**TRƯỜNG ĐẠI HỌC TRỰC TUYẾN FUNIX**

**MÔN HỌC: HỆ THỐNG CƠ SỞ DỮ LIỆU**

**MÃ MÔN HỌC: DBI202x\_0101\_FX\_VN**

**BÀI TẬP LỚN SỐ 1: THIẾT KẾ MẠNG XÃ HỘI**

**SV: Nguyễn Quốc Tuấn**

**MSSV: CC00770x**

1. **LỜI MỞ ĐẦU**

Mạng xã hội là dịch vụ kết nối các thành viên cùng sở thích trên Internet lại với nhau với nhiều mục đích khác nhau không phân biệt không gian và thời gian . Những người tham gia vào dịch vụ mạng xã hội còn được gọi là cư dân mạng, cộng đồng mạng.

Dịch vụ mạng xã hội có những tính năng như chat, e-mail, phim ảnh, voice chat, chia sẻ file, blog và xã luận. Mạng đổi mới hoàn toàn cách cư dân mạng liên kết với nhau và trở thành một phần tất yếu của mỗi ngày cho hàng trăm triệu thành viên khắp thế giới. Các dịch vụ này có nhiều phương cách để các thành viên tìm kiếm bạn bè, đối tác: dựa theo group (ví dụ như tên trường hoặc tên thành phố), dựa trên thông tin cá nhân (như địa chỉ e-mail hoặc screen name), hoặc dựa trên sở thích cá nhân (như thể thao, phim ảnh, sách báo, hoặc ca nhạc), lĩnh vực quan tâm: kinh doanh, mua bán...

Xét về mức độ phổ biến hiện nay ở Việt Nam hiện nay, những kênh mạng phổ biến nhất có thể kể đến là Facebook, Zalo (chức năng chính dùng để kết nối bạn bè), Youtube (chức năng giải trí), YmeetMe (chức năng chính là hẹn hò),….

Ở phạm vi bài tập thiết kế này, em xin trình bày lại một phần “**thu nhỏ của thiết kế Facebook**” nhằm mục đích chính là nắm được các **khái niệm cơ bản của Hệ cơ sở dữ liệu**, biết cách sử dụng **ERD và Database Schema**. Trong buổi trao đổi qua kênh tổng đài Hannah của Funix, sau khi hỏi đáp với Mentor về nội dung thiết kế, em được hồi đáp rằng nội dung của thiết kế phát triển đến mức nào sẽ tùy thuộc vào nhu cầu sử dụng. Xét đến Facebook là một trong những mạng xã hội phổ biến nhất, lớn nhất hiện nay nên việc mô phỏng chi tiết thiết kế Facebook là một đề tài tương đối lớn, vì thời gian học tương đối hạn hẹp cho nên em chỉ xin phép trình bài những tính năng tối thiểu và đơn giản nhất, nếu có thiếu sót mong Mentor tận tình chỉ bảo.

1. **PHÂN TÍCH**

**Phân tích yêu cầu nâng cao của bài tập**

Do yêu cầu nâng cao ở bài này em cho là hơi khó để hình dung tiêu chuẩn đánh giá nên em xin phép phân tích lý do em làm phần nâng cao trước, sau đó sẽ trình bày tiếp đến phần cơ bản.

Nếu xét về khía cạnh hoàn hảo thì không có gì gọi là đủ, cho nên các mạng xã hội ngày nay luôn luôn không ngừng cho ra những tính năng mới và các bản thiết kế theo đó cũng sẽ có sự thay đổi. Một trong những ý tưởng mới em muốn áp dụng cho Facebook theo **yêu cầu nâng cao** của bài tập, đó chính là tích hợp **tính năng hẹn hò** (một kiểu tương tác giữa người dùng với nhau) vào trong Facebook. Em nghĩ đây là một ý tưởng khả thi, bởi vì em có một số phân tích như sau:

- Thứ nhất, về các tính năng, trải nghiệm người dùng, sau khi em sử dụng cả Facebook và YmeetMe em có nhận xét YmeetMe tương tự như một bản thu nhỏ không đầy đủ chức năng của Facebook, điều khác biệt duy nhất chính là tính năng thống kê và tìm ra những người dùng có điểm tương đồng về tính cách, sở thích để kết đôi hẹn hò. Cho nên việc phát triển thêm tính năng hẹn hò sẽ khiến Facebook ngoài những tính năng rất mạnh đã có sẽ có thêm trải nghiệm mới cho người dùng.

