

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA TRUYỀN THÔNG ĐA PHƯƠNG TIỆN

ĐỀ THI KẾT THÚC HỌC KỲ
HỌC KỲ 2, NĂM HỌC 2023 - 2024

- **Học phần:** Lập Trình Ứng Dụng Mạng – TN414
- **Lớp học phần:** TN41401
- **Thời gian làm bài:** 90 phút
- **Lưu ý:** Sinh viên được phép sử dụng tài liệu giấy
- **Nội dung đề thi:**

Câu 1 (5 điểm): Viết chương trình Simple Chat có cho phép thu hồi tin nhắn

- Sinh viên sử dụng UDPClient để gửi và nhận dữ liệu.
- Thiết kế form như hình 1 với các control sau:
 - **txtIPR:** là TextBox để người dùng nhập IP của máy sẽ gửi tin nhắn đi.
 - **txtPortR:** là TextBox để người dùng nhập Port của máy sẽ gửi tin nhắn đi.
 - **txtPortL:** là TextBox để người dùng nhập Port của máy cục bộ nhận tin nhắn gửi đến.
 - **btnConnect:** là nút **Connect**, khi nhấn vào sẽ kết nối với máy nhận tin nhắn theo cấu hình đã thiết lập.
 - **lstMsg:** là ListBox hiển thị tin nhắn gửi đi và gửi đến.
 - **btnThuHoi:** là nút **Unsent**, khi nhấn vào sẽ gọi phương thức **unsent()**
 - **btnXoa:** là nút **Clear All**, khi nhấn vào sẽ gọi phương thức **clearall()**
 - **txtMsg:** là TextBox để người dùng nhập nội dung tin nhắn gửi đi.
 - **btnGui:** là nút **Send**, khi nhấn vào sẽ thực hiện việc gửi tin nhắn.

* Phương thức **unsent** thực hiện các việc như sau:

- Thay thế tin nhắn **người dùng đang chọn** thành **UNSENT MESSAGE [YOU]**
- Gửi chỉ số tin nhắn sang phía người nhận, tin nhắn có dạng: **UNS: <chỉ số tin nhắn>**
- Thay thế nội dung tin nhắn (theo chỉ số tin nhắn) bên phía nhận thành **UNSENT MESSAGE [FRIEND]**.

* Phương thức **clearall** thực hiện các việc như sau:

- Xóa hết tất cả tin nhắn hai phía người gửi và người nhận
- Gửi lệnh xóa sang phía nhận, tin nhắn có dạng: **CLE:**
- Xóa hết nội dung tin nhắn bên phía nhận

* **Chú ý:**

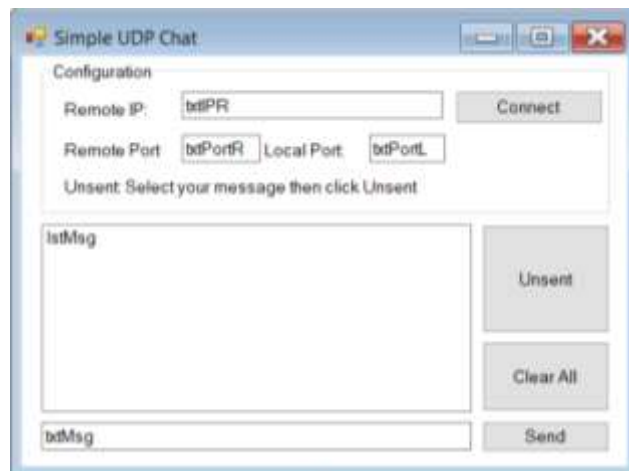
- Tin nhắn gửi đi hiển thị trong listbox là **SEN: <nội dung tin nhắn>**
- Tin nhắn nhận được hiển thị trong listbox là **REC: <nội dung tin nhắn>**
- Người dùng chỉ có thể thu hồi tin nhắn do mình gửi (tin nhắn bắt đầu bằng **SEN** trong listbox)

* **Các phương thức của lớp Listbox:**

- **Listbox.Items:** mảng các phần tử

- **Listbox.Items.Add**: thêm phần tử mới

Listbox.Items[index]=<nội dung mới>: thay thế nội dung của phần tử ở vị trí **index** bằng <nội dung mới>



Hình 1. Giao diện ứng dụng Simple Chat

Câu 2 (5 điểm): Sử dụng các lớp TCP thích hợp, viết ứng dụng quản lý tin nhắn như sau:

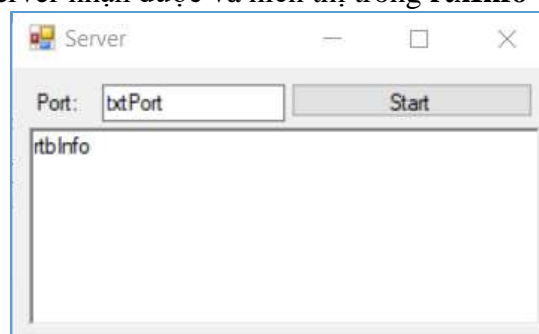
- **Mô tả ứng dụng**:

Ứng dụng cho phép người dùng lưu tin nhắn có mật khẩu trên server mà không cần đăng ký tài khoản. Để đọc tin nhắn, người dùng chỉ cần cung cấp tiêu đề tin nhắn và mật khẩu.

- **Các yêu cầu cài đặt**

- Thiết kế form như hình 2 (1 điểm)
 - **txtPort**: là textbox để nhập số hiệu cổng server sẽ nhận dữ liệu đến
 - **btnStart**: là button, khi nhấn vào sẽ khởi động server theo cấu hình đã thiết lập
 - **rtxInfo**: là richtextbox hiển thị các thông tin trạng thái, các lệnh nhận từ client
- Yêu cầu sử dụng MultiThread để phục vụ cùng lúc nhiều Client (1 điểm)
- Cài đặt các lệnh xử lý như sau: (4 điểm)
 - **CRE#<tiêu đề>#<mật khẩu>#<nội dung tin nhắn>**: Tạo tin nhắn mới và máy chủ sẽ lưu tin nhắn vào tập tin có tên là <tiêu đề>_<mật khẩu>.txt.
 - ✓ Nếu tập tin đã có, ghi đè nội dung cũ.
 - ✓ Nếu lưu thành công, phản hồi là **100. Tin nhan luu thanh cong**
 - ✓ Nếu lưu không thành công, phản hồi là **203. Khong the luu tin nhan.**
 - **REA#<tiêu đề>#<mật khẩu>**: Đọc tin nhắn lưu trong tập tin <tiêu đề>_<mật khẩu>.txt.
 - ✓ Nếu tin nhắn tồn tại, có 2 trường hợp
 - Nếu mật khẩu sai, phản hồi là **204. Sai mat khau.**
 - Nếu mật khẩu đúng, phản hồi là nội dung tin nhắn.
 - ✓ Nếu tin nhắn không tồn tại, phản hồi là **205. Tin nhan khong ton tai.**

* **Chú ý**: Các lệnh server nhận được và hiển thị trong **rtxInfo**



Hình 2. Giao diện Server