User Manual

Introduction:

A messenger app is any app that enables communication between one or more people. With more people having access to mobile devices that can connect to the internet, users seek to communicate with family and friends over text rather than calling as it is more private and less intrusive. Using a messenger app can also be seen as an alternative to paying for an expensive data plan since anyone anywhere with a wifi connection can access the application and message away to their heart's content. Without a mobile messaging app people in some countries such as Bangladesh would have to pay an exorbitant per text charge. Our application currently supports both one to one and group messaging depending on who the host user allows to join their chat. Users seeking to communicate with their entire family simultaneously, figuring out plans for the night with their friend group, or a professor seeking to get a message out quickly to his class are all people that would benefit from this group message feature. On the other hand, someone reaching out to their partner establishing dinner plans, a daughter checking in on her father are people who would appreciate the one-to-one feature of our messaging app.

Privacy is another concern with mobile messaging, generally people do not like the idea of having someone read their personal conversations, similar to a stranger eavesdropping on a conversation you're having with a friend. To solve this problem, our application offers secure end to end encryption that makes it impossible for malicious users to extract information that does not belong to them. This form of encryption ensures that only you and the person you want to talk to can see the messages that are being sent. End-to-end encryption works by encrypting

your data before it is sent and is not decrypted until it is read by the receiver. That means that even the messaging app that is hosting your conversation only has access to the encrypted version of your messages. Although all users value their data being secured, users who use this platform to send confidential information are protected from having their data being seen by those who shouldn't see it. Users who use our platform for communicating potentially illicit subjects are protected from government surveillance that could lead to an unwanted arrest.

Background and Motivation:

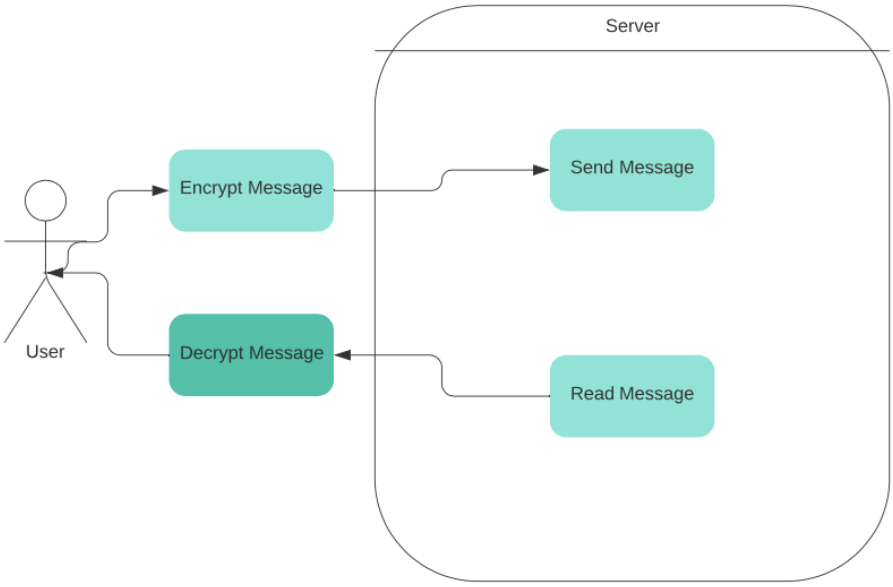
We sought to create a messaging app that would allow users over the same network to securely and efficiently manage communication.

We began our research by trying to understand networking and how exactly sending a message over the internet works. At this point we learned about network configurations and the client-server model. This model is actually quite simple, you can think of the server as just a computer who sits waiting for requests from clients (other computers) to come in. Once connection is established with the server the client sends over their request and the server fulfills said request. In the context of our messaging application our server waits for users to connect, reads their message then sends the message to the other users that are also connected. Professor King's lecture on networking and java.net helped us understand how to begin building our client and server classes. A stack overflow article titled "How do I send a message over a network" that we found by conducting a simple google search that almost identically matched the title of this article pointed us towards some pseudocode and information that explained how to easily send

one simple message over a network. This stack overflow article pointed us towards an article called “Reading from and Writing to a Socket” from the oracle java documentation. Simply put, a program establishes connection to a server through something called a socket. The client can send and receive data from the server using that socket.

The second biggest part of the research we conducted for this project was how to ensure that messages were secure between clients while traveling to, through, and from the server. Here we learned about end to end encryption. As the name suggests this form of encryption encrypts the message from end to end as in, the right as the client clicks send, the message is encrypted and right before the message is read on the other end of the server, the message is decrypted. The actual encryption itself is a simple cesar cipher. Every character in the message is adjusted by a private randomized key. For example if the key is two, ‘a’ becomes ‘c’, ‘b’ becomes ‘d’, and so on.

UML Use Case Diagram:



Pictures of Project and Sample User Input: