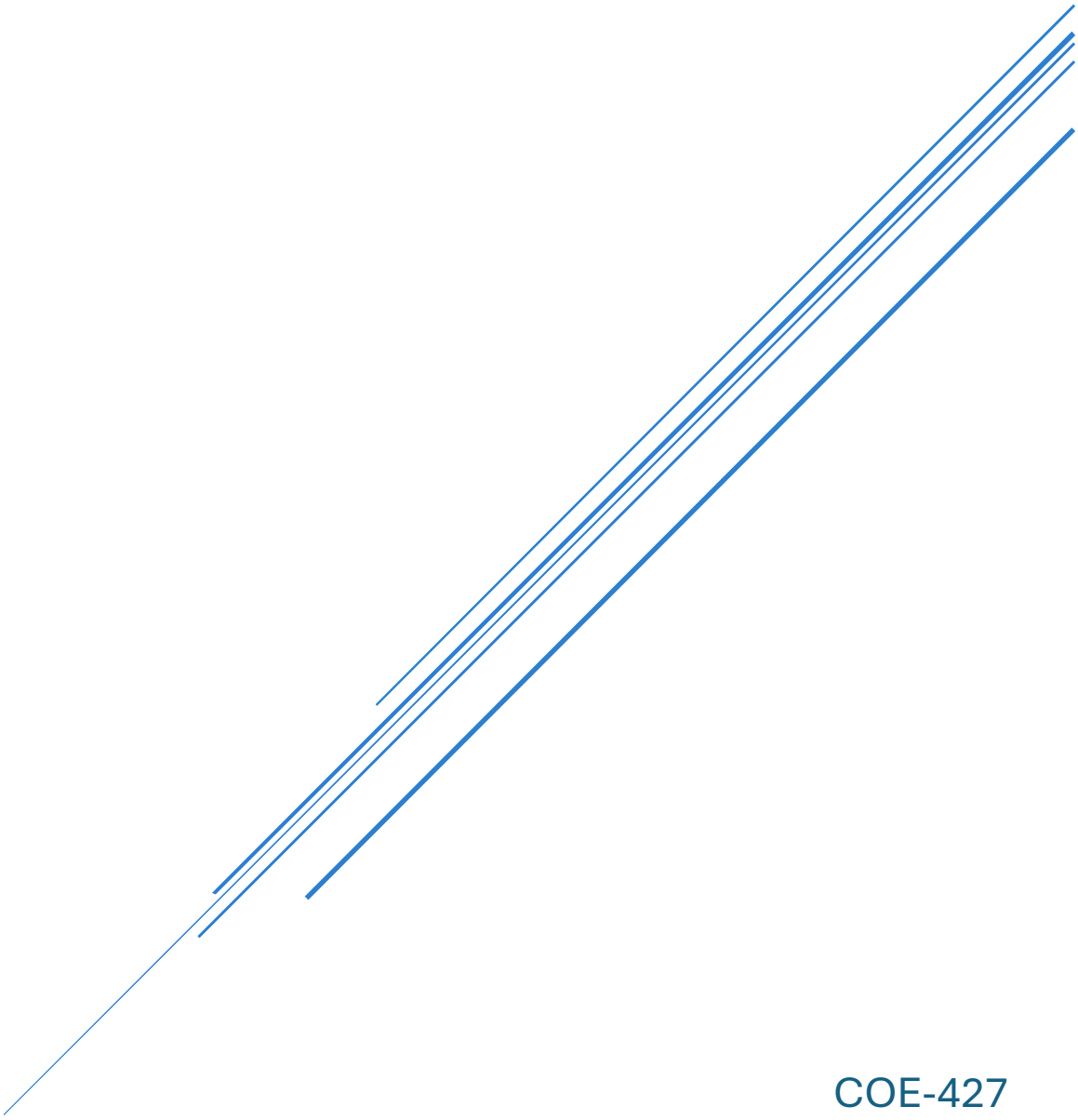
JALAL ALI ZAINADDIN

202154790



COE-427

Features Report

Table of Contents

1) Message History and Persistence

I implemented durable chat history so that users can still see recent messages even after leaving and re-joining the chat – or when a new instance of the application starts.

This is done using RTI Connext DDS QoS + RTI Persistence Service, instead of a local database:

- The message topic (type ChatMessage) now uses:
  - RELIABLE reliability
  - KEEP_ALL history
  - TRANSIENT durability
- A dedicated Persistence Service configuration (ChatPersistenceService.xml) runs on domain 0 and subscribes to all durable topics.
- When a new DataReader for message joins the same domain and group, the Persistence Service replays stored samples so the GUI immediately shows the previous chat messages.

On top of this, I added a search bar directly in the GUI Message Board so users can search through past messages. The search feature highlights all occurrences of the query string and can be cleared with a "Clear" button

2) Multimedia Support

I added full multimedia support to the chat so users can share images and other files (e.g., videos) with interactive display in the GUI.

DDS side

- Defined a new DDS type FileMessage (in chat.py) with fields:
  - fromUser, toUser, toGroup
  - fileName, mimeType
  - data as a bounded sequence<octet> (implemented in Python as Sequence[idl.uint8])
- Created a new topic "file" with the same QoS profile used for text messages (ChatMessage_Profile):
  - RELIABLE, KEEP_ALL, TRANSIENT
- The Persistence Service also subscribes to this topic, so file messages are persisted and can be replayed to late joiners.

