

This is the ARMv8 Architecture Reference Manual

[ClipboardAndChat]

[12/01/2026]

Îndrumător:

dr. ing. Daniel Morariu

Student:

Szanto Marius Cristian

(grupa: 23/1)

Istoric Versiuni

Data	Versiune	Descriere	Autor
16/11/2025	0.1	Am realizat totul legat de GUI, am creat primele 2 clase in ServerSide, Message si MessageManager.	Szanto Marius Cristian
24/11/2025	0.1	Am realizat reseaua deoarece am considerat ca este unul dintre cele mai importante lucruri in acest proiect. Am dezvoltat mai mult clasele din server. Si am realizat afisarea logs-urilor pe memo din server cand client face o actiune.	Szanto Marius Cristian
27/11/2025	0.2	Am creat CheckBox in Client pentru a salva primele mesaje. Mesajele se salveaza in mod default in fisierul proiectului. Insa am intampinat probleme.	Szanto Marius Cristian
30/11/2025	0.2	Am aflat mai multe despre ServerClient. Fiecare client este recunoscut de catre server prin handle-ul lui propriu. Asadar am creat un map pentru a stoca aceste handle-uri. In logs acum fiecare client poate fi identificat iar in ClientSide fiecare client isi poate identifica mesajul dupa forma „Client000: mesaj”.	Szanto Marius Cristian

5/12/2025	0.3	CheckBox se trimite deodata cu mesajul spre server ptr interpretare.	Szanto Marius Cristian
14/12/2025	0.4	Serverul nu doar ca citeste mesajul ci si stie sa interpreteze starea mesajului, salvat sau nu, dupa modelul „Client000: mesaj0” unde 1 sau 0 reprezinta starea checkboxului.	Szanto Marius Cristian
19/12/2025	0.5	Pentru a identifica si mai bine fiecare user s-a realizat partea de interfata ce tine de username. Am creat butonul, Tedit-ul unde va fi introdus user-ul si un label sugestiv pentru utilizator. Odata setat username-ul serverul schimba numele din „Client000” in orice a setat userul.	Szanto Marius Cristian
20/12/2025	0.6	Pentru a putea fi tracked fiecare user am realizat un mod prin care fiecare user sa nu isi mai poata schimba userul odata conectati. Fiecarui handle ii se asociaza username-ul trimis, serverul stocheaza fiecare user.	Szanto Marius Cristian
26/12/2025	0.7	Am creat butonul Themes, pentru a ii oferi userului customizare.	Szanto Marius Cristian

27/12/2025	0.7	Am realizat clasa UserSettings si interfata ThemeForm.	Szanto Marius Cristian
28/12/2025	0.8	Am realizat un folder cu un txt file unde se salveaza fiecare cod culoare pentru fiecare theme.	Szanto Marius Cristian
30/12/2025	0.9	Am creat butoanele legate de styling	Szanto MariusCristian
3/01/2026	1.0	Am dezvoltat clasa message si i-am adaugat moduri prin care sa interpreteze styling-uri.	Szanto Marius Cristian
8/01/2026	1.1	Am dezvoltat partea de save messages. Serverul are un folder de backup, creat la detectarea primului client unde tine minte tot chat log-ul. Am adaugat butonul Presets, Interfata Presets si butonul Copy to Clip	Szanto Marius Cristian
9/01/2026	1.2	Am setat o imagine pentru butonul de themes si am rezolvat cateva bug-uri legate de styling si presets.	Szanto Marius Cristian

Am inceput proiectul prin realizarea serverului si a clientului deoarece am considerat ca acest aspect este foarte important. In fiecare moment in care client-ul trimite un mesaj, mesajul este inregistrat in server logs. In server logs de asemenea putem observa daca mesajul a fost salvat sau nu, timpul la care a fost trimis, ce colorInt are mesajul, daca e bolduit, cu underline sau Italic.

