul de dezvoltare. Spațiul de lucru personalizabil și terminalul integrat din VS Code permit o interacțiune fluentă cu sistemul de build *qmake*, asigurând o compilare și testare eficientă a aplicației bazate pe Qt. Suportul IDE-ului pentru integrarea cu sisteme de control al versiunilor, precum Git, simplifică și mai mult gestionarea codului și colaborarea pe parcursul dezvoltării.

A computer screen shot of a program

AI-generated content may be incorrect.

Fig 7 – Interfata VS Code IDE.

## **3.4 – Sistemul de gestiune al buildului (qmake)**

Sistemul de build **qmake**, parte a ecosistemului Qt, este utilizat pentru a gestiona compilarea și legarea proiectului editorului de text. Acesta simplifică procesul de construire prin generarea de **Makefile-uri specifice platformei** plecând de la un singur fișier de proiect .pro, asigurând astfel compatibilitate cross-platform între Windows, macOS și Linux. qmake gestionează automat dependențele, cum ar fi bibliotecile Qt și resursele personalizate (de exemplu, pictogramele), și include automat fișierele header și sursă necesare. Opțiunile sale de configurare permit un control detaliat asupra setărilor de compilare, cum ar fi activarea modurilor *debug* sau *release*, făcând din qmake un instrument eficient pentru compilarea codului C++ și generarea unui executabil fiabil pentru aplicație.

A screen shot of a computer

AI-generated content may be incorrect.

Fig 8 –Fisierul Makefile.

## **3.5 – Interactiunea cu sistemul de operare nativ**

Editorul de text interacționează cu sistemul de operare nativ pentru a oferi o experiență fluentă, adaptată fiecărei platforme. Prin intermediul claselor **QSettings** și **QProcess** din Qt, aplicația detectează tema sistemului (de exemplu, modul light sau dark) pe Windows, macOS și Linux, aplicând dinamic foi de stil (stylesheets) corespunzătoare pentru un aspect și comportament cât mai apropiat de cele native. Operațiile cu fișiere respectă convențiile specifice fiecărei platforme, cum ar fi utilizarea separatorilor de cale în stil Windows (\) sau a caracterelor de sfârșit de linie Unix (CRLF, LF sau CR), configurabile direct din bara de stare.

Aplicația folosește, de asemenea, API-uri specifice sistemului, cum ar fi **registrul Windows** pentru detectarea temei sau comanda **gsettings** pe Linux pentru interogarea temei GNOME, asigurând astfel o integrare strânsă cu mediul de operare gazdă, păstrând în același timp o coerență cross-platform.

# **4 – Explicarea Codului**

## **4.1 - Configurarea si Interfata Temei**

Framework-ul Qt facilitează configurarea unei interfețe grafice adaptabile, care detectează automat tema sistemului de operare (întunecată sau luminoasă) folosind funcția detectSystemTheme(). Aceasta interoghează setările sistemului (de exemplu, registrul Windows sau gsettings pe Linux) pentru a aplica foi de stil dinamice, definite în funcția createStyleSheet(), care personalizează culorile și aspectul widget-urilor precum QTextEdit, QTabBar și QPushButton. Temele predefinite (DarkContrast, DarkGray, Light, Solarized) oferă utilizatorului opțiuni de personalizare, iar funcționalitatea de ajustare a razelor de colț (borderRadius) permite un control detaliat al esteticii interfeței, asigurând o experiență vizuală consistentă și plăcută pe toate platformele suportate.

A screen shot of a computer program

AI-generated content may be incorrect.

Fig. 9 –Functie care detecteaza daca sistemul de operare este pe o tema intunecata

sau deschisa.

A computer screen shot of text

AI-generated content may be incorrect.

Fig 9.1 – Dropdown din setari care asigura schimbarea temei

A screenshot of a computer program

AI-generated content may be incorrect.