GUI side

- Added a "Send file…" button next to the text Send button.

- Clicking “Send file…” opens a file dialog filtered for images and videos first, with an option for all files.
- The selected file is read into memory, packed into a FileMessage, and published.
- On the receiver side:
    - The FileMessage is received, its byte sequence is reconstructed and saved under a downloads/ folder.
    - If the MIME type starts with image/, the GUI:
        - Shows a normal text line (“user sent an image to group: filename”),
        - And displays an inline thumbnail inside the Message Board using Tkinter’s PhotoImage.
    - For other file types (e.g., videos), the GUI shows a clickable blue link with “[open]”; clicking it launches the file using the operating system’s default viewer.

This satisfies the “Multimedia Support” feature with real DDS-based file transfer.

3) What changed (key files)

chat_qos.xml

- Updated Chat_Library::ChatMessage_Profile to use RELIABLE, KEEP_ALL, TRANSIENT for both DataWriter and DataReader so that Persistence Service can store and replay chat messages and files.

ChatPersistenceService.xml

- New configuration file that starts ChatPersistenceService on domain 0, with a persistence_group that matches all durable topics (filter="*"), and subscriber/publisher partitions set to "*" so it can see all groups.

chat.py

- Existing ChatUser and ChatMessage kept.
- Added FileMessage type with bounded Sequence[idl.uint8> for binary payloads.

dds_app.py

- Created a new FileMessage topic ("file"), DataWriter and DataReader using ChatMessage_Profile.
- Added a file_send() API that reads a file from disk, truncates it to the allowed size if necessary, fills FileMessage, and writes it.

- Added a _file_monitor thread that waits on a WaitSet and calls handlers.file_received(samples) when file messages arrive.

app.py

- Connected GUI handlers to DDS: send_file() forwards requests to dds_app.file_send().
- Implemented received_file() to:
    - Convert the DDS sequence of octets back to bytes,
    - Save the file into downloads/,
    - Call gui.file_received() with the sender, destination, file path and MIME type.

gui.py

- Added a **Search bar** above the Message Board with “Search” and “Clear” buttons that highlight matches in the text widget.
- Added a **“Send file…”** button next to the normal Send button, wired to a Tk file dialog.
- Implemented file_received() in GuiApp and helper methods in _GuiWidgets:
    - insert_image(file_path) for inline image thumbnails,
    - insert_file_link(label, file_path) for clickable links that open the file in the OS.

.

4) DDS-side implementation (late-join replay)

Message history:

- ChatMessage and FileMessage topics:
    - Reliability: RELIABLE_RELIABILITY_QOS
    - History: KEEP_ALL_HISTORY_QOS
    - Durability: TRANSIENT_DURABILITY_QOS
- Persistence Service (ChatPersistenceService):
    - Joins domain 0 and creates a persistence_group with filter="*" so it persists all durable topics (both text and files).
    - Uses partition "*" so it matches any chat group/room.
    - Because in-memory TRANSIENT storage is used (no filesystem configured), history is kept while the service is running, which is sufficient for the assignment.

Late-join behavior:

- When a new chat instance joins a room, its DataReader is created with the same QoS and partition.
- Persistence Service detects the new reader and replays stored samples for that reader.

- The message monitor threads (_message_monitor and _file_monitor) immediately deliver these historical samples to the GUI, so the Message Board is populated with previous messages and file notifications

.

5) Application-side persistence & search

History replay:

- When the user clicks **Join**, the app creates the DDS entities and subscribes to the topics.
- Any historical ChatMessage and FileMessage samples replayed by Persistence Service are appended to the Message Board via GuiApp.message_received() and GuiApp.file_received().

Search:

- Above the Message Board there is a **Search** entry and two buttons:
  - **Search**: highlights all case-insensitive matches of the term in the text widget using a search_highlight tag.
  - **Clear**: removes the highlight and clears the search box.
- This allows the user to quickly locate messages in the persisted chat history.

Multimedia:

- User presses **Send file…**:
  - A Tk askopenfilename dialog opens with filters for images and videos.
  - The destination is either the selected user from the Online Users tree or the current group name.
  - The selected file is sent over DDS as a FileMessage.
- When a FileMessage is received:
  - The file is saved into a downloads/ folder.
  - A line is added to the Message Board indicating who sent what and to whom.
  - If it is an image:
    - An inline thumbnail is rendered directly inside the Message Board.
  - For other files:
    - A blue underlined “[open]” link is inserted; clicking it launches the file in the system’s default viewer (e.g., image viewer or video player).

6) Problems Faced

- **It took a long time to figure out the appropriate location for license file**
- **The Windows OS does not launch the program smoothly, so I needed to redo the application on Linux OS**
- **Attempting to model file payloads as bytes in FileMessage caused TypeError: Unsupported member type <class 'bytes'>. The correct mapping is Sequence[idl.uint8] with a bounded sequence, and then manually converting between bytes and lists of integers when sending/receiving.**

These issues were resolved, and the final application now demonstrates both Message History & Persistence and Multimedia Support as required by the assignment.

7) Output screenshots implementation

You can access to the screenshot presentation of these features inside of “Screenshots Presentation” folder.