Aplicatia contine urmatoarele fisiere de care se foloseste(excluzand fisierul Clase): -Client: UserSettings,UserPresets,SavedMessages(ca si backup in caz ca da fail locul in care user-ul isi seteaza txt file-ul pentru save message),Imagini(care sunt incarcate si ca resursa)

-Server: Documents/ClipboardBackup unde se afla backup.txt, serverul salveaza fiecare mesaj pe care il primeste.

Cuprins

ISTORIC VERSIUNI	2
CUPRINS	5
1 SPECIFICAREA CERINTELOR SOFTWARE	6
1.1 Introducere	6
1.1.1 Obiective	6
1.1.2 Definiții, Acronime și Abrevieri	6
1.1.3 Tehnologiile utilizate	7
1.2 Cerințe specifice	7
2 METODA INTERPRETEAZAMESAJ SI REGISTERUSER ERROR! BOOKMARK NOT DEFINED.	
2.1 Descriere	8
2.2 Fluxul de evenimente	8
2.2.1 Fluxul de bază - Trimiterea unui mesaj,salvarea si customizarea lui	Error!
Bookmark not defined.	

2.2.2	Fluxuri	alternative	8
2.2.3	Pre-condiții		8
2.2.4	Post-condiții		8
3	IMPLEMENTARE		9
3.1	Diagrama de clase		9
3.2	Descriere detaliată		10
4	BIBLIOGRAFIE		12

1 Specificarea cerințelor software

1.1 Introducere

O aplicatie c++ care permite chatul intre useri precum si salvarea mesajelor pe calculatoarele lor proprii si customizarea mesajelor trimise pentru realizarea textelor.

1.1.1 Obiective

- 1) Posibilitatea de a trimite mesaje customizabile.
 - 2) Posibilitatea de a salva mesajele trimise.
 - 3) Posibilitatea de a edita mesajele si de a le trimite.
 - 4) Posibilitatea de a sterge mesaje.
 - 5) Posibilitatea de a lucra cu userii la texte mai mari prin preset-uri.
 - 6) Posibilitatea de a converti tot chatul din clipboard intr-un fisier word cu tot cu formatarea prezenta in clipboard.
- 3) Obiectivul acesta a fost indeplinit in mod partial, n-am reusit sa implementez un buton care sa incarce textul din preset in clipboard-ul principal.
- 4)Obiectivul nu a fost indeplinit din cauza problemelor Server-Client, am obtinut multe erori legate de accesari de adresa asadar am renuntat la aceasta functionalitate.
- 5)Obiectiv indeplinit partial.
- 6)Obiectiv esuat.

1.1.2 Definiții, Acronime și Abrevieri

Am aplicat cateva reguli folosite de domnul profesor, Daniel Morariu pentru a nota anumite variabile dar de asemenea am folosit si reguli proprii: **btn** – Reprezinta abrevierea cuvantului button.

ServerSide – Reprezinta partea de Server a proiectului.

ClientSide – Reprezinta partea de Client a proiectului.

get – Abreviere folosita de obicei la metode pentru a sugera faptul ca se obtine o valoare/string de la un membru protected/private a unei clase.

set – Abreviere folosita de obicei la metode pentru a sugera faptul ca se seteaza o valoare/string la un membru protected/private a unei clase.

Server Logs – Reprezinta locul in care sunt salvate fiecare actiuni a fiecarui client.

De asemenea am alternat intre limba engleza si limba romana pentru declararea variabilelor, metodelor in mod neintentionat.

1.1.3 Tehnologiile utilizate

Embarcadero C++ RAD13

Microsoft Word

#include <Vcl.Imaging.pngimage.hpp> - Convertire din png in bmp pentru TSpeedButton.

#include <Vcl.Dialogs.hpp> - Pentru posibilitatea userului de a isi alege folderul cu mesaje salvate.

1.2 Cerințe specifice

InterpreteazaMesaj - principala metoda responsabila pentru procesarea stylurilor si customizarea mesajelor.

MessagesToFile – principala clasa pentru a salva mesajele trimise de client.

MessageManager – principala clasa din server care salveaza toate mesajele trimise de clienti.

CreazaFolder() – metoda prin care se creeaza fisierul din documente de backup. **getBFolderPath()**

– metoda prin care facem rost de path-ul fisierului de backup

2 Metoda InterpreteazaMesaj si Metoda RegisterUser

2.1 Descriere

Metoda este apelata cand se citeste un pachet trimis de server spre client. Aceasta cauta flag-uri dupa care stie ce optiuni de formatare sa aplice, , <U>, <i>, <c>. Culoarea este transmisa in int, asadar este nevoie de transformare in hex.

2.2 Fluxul de evenimente

2.2.1 Customizare

`void ExtractTimestamp(String &text);` //Extrage si elimina Timestamp din mesaj.

`void ExtractUsername(String &text, int colonPos);` //Elimina user din mesaj. `void`

`ExtractColor(String &text);` //Extrage culoarea din mesaj si o aplica.

`void ExtractStyleTags(String &text);` //Extrage alte formatare, bold, italic, underline.

Inregistrare User:

`void RegisterUser(int SocketHandle);` //Face rost de handle-ul unic al fiecarui client si il asociaza username-ului

2.2.2 Fluxuri alternative

2.2.3 Pre-condiții

Pre-conditii:

- Serverul trebuie sa fie pornit.
- Clientul trebuie sa aiba conexiune la retea.
- Utilizatorul trebuie sa introduca un username valid.

2.2.4 Post-condiții

Post-conditii:

- Mesajul este afisat în fereastra de chat alaturi de timpul trimis si de formatarea aleasa de celalalt client.
- Mesajul este transmis catre ceilalti clienti conectati.
- Mesajul poate fi salvat local în fisierul de backup.

3 Implementare

3.1 Diagrama de clase

Message

#bool: Bold,Italic,Underline,isBold,isItalic,isUnderline #Tcolor: messageColor #String username,messageText,timestamp
+bool returnisBold(); +bool returnisItalic(); +bool returnisUnderline(); +TColor getMessageColor(); +String getUsername(); +String getMessageText(); +String getTimestamp(); +void setisBold(bool value); +void setisItalic(bool value); +void setisUnderline(bool value); +void resetValues(); +void ExtractTimestamp(String &text); +void ExtractUsername(String &text, int colonPos); +void ExtractColor(String &text); +void ExtractStyleTags(String &text); +void InterpreteazaMesaj(String receivedText, TRichEdit *richEdit);

UserSettings
-String ColourForm;ColourMemo;ColourEdit;ColourText; -bool CheckThemeFolder=0;
+void SetColourForm(String Color); +void SetColourMemo(String Color); +void SetColourEdit(String Color); +void SetColourText(String Color); +void SetCheckThemeFolder(bool set);

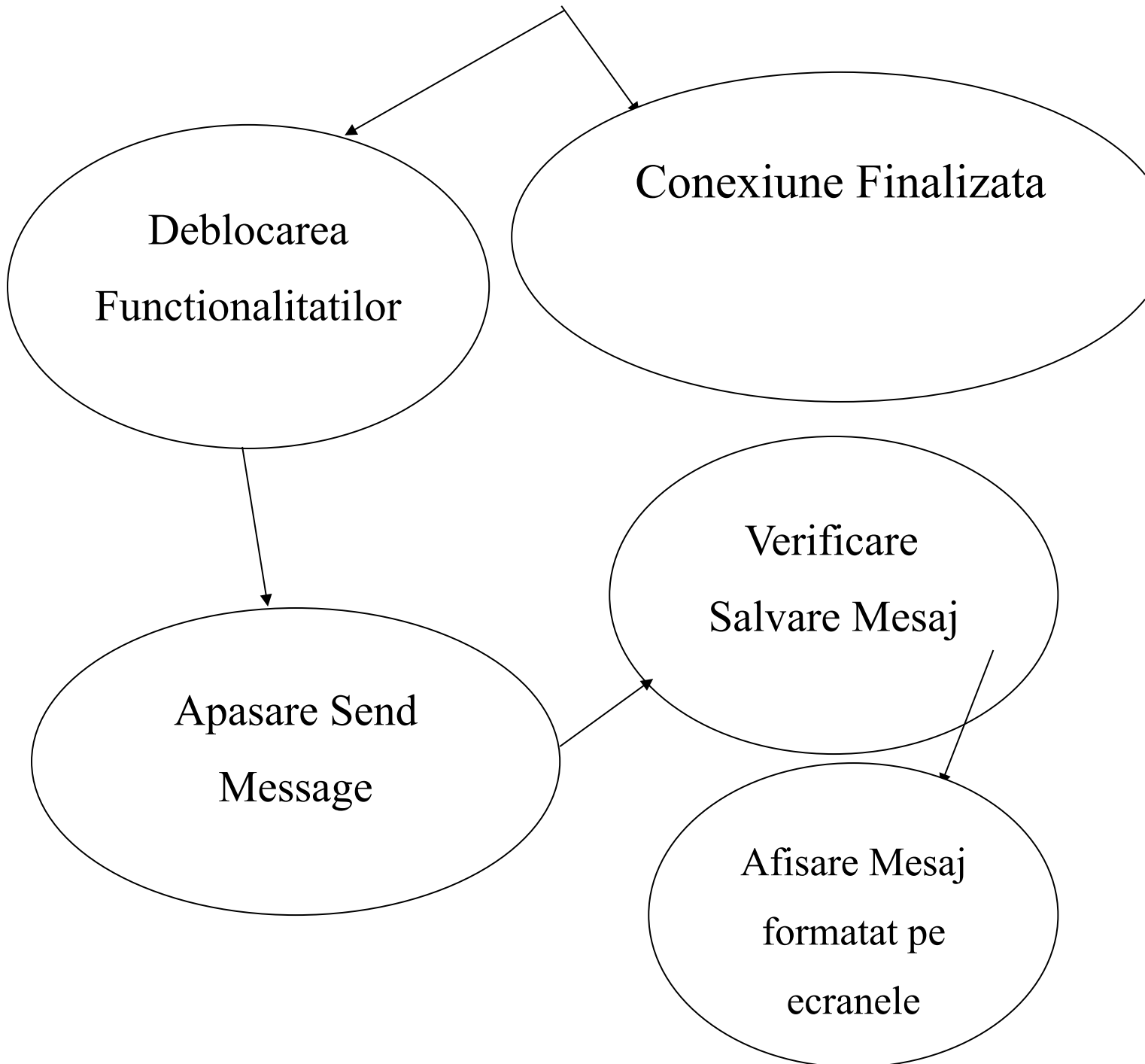
```
+String GetColourForm();  
+String GetColourMemo();  
+String GetColourEdit();  
+String GetColourText();  
+bool GetCheckThemeFolder();
```

MessagesToFile

-UserSettings setari

```
#String Path  
#vector<String>SavedMessages;  
#bool GlobalCheck  
+void AddVector(String text);  
+String returnPath();  
+void setPath(String filePath);  
+void SetGlobalCheck(bool ok);  
+bool GetGlobalCheck();  
+bool GetCheckThemeFolder();  
+void SetCheckThemeFolder(bool set);
```

3.2 Descriere detaliată



4 Bibliografie

Introducere in C++ de Macarie Breazu

1. <https://docwiki.embarcadero.com/>
2. <https://www.w3schools.com/>
3. <https://www.geeksforgeeks.org/>
4. <https://www.embarcadero.com/RADAICompanion>