-Thứ hai, về mặt thị trường Châu Á, một nơi vốn rất nhiều tín đồ về PhongThủy, với các hình thức phổ biến như là xem tuổi kết hôn, tuổi vợ chồng hợp nhau hay không, hướng xây nhà,… thì trong tính năng hẹn hò nếu có thể xây dựng công cụ tìm kiếm đối tượng phù hợp dựa trên Tử Vi về kết hôn em nghĩ cũng không phải ý tưởng tồi. Sở dĩ em cho điều này là khả thi bởi vì trên thị trường SoftWare hiện nay đã có rất nhiều ứng dụng liên quan tuổi kết hôn, nếu sao chép các chức năng này vào tính năng hẹn hò em nghĩ sẽ có thêm một trải nghiệm tuyệt vời cho người dùng Facebook.

-Thứ ba, về mặt tiền đề để tiếp tục nghiên cứu và phát triển thì, nếu như đã làm được tính năng hẹn hò để thay thế các mạng xã hội hẹn hò thì tại sao lại không nghĩ đến việc tạo một tính năng tương tự như là việc tìm người, người tìm việc? Mặc dù những trang web chuyên tìm việc rất nhiều nhưng khi lướt facebook em vẫn thấy các group tuyển dụng đăng tin, đăng bài lý do là bởi vì ở Việt Nam khá là nhiều lao động phổ thông biết sử dụng facebook, tuy nhiên lại không biết đến web tuyển dụng hoặc không biết cách thức đăng thông tin cá nhân lên web tuyển dụng. Ý tưởng phát triển tiếp theo của em là tạo một tính năng tìm việc giúp cho người dùng liệt kê những điểm mạnh của mình để ứng tuyển, những yêu cầu của mình để tuyển dụng, nếu người tuyển và người ứng tuyển có những yêu cầu hợp nhau ở khoảng % tương đối nào đó thì sẽ có công cụ tự thông báo (gửi message chẳng hạn) để kết nối người dùng với nhau.

Với những lý do như trên, em nghĩ việc thêm một tính năng giống với các mạng xã hội khác nhằm chiếm lĩnh số lượng người dùng chính là một hướng đi mà Facebook nên cân nhắc về khả năng phát triến sau này. Ở phần hiện thực hóa thành thiết kế ở bài tập này, em sẽ thêm **chức năng hẹn hò cho Facebook**. Ở đây về ý tưởng là em sẽ thống kê thông tin cá nhân, sở thích, tiêu chuẩn tìm người yêu,… nói chung là tất cả các yếu tố cần thiết để xác định hai người có hợp nhau không, cuối cùng hiển thị dưới dạng một số phần trăm tỉ lệ hai người hợp nhau.

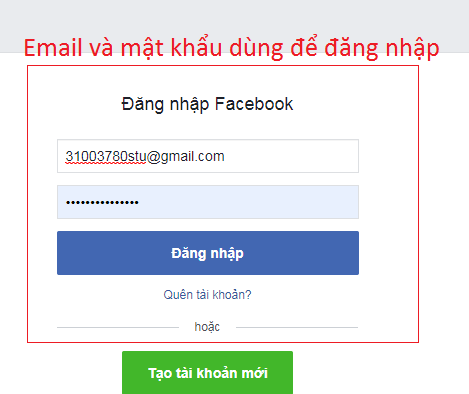
Ngoài ra, về mặt trải nghiệm người dùng, em thấy có trường hợp những bình luận rất hay tuy nhiên muốn chia sẻ lại bình luận thì người dùng toàn phải chụp màn hình rồi mới viết bài đăng dưới dạng ảnh, cho nên em đề xuất thêm tính năng **chia sẻ bình luận** **dưới dạng bài đăng**, tự động chụp màn hình và chia sẻ lại bình luận.

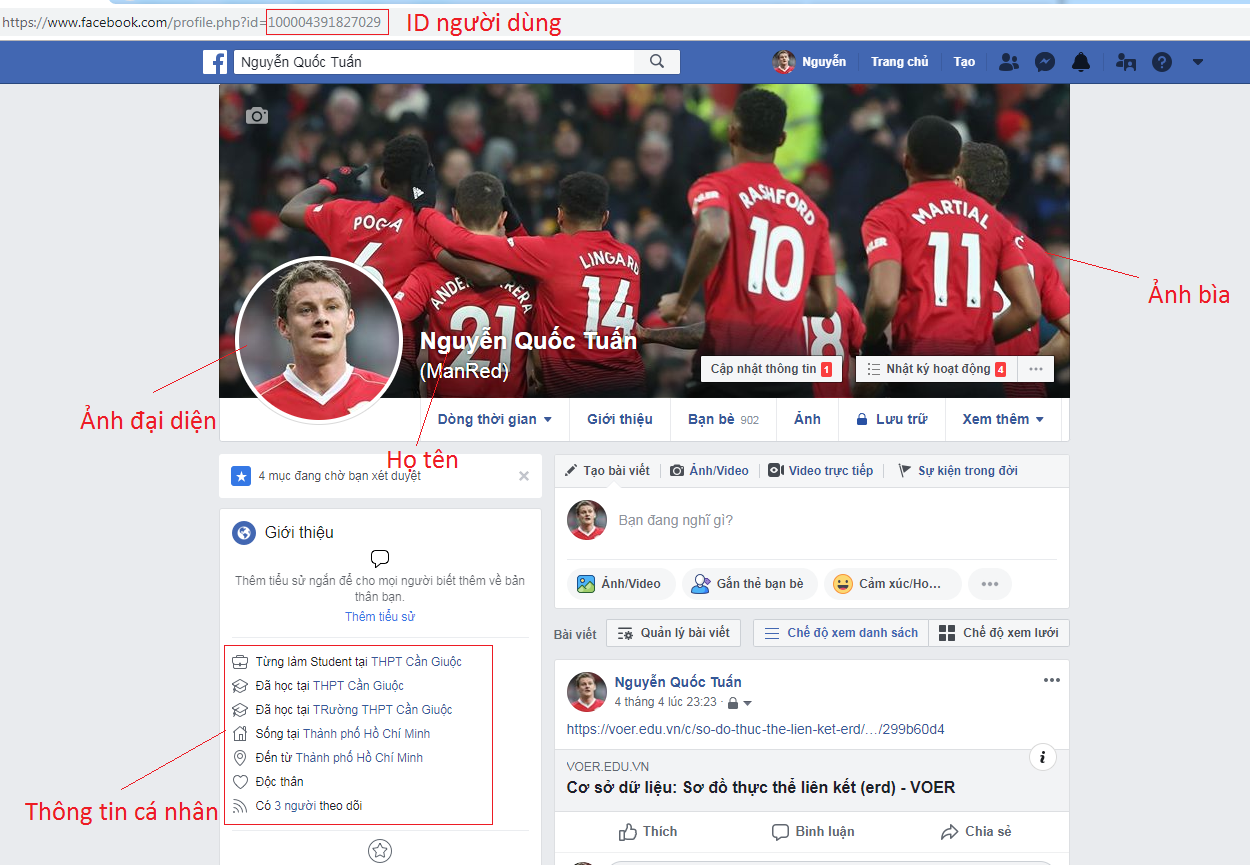
**Phân tích yêu cầu cơ bản của bài tập**

Facebook có rất nhiều thành phần, tuy nhiên trong bài này em xin phân tích những đặc điểm, đối tượng cơ bản có ở trong facebook như sau:

Về các thực thể cơ bản, vì những lý do độ phức tạp đã trình bài ở phần mở đầu em xin lược bỏ bớt để quá trình làm bài được nhanh hơn (ví dụ như hiện tại người dùng có thể tạo nhóm cũng có thể xem là thực thể, ngoài nút thích còn có các biểu lộ cảm xúc khác,…). Sau khi giảm bớt, em xin trình bày những đối tượng cơ bản sau:

1. Người dùng có ID người dùng, email, mật khẩu, họ tên , ảnh đại diện, ảnh bìa, thông tin cá nhân (giới tính, ngày sinh, địa chỉ, nơi làm việc, học vấn, tình trạng hôn nhân)





1. Bài đăng có ID bài đăng, nội dung, thời gian đăng, chế độ hiển thị



1. Bình luận có ID bình luận, nội dung, thời gian đăng



Về các kiểu liên kết

1.Người dùng có thể tương tác với bài đăng dưới các hình thức: người dùng viết bài đăng, người dùng thích bài đăng, bài đăng được chia sẻ bởi người dùng



