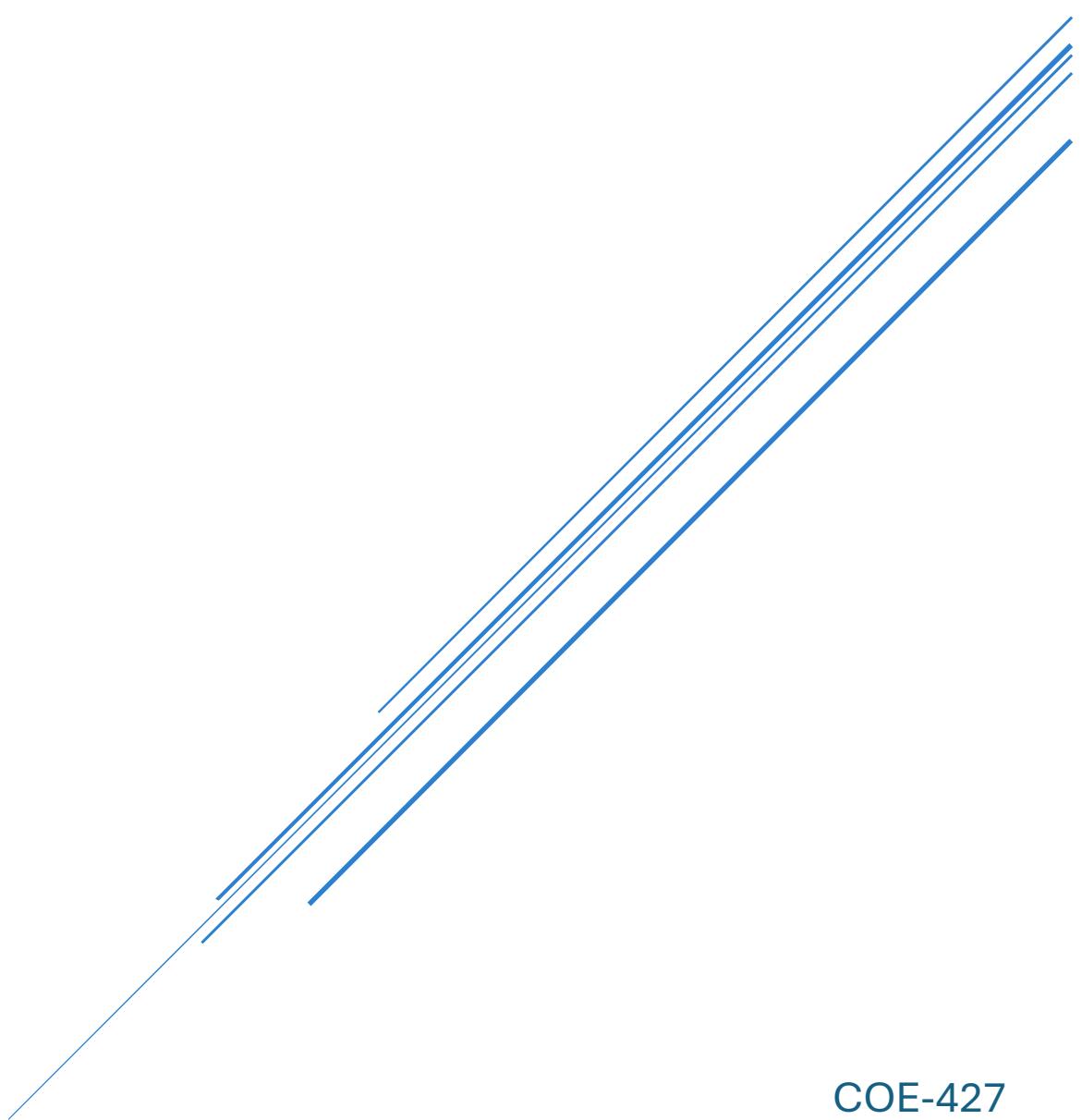


# JALAL ALI ZAINADDIN

202154790



COE-427  
Features Report

## Table of Contents

1)	Message History and Persistence .....	2
2)	What we changed (key files) .....	2
3)	DDS-side implementation (late-join replay) .....	2
4)	Application-side persistence & search.....	2
5)	Problems Faced.....	2
6)	Output screenshots implementation.....	3

## 1) Message History and Persistence

I added durable chat history so users can receive recent messages when they join late.

