



# Chat Application Using WCF

---

Shady Ahmed Mohamed [shady@elyaski.com](mailto:shady@elyaski.com)

1/9/2010



## System Architecture & Design

- Service
  - Interface **IChat**
    - Login, Logoff, Send Message, File, Notification, etc..
    - ChatService : IChat (inheritance)
  - Interface **IChatCallback**
    - Receive Message, File, Notifications, etc..
    - Clients : IChatCallback (inheritance)
- Host
  - Run on server (Singleton Instance)
  - Define EndPoints
  - Expose Metadata
  - Define ServiceThrottlingBehavior
    - MaxConcurrentCalls
    - MaxConcurrentSessions
    - MaxConcurrentInstances
  - Define NetTcpBinding
    - MaxBufferSize
    - MaxReceivedMessageSize
    - TransferMode
    - MaxStringContentLength
- Client
  - Implement IChatCallback functions



## System Features

- + Support user registration
  - + Support adding contacts using their usernames
  - + Allow user to accept or reject adding requests
  - + Show user's contact list showing who's online and who's offline
  - + Support Contact status
  - + Support Contact DP
  - + Update status & DP for online conacts
  - + Allow sending and receiving text message to and from any online contact
  - + 'Enter' to send 'Cntl-Enter' for new line
  - + Toolbar notifications
  - + User is writing..
  - + Added Icons
  - + Added invitations
  - + Support sending emotions within text messages
    - + Selecting an emotion
    - + Auto detect emotions (When writing or copy-paste)
  - + Nudge
    - + Nudge lock for 1 minute
  - + Asynchronous Calls
  - + Send offline msgs
  - + Show offline messages to the user once s/he goes online
  - + Allow sending files (of any type or format) to an online contact (700 MB+)
  - + Support Video conversation (each peer sees the other peer's live camera)
  - + Support Audio conversation (each peer hears the other peer's live voice)
- > [Security Measures] <-
- + Generate a session key (random number) to be assigned to each online user & used to communication with the server
  - + Appending a random number to each contacts' Display Pictures
  - + Generate a random invitation ID foreach sent invitation and store receiver's response along with the invitation ID



## Service Configuration

### 1- Base address

```
ServiceHost host = new ServiceHost(typeof(ChatSvc.ChatSvc),  
new Uri("net.tcp://localhost:8080/Chat"));
```

### 2- ServiceThrottling

```
ServiceThrottlingBehavior throttle;  
throttle =  
host.Description.Behaviors.Find<ServiceThrottlingBehavior>();  
if (throttle == null)  
throttle = new ServiceThrottlingBehavior();  
throttle.MaxConcurrentCalls = 100;  
throttle.MaxConcurrentSessions = 100;  
throttle.MaxConcurrentInstances = 100;  
host.Description.Behaviors.Add(throttle);
```

### 3- NetTcpBinding & Endpoint

```
NetTcpBinding nettcp = new NetTcpBinding(SecurityMode.None);  
nettcp.MaxBufferPoolSize = 67108864; //64 MB  
nettcp.MaxBufferSize = 524288; //512 KB  
nettcp.MaxReceivedMessageSize = 524288; //512 KB  
nettcp.TransferMode = TransferMode.Buffered;  
nettcp.ReaderQuotas.MaxArrayLength = 67108864;  
nettcp.ReaderQuotas.MaxBytesPerRead = 67108864;  
nettcp.ReaderQuotas.MaxStringLength = 268435456;  
host.AddServiceEndpoint(typeof(ChatSvc.IChat), nettcp,  
"");
```