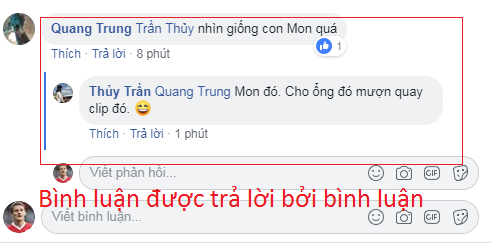
2.Người dùng tương tác với bình luận dưới các hình thức: người dùng viết bình luận, người dùng thích bình luận



3.Bình luận tương tác với bài đăng dưới hình thức: bình luận hiển thị trên bài đăng



4.Các bình luận tương tác với nhau dưới hình thức bình luận này được trả lời bởi bình luận khác



5.Người dùng có thể tương tác với nhau thông qua tin nhắn.



1. **Logical Design**
2. **USER** là một thực thể (entity) với các thuộc tính: **UserID, Email, Password, Name, Picture, Info**

+ Picture (gồm Avatar, Background) và Info (gồm Sex, Birthday, Address, Workplace, Education, MaritalStatus) là thuộc tính phức hợp (composite). Tất cả các thuộc tính còn lại đều là thuộc tính đơn (simple).

+**UserID** là thuộc tính mang giá trị duy nhất để phân biệt giữa người dùng này với người dùng khác, gọi là **khóa** (key attribute).

+Tất cả các thuộc tính của USER đều là thuộc tính đơn trị (single value)

1. **POST** là một thực thể (entity) với các thuộc tính: **PostID, PostContent, PostTime, DisplayMode.**

+**PostID** là thuộc tính mang giá trị duy nhất để phân biệt giữa người dùng này với người dùng khác, gọi là **khóa** (key attribute).

+Tất cả các thuộc tính của POST đều là thuộc tính đơn trị (single value)

1. **COMMENT** là một thực thể (entity) với các thuộc tính: **CommentID, CommentContent, CommentTime.**

+ **CommentID** là thuộc tính mang giá trị duy nhất để phân biệt giữa người dùng này với người dùng khác, gọi là **khóa** (key attribute).

+Tất cả các thuộc tính của COMMENT đều là thuộc tính đơn trị (single value)

1. Liên kết **USER <WRITE\_POST>POST** là kiểu liên kết với tỷ số lực lượng là **1:N** bởi vì một USER có thể viết nhiều bài POST nhưng mỗi bài POST chỉ có thể được viết bởi một USER.

+Sự tham gia liên kết(Participation Constraint) của **USER** là **bộ phận**(partial participation) vì USER không nhất thiết phải viết POST

+Sự tham gia liên kết của **POST** là **toàn bộ** (total participation ) vì bất cứ POST nào cũng có USER viết

1. Liên kết **USER<LIKE\_POST>POST** là kiểu liên kết với tỷ số lực lượng là **M:N** bởi vì một USER có thể thích nhiều bài POST và ngược lại một bài POST cũng có thể được nhiều USER thích.

+Sự tham gia liên kết của cả **hai phía đều là bộ phận** bởi vì USER có thể không thích bất cứ bài POST nào và ngược lại một bài POST cũng không nhất thiết phải có USER thích

1. Liên kết **POST<SHARE\_POST>USER** là kiểu liên kết với tỷ số lực lượng là **M:N** bởi vì một POST có thể được chia sẻ bởi nhiều USER và ngược lại một USER cũng có thể chia sẻ nhiều POST

+Sự tham gia liên kết ở cả **hai phía đều là bộ phận** vì POST không nhất thiết phải được chia sẻ và USER cũng không nhất thiết phải chia sẻ bất kỳ bài POST nào

+Liên kết này có thêm thuộc tính đơn trị **SharePostTime** để biểu thị thời gian USER chia sẻ POST

1. Liên kết **USER <WRITE\_CM>COMMENT** là kiểu liên kết với tỷ số lực lượng là **1:N** bởi vì một USER có thể viết nhiều COMMENT nhưng mỗi COMMENT chỉ có thể được viết bởi một USER.

