**BÀI TẬP LỚN 1**

**Môn học: Mạng Máy tính**

1. **Nhiệm vụ:**

Xây dựng ứng dụng messenger đa năng.

1. **Yêu cầu:**
2. Giúp người dùng ở các máy tính khác nhau liên lạc nhanh chóng.
3. Chia sẻ file qua giao thức P2P.
4. Hiện thực chat bằng P2P. Mô hình client-server được dùng để nhận và xử lý dữ liệu.
5. Voice messaging: cho phép người dùng liên lạc bằng giọng nói.
6. Stickers & reactions: cho phép người dùng thể hiện cảm xúc đối với các tin nhắn.
7. Chat ẩn danh: cho phép người dùng tạo nhóm chat trong đó người tham gia được ẩn danh tính.
8. Ghim tin nhắn: cho phép lưu lại những tin nhắn quan trọng để tiện đọc
9. Cloud storage: sử dụng Cloud Firestore làm cơ sở dữ liệu.
10. OAuth2 Authentication.
11. Danh sách bạn bè: mỗi người dùng sẽ có một danh sách bạn bè. Chỉ những người trong danh sách bạn bè của người dùng mới có thể chat được với người đó. Người dùng có thể kết bạn với nhau.
12. Block: Chặn tin nhắn của người dùng bất kỳ trong danh sách bạn bè. Người bị chặn sẽ không thể gửi tin nhắn đến người chặn.
13. Ignore: Phớt lờ tin nhắn của người dùng bất kỳ trong danh sách bạn bè. Người sử dụng tính năng này vẫn có thể nhận tin nhắn từ người bị phớt lờ, nhưng sẽ không nhận thông báo có tin nhắn mới.
14. QR Scanning: Kết bạn thông qua quét mã QR. Mỗi người dùng sẽ có một mã QR để quét.
15. Games: Người dùng có thể chơi game với nhau trong lúc chuyện trò.
16. **Hiện thực:**
17. Ngôn ngữ: Java, NodeJs.
18. Nền tảng: Android.
19. Giao diện: Được hiện thực bằng Android Studio.
20. Server: Server được hiện thực bằng NodeJs. Đó là một server có khả năng cung cấp API để các client trên android có thể truy cập cơ sở dữ liệu, nhằm mụ đích
21. Database: Cloud Firestore là một cơ sở dữ liệu NoSQL của Google và được đặt trên cloud.

Schema sơ bộ:

users:

user1:

uid: (user1's uid),

isInRooms: [

roomA: True,

roomB: True

]

user2:

uid: (user2's uid),

isInRooms: [

roomA: True,

roomB: False

]

user3:

uid: (user3's uid),

isInRooms: [

roomA: False,

roomB: True

]

rooms:

roomA:

name: "user1\_user2",

messages:

message1:

content: "Hi",

sentAt: (some timestamp),

sender: (user1)

message2:

content: "Hello",

sentAt: (another timestamp),

sender: (user2)

users:

user1:

uid: (uid of user1),

alias: "conbocuoi"

user2:

uid: (uid of user2),

alias: "thuythumattrang"

roomB:

name: "user1\_user3",

messages:

message1:

content: "Will you marry me?",

sentAt: (a timestamp),

sender: (user1)

message2:

content: "no.",

sentAt: (a sad timestamp),

sender: (user3)

users:

user1:

uid: (uid of user1),

alias: null

user2:

uid: (uid of user3),

alias: null