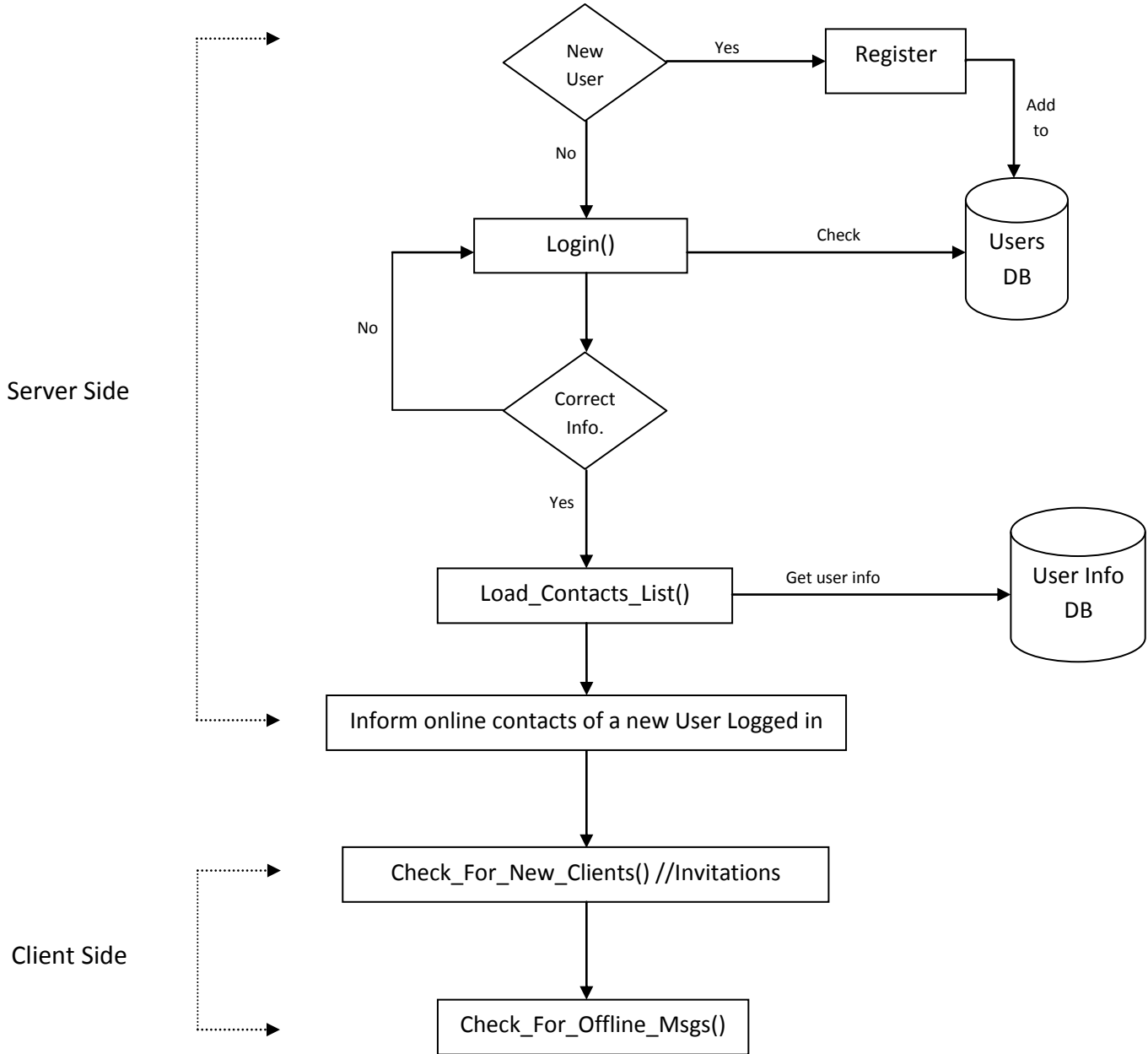
### 4- Metadata Exposing endpoint

```
ServiceMetadataBehavior smb =  
host.Description.Behaviors.Find<ServiceMetadataBehavior>();  
if (smb == null)  
smb = new ServiceMetadataBehavior();  
smb.MetadataExporter.PolicyVersion =  
PolicyVersion.Policy15;  
host.Description.Behaviors.Add(smb);  
host.AddServiceEndpoint(  
ServiceMetadataBehavior.MexContractName,  
MetadataExchangeBindings.CreateMexTcpBinding(),  
"net.tcp://localhost:8081/Chat/Mex"  
);
```



## Basic Operations Sequence Logic

### ✓ User Login



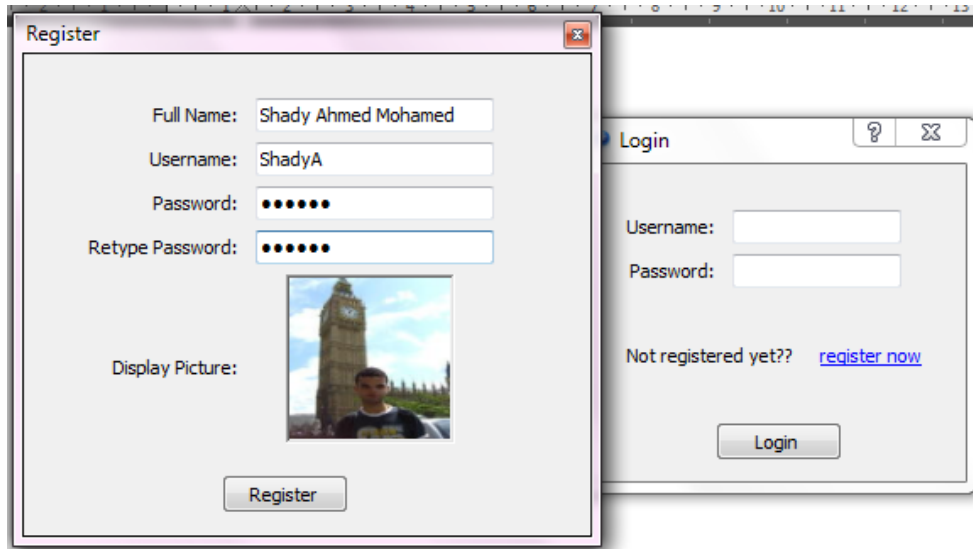


## **Datastore: Dataset Schema (XML)**

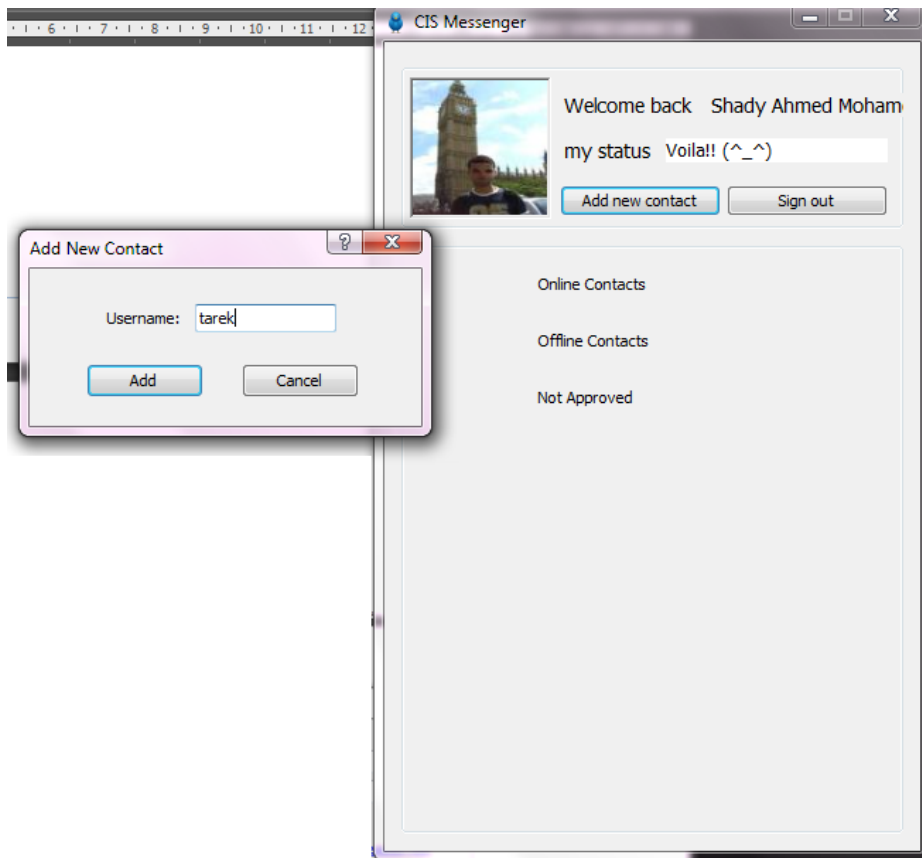
- ❖ Users Table //Includes all registered users
  - Username <PK>
  - Fullname
  - Password
  - Display\_Picture\_Path
  
- ❖ Clients DB //Dataset for each client. Name=Client's username
  - Contacts Table //Includes all contacts related to this client
    - Username
    - Approved //Status variable that shows approved/blocked/new contact.
  - Offline Messages Table //Includes all offline Messages
    - Username
    - Message