+Sự tham gia liên kết của **USER** là **bộ phận** vì USER không nhất thiết phải viết COMMENT

+Sự tham gia liên kết của **COMMENT** là **toàn bộ** vì bất cứ COMMENT nào cũng có USER viết

1. Liên kết **USER<LIKE\_CM>COMMENT** là kiểu liên kết với tỷ số lực lượng là **M:N** bởi vì một USER có thể like nhiều COMMENT và ngược lại một COMMENT cũng có thể được nhiều USER like.

+Sự tham gia liên kết của cả **hai phía đều là bộ phận** bởi vì USER có thể không thích bất cứ COMMENT nào và ngược lại một COMMENT cũng không nhất thiết phải có USER thích

1. Liên kết **COMMENT <SHOW\_CM>POST** là kiểu liên kết với tỷ số lực lượng là **N:1** bởi vì một COMMENT chỉ có thể hiển thị trên một POST nhưng ngược lại một POST có thể chứa nhiều COMMENT.

+Sự tham gia liên kết của **COMMENT** là **toàn bộ** vì bất cứ COMMENT nào cũng hiển thị lên POST

+Sự tham gia liên kết của **POST** là **bộ phận** vì POST không nhất thiết phải có COMMENT

1. Liên kết **COMMENT <REPLY>COMMENT** là kiểu liên kết với tỷ số lực lượng là **1:N** bởi vì một COMMENT có thể được trả lời bởi một hoặc nhiều COMMENT khác tuy nhiên mỗi COMMENT chỉ trả lời cho một COMMENT mà thôi

+Sự tham gia liên kết của cả **hai phía đều** là **bộ phận** vì một COMMENT thì không nhất thiết phải dùng để trả lời COMMENT khác, ngược lại mỗi COMMENT cũng không nhất thiết phải có COMMENT trả lời.

1. Liên kết **USER<SEND\_MSG>USER** là kiểu liên kết với tỷ số lực lượng là **M:N** bởi vì một USER có thể gửi tin nhắn cho nhiều USER khác và ngược lại mỗi USER cũng có thể nhận được tin nhắn từ nhiều USER

+Sự tham gia liên kết của cả **hai phía đều** là **bộ phận** vì một USER thì không nhất thiết phải nhắn tin cho USER khác, ngược lại mỗi USER cũng không nhất thiết phải được USER khác nhắn tin

+Liên kết này có thêm các thuộc tính đơn trị MessageText và MessageTime để biểu thị nội dung và thời gian nhắn tin và thuộc tính khóa MessageID để phân biệt các tin nhắn với nhau

**Xét đến phần nâng cao**

1. Liên kết **USER<SUITABLE>USER** là kiểu liên kết với tỷ số lực lượng là **M:N** bởi vì mỗi USER có thể thích hợp với rất nhiều người

+Sự tham gia liên kết ở cả hai phía đều là **bộ phận** vì USER không nhất thiết phải thích hợp với USER khác và ngược lại

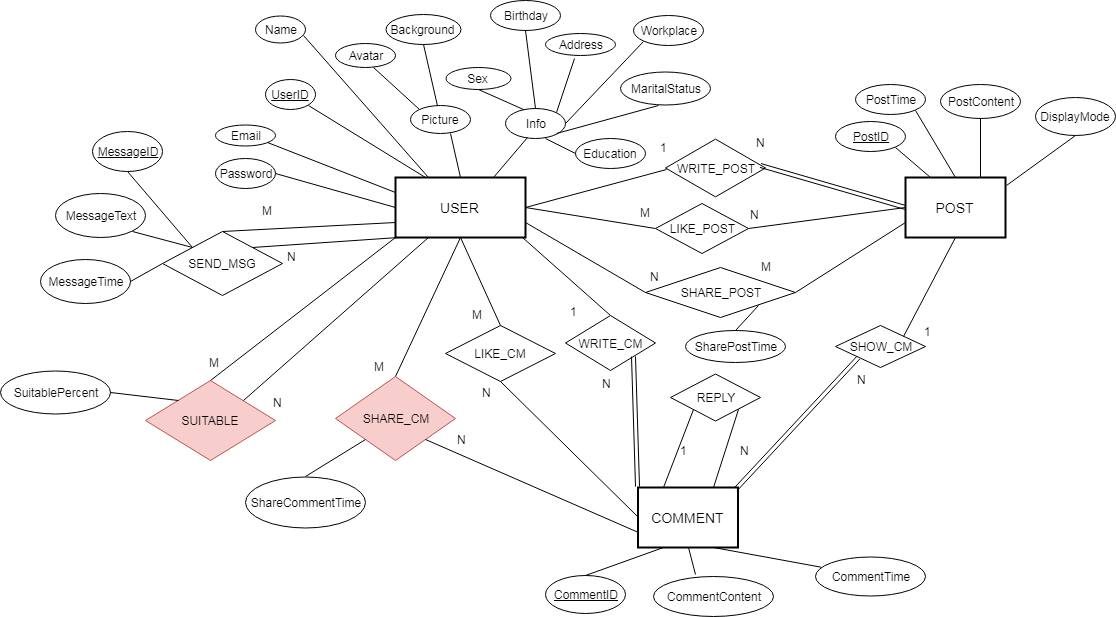
+Liên kết này có thêm thuộc tính đơn trị **SuitablePercent** để biểu thị mức độ phù hợp giữa hai người