## 2) What we changed (key files)

chat\_qos.xml: defined a dedicated profile for chat messages that enables late-join history (reliable, transient-local, keep-last).

Backend code – writes/reads messages using the above QoS and persists them locally so the UI can replay, and search history on demand through search\_history.py (no UI changes).

RTI backend now logs to chat.db (columns: ts, direction, from\_user, first\_name, last\_name, grp, text).

DDS late-join behavior is still provided via Connex QoS + Persistence Service; the app-level log is purely for search and auditing.

## 3) DDS-side implementation (late-join replay)

**Reliability:** set RELIABLE so messages are acknowledged and not dropped.

**Durability:** set TRANSIENT\_LOCAL so a writer keeps a cache of the last messages for late joiners.

**History:** KEEP\_LAST (depth 50) so a bounded window of recent messages is retained and delivered immediately to new readers in the same room/partition.

These settings are captured in Chat\_Library/ChatMessage\_Profile of chat\_qos.xml.

## 4) Application-side persistence & search

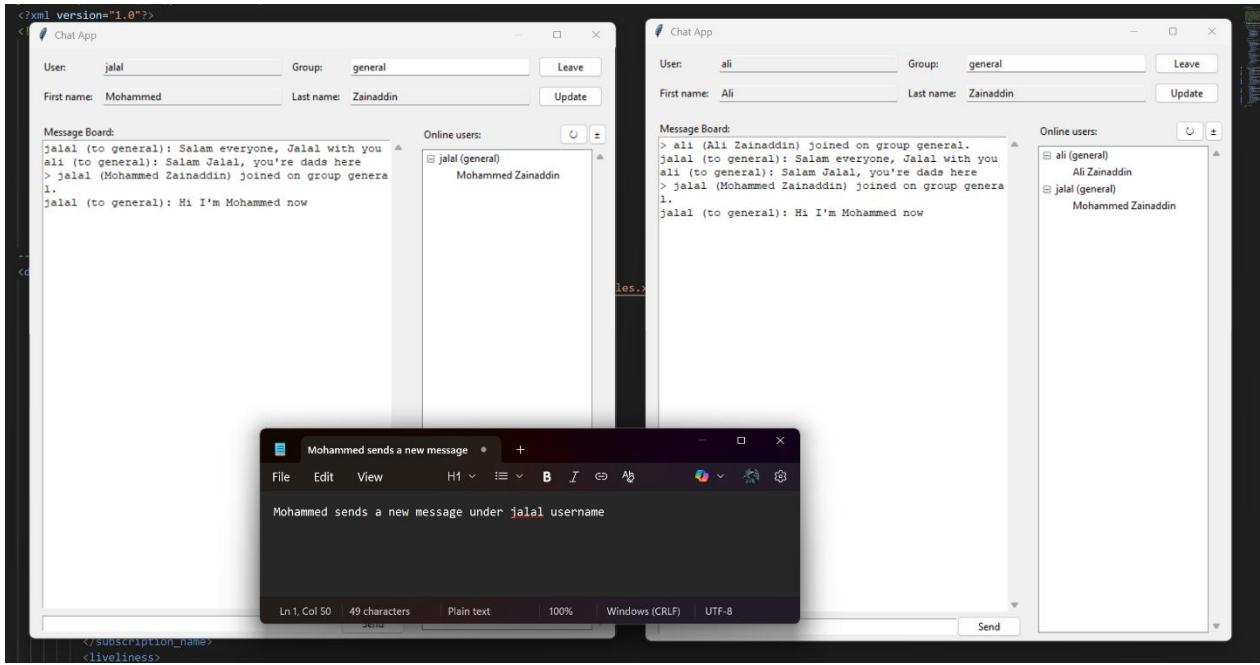
Local persistence: messages are stored so the app can replay recent chat on join and keep a longer trail than the DDS history window.