## Sample Runs: Screenshots

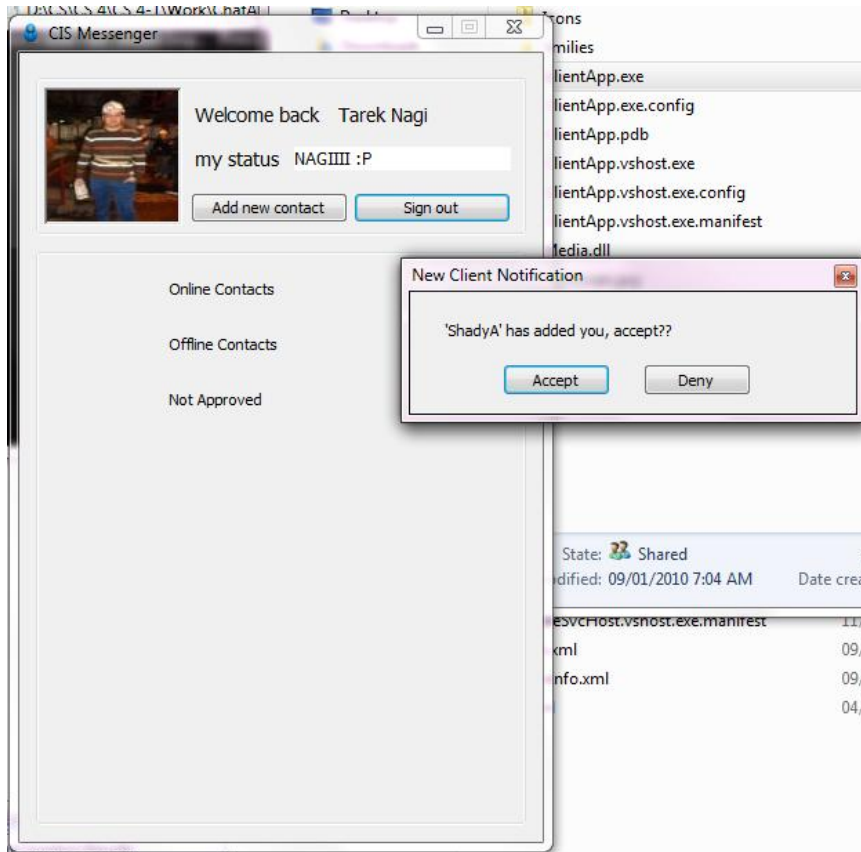
- User Registration:-



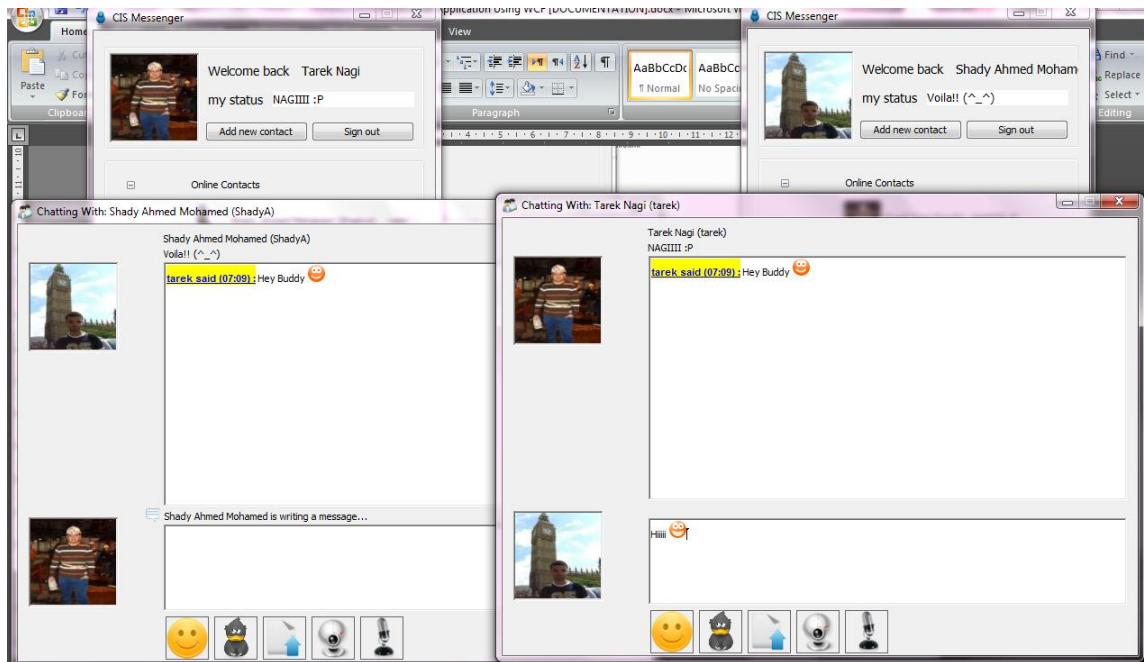
- Main Screen, Status & adding new contact:-



