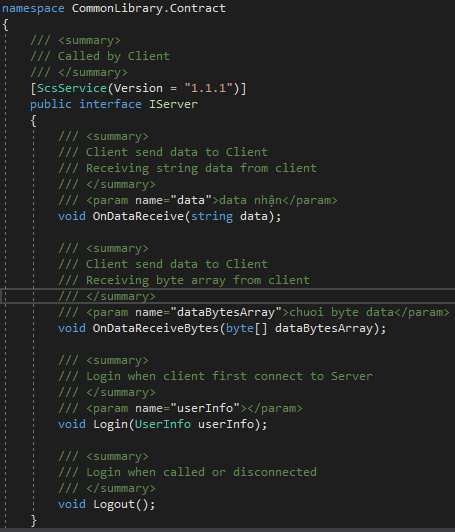
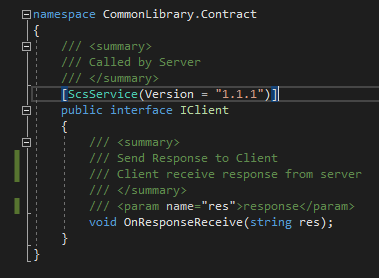
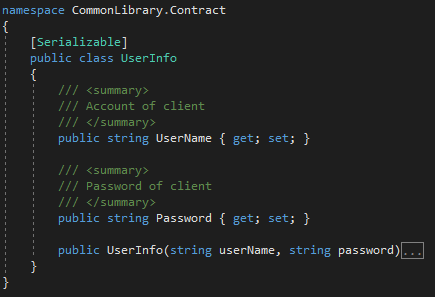
**BÁO CÁO 1**

1. ***Công việc hoàn thành:***
   * Tạo ra một Web API project, khi chạy tạo ra một TCP/Server lắng nghe tại một port (cố định) để các Client kết nối lên.
   * Server dùng một danh sách để quản lý các Client kết nối lên.
   * Server nhận dữ liệu của client gửi lên, kiểm tra CRC và phản hồi lại cho client.
2. ***Cấu trúc chương trình:***Solution gồm 3 project (cài đặt chi tiết xem trong link github đi kèm)
   1. **CommonLibrary** chứa hai Interface và một class
      * IServer: cung cấp các method cho Client gọi tới  
        
      * IClient: cung cấp các method cho Server gọi tới  
        
      * UserInfo: Class chứa thông tin của client kết nối đến Server  
        
   2. **Client** chứa file Implement của IClient và file chương trình chính (chi tiết trong project đi kèm)
      * Tracking\_Client: implement interface Iclient
      * Program: chương trình chính
   3. **TCPServer\_demo**: chứa file Implement interface Iserver và file tạo ra Server (chi tiết trong project đi kèm)
      * ServerApp: tạo ra Server để Listen
      * Tracking\_Server: Implement interface Iserver
3. ***Các Framework được sử dụng:***
   * Scs: TCP Server/Client Communication  
     <https://github.com/hikalkan/scs>
   * Crc32.Net: Tính giá trị CRC-32 của dữ liệu cần gửi, ghi thêm giá trị vào cuối dữ liệu và kiểm tra CRC.  
     <https://github.com/force-net/Crc32.NET>
4. ***Vấn đề chưa xử lý xong:***
   * Việc reconnect của client khi kết nối bị hỏng: sử dụng đối tượng ReConnecter do framwork Scs cung cấp; khi chạy đã kết nối lại được nhưng vẫn còn bug chưa xử lý xong.