



Project Documentation

“Chaty”

Idea: -

The app has been built on the idea of the IRC service, that you can chat with group of people just using the relay you send and the server relay to any one with you in the room.

As I hope to call the app the “big S”

Secure:

Provide the security you want to have, Chaty doesn't store any messages in the room, so you don't have to worry on your info and chats you send and receive.

Speed:

Faster and speed communication as we don't make a lot of overhead on the client nor on the server side, which 'll make you send and receive on the fly.

Simple:

As you can see from the moment you open the app the simplicity of the design and implementation makes the user interacts with the app easily and faster without too much complicated stuff and also gives you the ability to add more stuff and functionality to the app easily as the project structure built essentially on that.

I call the app “Chaty” as the app provides fast communication between you and anyone else you want to explain to him or them anything you just need to make a room and a password and send that to the people you want them to join to you.

Features: -

- [1] create a room with only your nickname and room name and room password.
- [2] join a room with only your nickname and room name and room password.
- [3] create more than one room with the same room name but different password.
- [4] info on how to use the apps features inside the app.
- [5] send and receive messages and know how many clients received this message.
- [6] delete your messages from your side and others side.
- [7] clear all the history of your room on your side.
- [8] live functionality provides to you the ability to show only messages to the connected people in the room right at this moment only.
- [9] show count of how many messages you have sent and received.
- [10] log out so you don't receive messages any more from this room.
- [11] more about us button to know other apps we provide and connections to us.

Programs we have used: -

Program	Usage
Android Studio	To build the client side (what user uses)
GitHub	To save and store the apps step miles
Word	To write project document

Programming Languages we have used: -

Language	Usage
Java	- in the client-side app to write an android app.

Markup Languages we have used: -

Language	Usage
xml	- in the client-side app to write the user interface

Tested On: -

Device	Spec.
Lenovo K6	5.0 inches, 1080 x 1920 pixels, Android 6.0
Lenovo Tab S8	8.0 inches, 1200 x 1920 pixels, Android 5.0

Project in detail: -

- Android app (client side)

Android App

As Chaty has been designed on modularity so you can add more functionalities easily so until now these are the modules inside the app.

The whole code is commented and organized in modules and here is the description for the modules: -

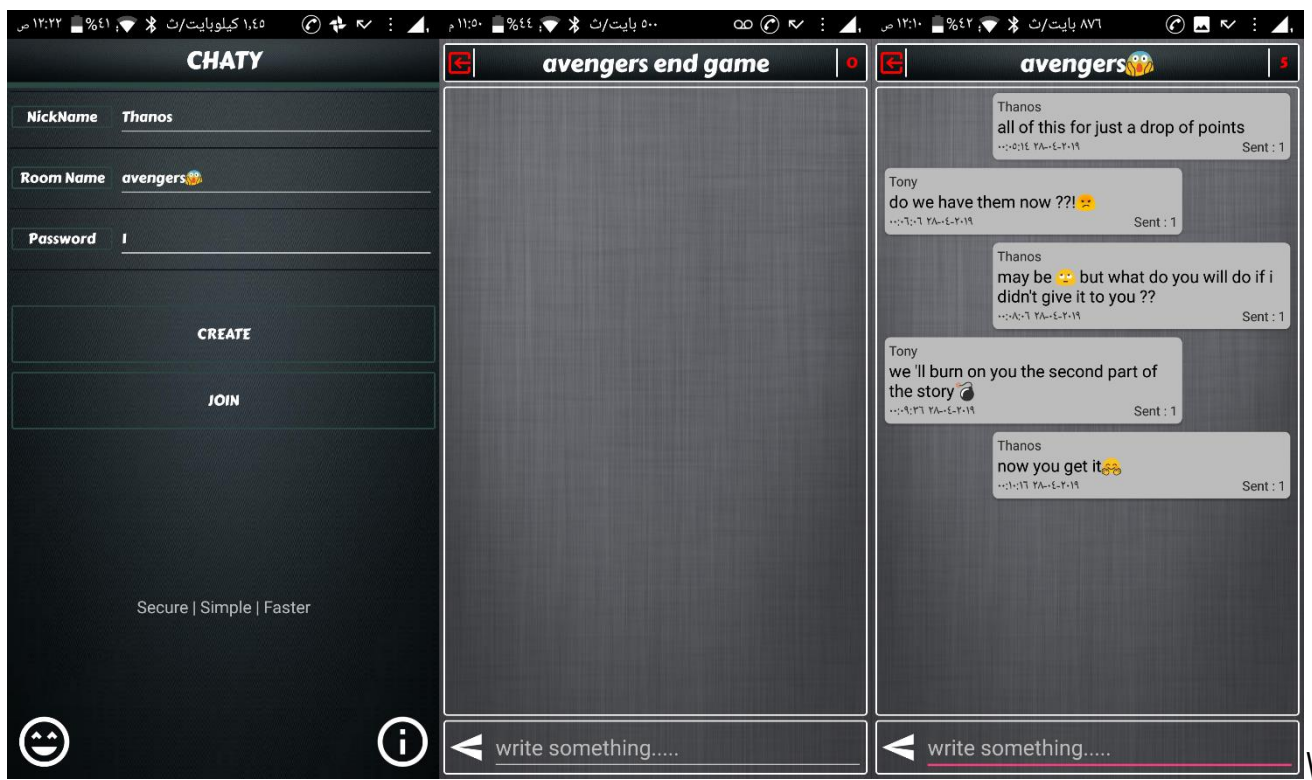
Android Modules	Usage
Adapters	Until now it has one class to use it to show the messages in the fragment chatFragment - MessageAdapter : to show the messages in the fragment chatFragment.
Connections	Has the receiver and sender connection and used to receive and send to clients
Controllers	Responsible on the logic in the client app - ChatController : responsible on all the logic in the chatFragment - SignInUpController : responsible on all the logic in CreateRoomFramgent - Controller: have the common functionalities and variables among other controllers
Data	To preserve the temporary data to in the client - Data : save current room name, room id and nickname - Message : same as what in the server side to send it to clients - Server : store the ip and ports of the server
Fragments	The view side - AboutMeFragment : identifies myself and apps and connections to me - ChatFragment : to call the queries you want - CreateRoomFragment : the create and join layout view - InfoFragment : show to you how to use the program
Functions	Include all the generic functions - Toasts : to show you beautiful success/error message - UpdateConnection : to periodically send to the server the alive message
Processing	To decode and encode the messages before sending it to server - MsgDecoder : responsible for all the decoding of the messages requests and responses from the server - MsgEncoder : to encode all the messages and add the functionality of the message before sending to the server
MainActivity	To load all the fragments from it and open the port of the client to listen to
Processing	To process the requests - ConnectedIps : to delete if a client didn't connect to the server from a while - MsgDecoder : to decode all the messages from clients - MsgEncoder : to encode all the messages to the clients

Android XML	Usage
activity_main	To open on it all the fragments later
fragment_chat	The view to see the room chat
view_chat_header	The header of the chat view
view_chat_input	The text you put into the message you send
view_chat_message	The cardview of the message
view_sided_chat	Adding the two view_chat_message to get two sides
view_chats	The recyclerview that you see the message in
fragment_create_room	The first view appears to you when you open the app
fragment_info	How to use Chaty

- The client listens on the port 5000.

Images from Tests: -

Lenovo K6



Lenovo Tab S8