## 5) Problems Faced

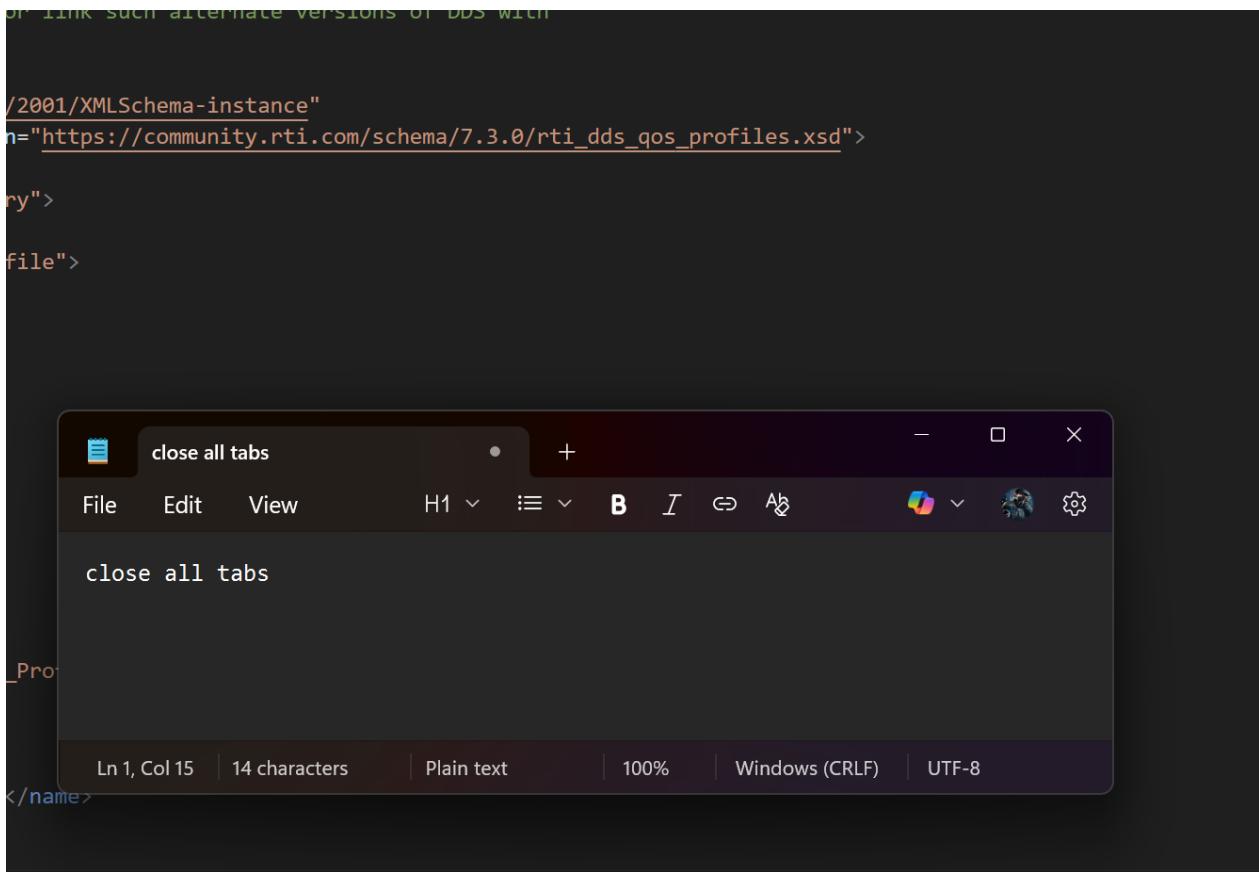
No GUI implementation for the search history of a specific message, only using CLI

Does not store messages using RTI History backend tool/s

## 6) Output screenshots implementation



## Close all GUI tabs to test message history functionality



The screenshot shows a dark-themed code editor window. The main area displays XML schema code:

