Project Title: Achieve Message Resiliency and Disaster recovery using queues

Organizations require a system where they can send and receive messages or data ,over the network. The data in this process can be lost due to different reasons like a network issue or a crash on the client or server side. In such a scenario ,recovery of data that is sent into the queue is important. Also system should be robust enough to retain data in case client is not available to consume data.

Problem Description

1.A system is required to send and receive messages across network

2.The messages should stay in queue if client is not available

3.The messaging system should be asynchronous in nature

4.The system should have capacity to recover from a disaster in case of a crash

Expectation

Messages should be sent from system to system over network in a secure way and monitoring of system should be available. System should be recoverable from any crash or disaster scenario and message should not be lost. The messages pending in message queue should be visible and recoverable. The system should be scalable to accept more clients for consuming messages

Solution

User Stories

1.A module should be created for authentication of user. Their will be two types of users ,client and Administrator

2.A module should be created for registering machines that will be sending and receiving messages through the system

3.A module should be created to view status of machines that are currently connected to the existing application

4.The client should be able to see ,other clients who are online

5.The client should be able to connect any client and send a message as text ,or file

6.The client should receive notifications for file or messages sent to his system

7.A module should be available on client side to track previous messages and files

8.The client should be allowed to download files sent through application

9.A module should be available to change settings of application for administrator and separate set of settings for client . Administrator can register or unregister machines , and monitor queue for messages that are not delivered to client and waiting in queue.

10.In case of crash on client side, all pending messages should be recovered by administrator

11.In case of disaster on administrator side ,all data should be saved as log in separate file

In a location desired by administrator and should be recovered by him when system is reinstalled

12.A separate module for managing dead letter queue should be available for administrator

13.The administrator should get warning messages for dead letter queue

14.The system should monitor for Poison Pill scenario and identify messages that can generate such scenarios and block them

15.All messages should be properly serialized at one end and deserialized at other end .Administrator should be allowed to select the type of serialization to be used for the system