Fig 9.2 – Exemplu din stylesheet

## **4.2 - Fereastra Principală**

Fereastra principală a editorului, construită cu QMainWindow, servește ca nucleu al aplicației, integrând o bară de meniu, o bară de unelte și o zonă de editare cu file gestionată prin FileTabBar și QStackedWidget. Clasa LineNumberTextEdit, o extensie a QTextEdit, afișează numerele de linie și evidențiază linia curentă, îmbunătățind experiența de editare. Bara de stare (QStatusBar) afișează informații esențiale, cum ar fi tipul fișierului, lungimea textului și poziția cursorului, în timp ce bara de unelte oferă acces rapid la funcții precum salvare, deschidere și formatare. Această structură modulară, combinată cu mecanismul de semnale și sloturi al Qt, asigură o interfață fluidă și interactivă, adaptată nevoilor utilizatorului.

A computer screen shot of a program code

AI-generated content may be incorrect.

Fig. 10 – Exemplu de cod.

Acest cod initializeaza aplicatia , fereastra principala , si meniul aplicatiei.

## **4.3 - Interacțiunile Utilizatorului**

Interacțiunile utilizatorului sunt gestionate prin butoane, meniuri și comenzi rapide, conectate la funcționalități prin mecanismul de semnale și sloturi al Qt. De exemplu, butoanele din bara de unelte (newButton, saveButton, findButton) declanșează acțiuni precum crearea unui nou fișier, salvarea conținutului sau deschiderea dialogului de căutare. Dialogurile precum cel de setări (settingsDialog) și cel de căutare/înlocuire permit personalizarea temei și manipularea textului, în timp ce filele din FileTabBar suportă comutarea rapidă între documente și închiderea acestora cu confirmare pentru salvarea modificărilor. Aceste interacțiuni sunt intuitive, cu indicatori vizuali (🔵 pentru fișiere nesalvate, 🔴 pentru modificări), oferind o experiență de utilizare eficientă și clară.

## **4.4 – Gestionarea fisierelor**

Gestionarea fișierelor este o componentă centrală a editorului, implementată prin funcții precum saveSession() și restoreSession(), care salvează și restaurează starea sesiunii utilizând QSettings. Operațiunile de deschidere, salvare și salvare ca (QFileDialog) suportă o gamă largă de extensii de fișiere, mapate în extensionMap și fileTypeFilterMap, de la .txt la .cpp și .py. Funcția detectFileType() identifică automat tipul fișierului pentru a seta formatul corect în bara de stare, iar manipularea conținutului se face eficient cu QFile și QTextStream, asigurând compatibilitatea cu formatele de sfârșit de linie (CRLF, LF, CR) specifice fiecărui sistem de operare. Aceste mecanisme garantează o gestionare robustă și flexibilă a fișierelor.

A screen shot of a computer program

AI-generated content may be incorrect.

Fig. 11 – Exemplu de cod (o portiune din extensiile de fisiere find adaugate unei liste

# **5 – Concluzie**

Editorul de text dezvoltat în Qt și C++ reprezintă o soluție robustă și versatilă pentru editarea fișierelor text, combinând o interfață grafică intuitivă cu funcționalități avansate, adaptate nevoilor utilizatorilor. Prin integrarea perfectă cu Qt, aplicația oferă suport cross-platform, personalizare estetică prin teme dinamice și o gestionare eficientă a fișierelor, cu suport pentru multiple formate și sesiuni persistente. Utilizarea VS Code și qmake a simplificat procesul de dezvoltare, asigurând un cod modular și ușor de întreținut, în timp ce interacțiunile native cu sistemul de operare garantează o experiență fluidă pe Windows, macOS și Linux. Acest proiect demonstrează capacitatea de a construi un instrument practic și personalizabil, care echilibrează performanța și accesibilitatea, fiind potrivit atât pentru utilizatori obișnuiți, cât și pentru dezvoltatori.

Daca ar fi sa continui proiectul as vrea sa adaug optiunea sa il transform intr-un IDE , prin adaugarea unui terminal (CMD / POWERSHELL) si o portiune care sa imi arate sub forma unui tree folderul pe care l-am deschis.

# **6 – Bibliografie**

1. <https://www.qt.io/>

2. <https://doc.qt.io/qt-6/>

3. <https://isocpp.org/>

4. <https://en.cppreference.com/w/>

5. <https://www.qt.io/product/development-tools>

6. <https://doc.qt.io/qtcreator/>

7. <https://cmake.org/documentation>

8. <https://github.com/qt>

9. <https://chatgpt.com>