1. Liên kết **COMMENT<SHARE\_CM>USER** là kiểu liên kết với tỷ số lực lượng là **M:N** bởi vì một COMMENT có thể được chia sẻ bởi nhiều USER và ngược lại một USER cũng có thể chia sẻ nhiều COMMENT

+Sự tham gia liên kết ở cả **hai phía đều là bộ phận** vì COMMENT không nhất thiết phải được chia sẻ và USER cũng không nhất thiết phải chia sẻ bất kỳ COMMENT nào

+Liên kết này có thêm thuộc tính đơn trị **ShareCommentTime** để biểu thị thời gian USER chia sẻ COMMENT

Sau tất cả các bước trên ta có lược đồ ERD như sau:



1. **PHYSICAL DESIGN**

**Bước 1: Với mỗi entity type bình thường (bỏ qua các subclasses / weak entities) tạo một relation tương ứng.**

**USER(UserID, Email, Password, Name, Avatar, Background, Sex, Birthday, Address, Workplace, MaritalStatus, Education)**

**POST(PostID, PostContent, PostTime, DisplayMode)**

**COMMENT(CommentID, CommentContent, CommentTime)**

**Bước 2: Với mỗi ID-dependent entity tạo một relation tương ứng. Relation này có khóa chính bao gồm khóa chính của relation cha và một thuộc tính của chính nó.**

**Do không có thực thể yếu nên bỏ qua bước này**

**Bước 3:  Bước này xử lý cho các supertype và subtype nên tạm thời các relation sẽ không thay đổi gì (Do phân tích trên không có supertype và subtype) .**

**Bước 4: Đối với mỗi quan hệ Unary hoặc Binary có kiểu 1-1. Tạo một Foreign key từ một entity và tham chiếu sang entity còn lại.**

**COMMENT(CommentID, CommentContent, CommentTime, ReplyID) trong đó ReplyID là khóa ngoại tham chiếu tới COMMENT được trả lời**

**Các quan hệ Unary có tỷ số lực lượng M:N chuyển xuống xử lý ở bước 6**

**Bước 5: Với mỗi quan hệ 1-N, đặt khóa ngoại vào relation phía N tham chiếu khóa chính của relation phía 1**

**COMMENT(CommentID, CommentContent, CommentTime, ReplyID, UserID, PostID) trong đó UserID và PostID là khóa ngoại tham chiếu đến USER và POST**

**POST(PostID, PostContent, PostTime, DisplayMode, UserID) trong đó UserID là khóa ngoại tham chiếu đến USER**

**Bước 6: Với mỗi quan hệ N-M tạo ra một relation mới, khóa chính của relation này bao gồm các khóa ngoại tham chiếu tới các relation tham gia vào quan hệ, nó cũng chứa luôn cả các attribute của quan hệ như  là attribute của chính nó.**

**LIKE\_POST(UserID, PostID) trong đó UserID và PostID là khóa ngoại tham chiếu đến USER và POST**

**SHARE\_POST(UserID, PostID, SharePostTime) trong đó UserID và PostID là khóa ngoại tham chiếu đến USER và POST**

**LIKE\_CM(UserID, CommentID) trong đó UserID và CommentID là khóa ngoại tham chiếu đến USER và POST**

**SHARE\_CM(UserID, CommentID, ShareCommentTime) trong đó UserID và CommentID là khóa ngoại tham chiếu đến USER và POST**

