END TO END ENCRYPTED CHAT PLATFORM

Problem Statement :- The task is to make a chat machine which allows to users to interact with each other through messages and also to ensure that anyone in between cannot read their messages.

Design :- This is implemented by first making an encrypter, here I have implemented HILL CIPHER which translates (maps) a purely alphabetical message to another alphabetical message through the use of matrices (for which I have used math/matrix library). I have made both encrypter(encrypt-by-hill-cipher) and decrypter(decrypt-by-hill-cipher) to enure it works both ways.

Now to our chat machine, I have used two classes, the first one is sys% class that is basically a class which contains all server elements and functions the server can perform (some explaination of each function is given on side of each function in comments), The next class that I have used is account% class, this is the class which helps a user make account on the server(system), also I have created my-system as a default server of type sys%. Though a user never directly goes through account class (the user always uses my-system to sign-up login send-message recieve-message, logout from the system). Whenever two users start to interact they always do so by first creating a chat session between the two (which is like sending a friend request on facebook), this creating of chat session also randomly gives the two users one key randomly from the set keys stored in system which is basically the key that will encrypt the messages on both sides. The message when sent is encrypted in account class and reciever in the server

recieves the message in encrypted form so system never sees the actual message sent by the user, then the reciever also diverts the message the desired address and the message is then decrypted by the recieving user with their session-key. The message is then stored in inbox of reciever and simultanously also in chat area (in chat area the history of all messages both sent and recieved is stored) of both users.

Limitations/Bugs :-
-> Although the messages are always encrypted when sent but the administrator can always see through the message(Refer to line 89 and 99 of the code).
-> There is no GUI to use this chat machine and works only through terminal.
-> The encryption is very weak as it has only 28 possible keys though it can be made stronger with 1 million keys (Refer line 35 of code).
-> This is not an online chat machine and just a model of actual chat machines.

What is the scope of further improvement :-
-> Develop a GUI to use this chat machine through racket/gui.
-> Use better encryption methods like Fiestel encryption to ensure more security.

;;; Note use the test cases to know more about working of the code. (test cases are saved as testcases.rkt).