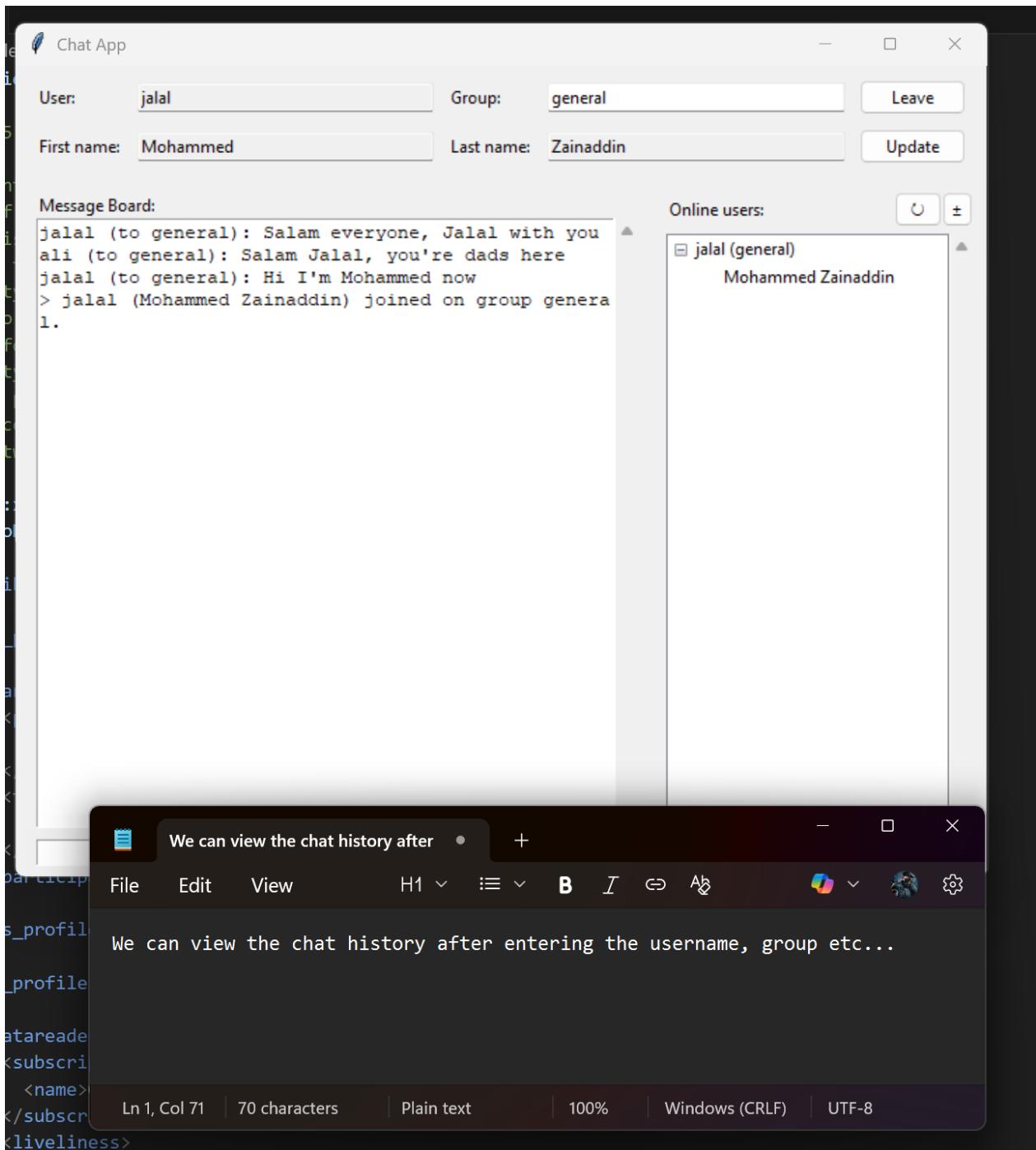
```
OR LINK SUCH ALTERNATE VERSIONS OF DDS WITH  
  
/2001/XMLSchema-instance  
n="https://community.rti.com/schema/7.3.0/rti_dds_qos_profiles.xsd">  
  
ry">  
  
file">
```

A context menu is open at the top left of the editor, with the option "close all tabs" highlighted.

The editor's interface includes:

- File Edit View H1  $\equiv$  **B** *I*  $\Lsh$   $\mathbb{A}$
- Icons for Undo, Redo, Find, Replace, and others.
- Text status bar: Ln 1, Col 15 | 14 characters | Plain text | 100% | Windows (CRLF) | UTF-8

## Run a GUI tab to test message history functionality



## Search for a specific message using search\_history.py to view messages stored on chat.db

```
(rti_venv) PS C:\Users\jazai\OneDrive\Desktop\COE427\Distributed-Computing> python .\COE427_HW1\chat_app\python\search_history.py --group general --query Salam --limit 50 --db chat.db
>>
[out] jalal: {"_type": "chat", "from": "jalal", "to": "general", "group": "general", "text": "Salam everyone, Jalal with you"}
[in] jalal: {"_type": "chat", "from": "jalal", "to": "general", "group": "general", "text": "Salam everyone, Jalal with you"}
[out] ali: {"_type": "chat", "from": "ali", "to": "general", "group": "general", "text": "Salam Jalal, you're dads here"}
[in] ali: {"_type": "chat", "from": "ali", "to": "general", "group": "general", "text": "Salam Jalal, you're dads here"}
```