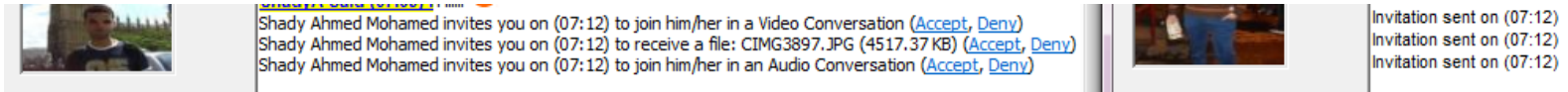
- Contact Confirmation:-



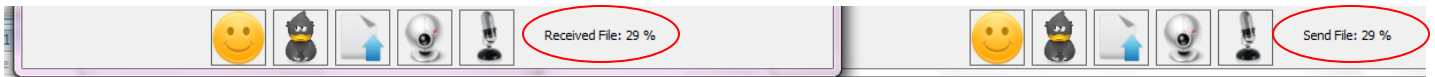
- Chatting, Smilies, User is writing



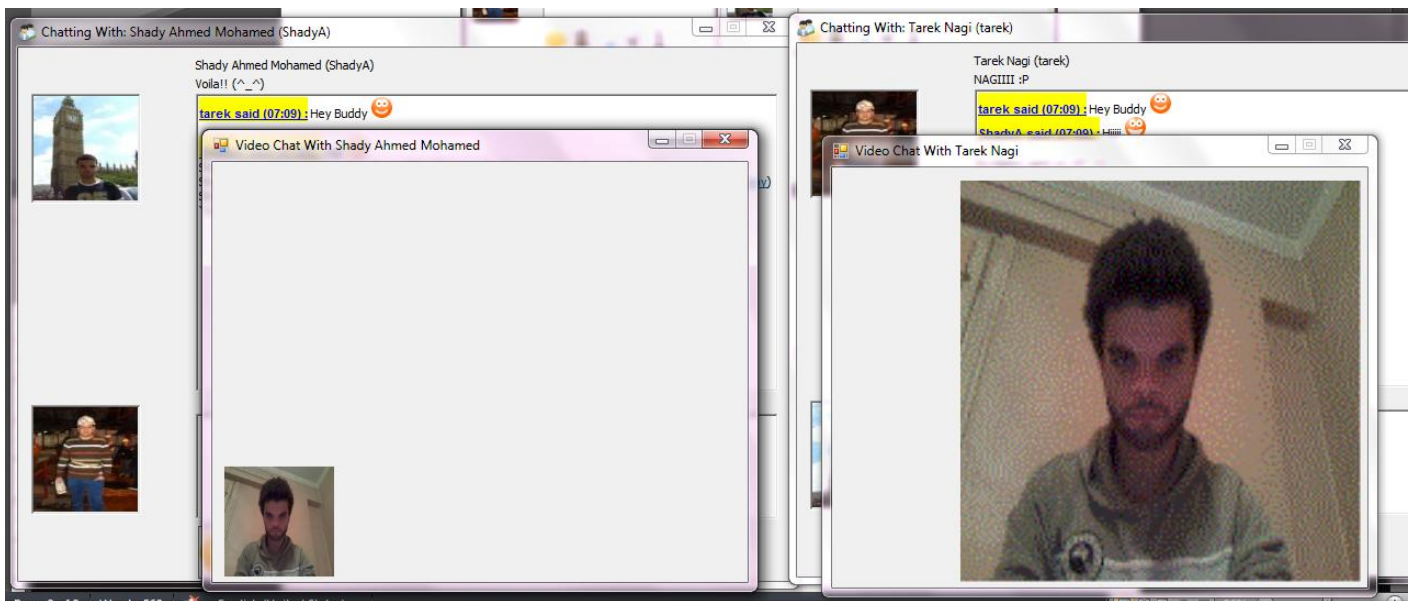
- Invitations sent:-



- File Sending:-



- Video Conference (Audio Enabled):-



- Smilies, Nudge, Send a file, Video Conference & Audio Chat:-