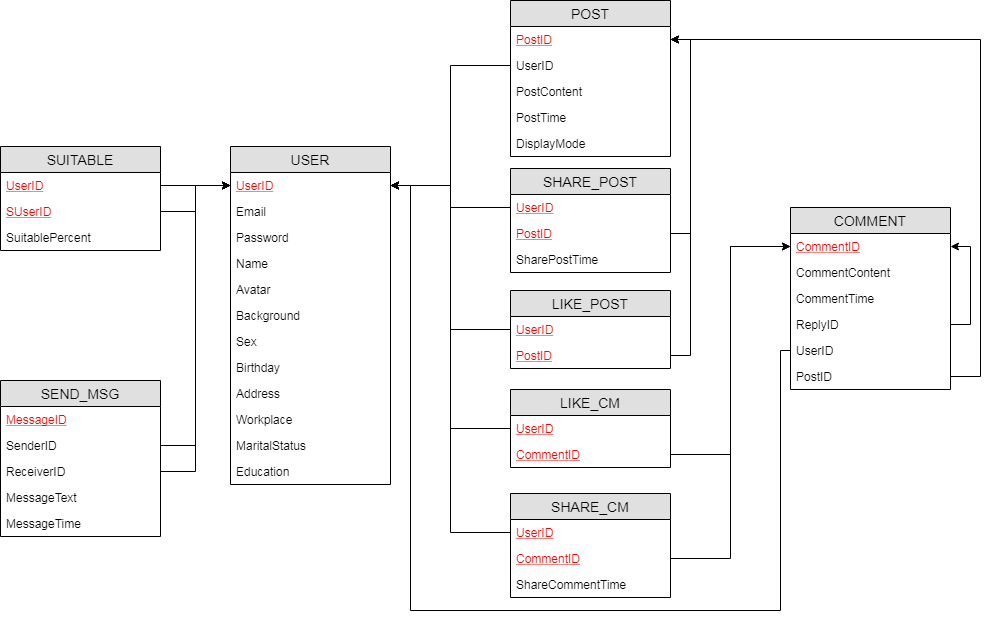
**SUITABLE(UserID, SUserID, SuitablePercent) trong đó UserID và SUserID đều là khóa ngoại tham chiếu đến USER**

**SEND\_MSG(MessageID, SenderID, ReceiverID, MessageText, MessageTime) trong đó SenderID và ReceiverID đều là khóa ngoại tham chiếu đến USER. Ở đây ta không dùng SenderID và ReceiverID để tạo khóa chính mà dùng MessageID để làm khóa chính vì trong trường hợp một USER gửi nhiều tin nhắn cho USER khác thì việc kết hợp SenderID và ReceiverID làm khóa chính sẽ không đảm bảo tính duy nhất.**

**Bước 7 : Tương ứng với mỗi multi-valued attribute, ta sẽ tạo một relation mới. Primary Key của relation này bao gồm primary key của bảng gốc và value attribute**

Ở đây ERD không thiết kế thuộc tính đa trị (nếu có thì có thể xét đến thuộc tính Education của USER tuy nhiên không bắt buộc, chỉ tùy vào nhu cầu sử dụng) cho nên không xử lý bước này.

Sau tất cả 7 bước ta có danh sách bảng hoàn chỉnh như bên dưới



1. **LỜI KẾT**

Bài tập với đề tài mô phỏng, thiết kế mạng xã hội là một đề tài khá hay và hấp dẫn, những yêu cầu của để tài đã giúp em có được cái nhìn cụ thể hơn về môn học hệ thống cơ sở dữ liệu cũng như là rèn luyện tư duy logic. Tuy nhiên vì khả năng của em còn hạn chế cho nên chắc chắn khó tránh khỏi sai sót. Mong mentor tận tình chỉ bảo. Em xin chân thành cám ơn và chúc mentor thật nhiều sức khỏe

**DANH MỤC CÁC ĐƯỜNG LINK THAM KHẢO ĐỂ LÀM BÀI**

<https://voer.edu.vn/m/mo-hinh-quan-he-thuc-the-entity-relationship-model/ff2250a9>

<https://www.youtube.com/watch?v=A3uj6X7IjJE>

https://www.youtube.com/watch?v=CZTkgMoqVss