**UNIVERSITY OF PUERTO RICO**

**MAYAGÜEZ CAMPUS**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**Conceptual Design**

**By:**

Coralis Camacho Rodríguez (coralis.camacho1@upr.edu)

Juan C. Cabrera Lebrón (juan.cabrera2@upr.edu)

Carlos A. Rodríguez Santiago (carlos.rodriguez75@upr.edu)

**Dr. Manuel Rodríguez Martínez**

manuel.rodriguez7@upr.edu

ICOM 5016 – Introduction to Database Systems

**Due Date**

28 March 2018

**ER Diagram**

**Entity Tables**

1. **Users**: Contains information of all users registered in the application. The information is limited to their display name, password, full names, phone and/or e-mail and a unique identifier for each user.
2. **GroupChats**: Contains information regarding active group chats in the application. This includes the name of the chat group, the date when it was created, the ID of the chat owner and a unique identifier for each chat. All group chats **must** have an owner of the Users entity.
3. **Messages**: Contains information of all messages that are readable through the application. This includes a text section containing the readable message, the date in which the message was posted, a Boolean indicating if the message contains at least one hashtag, the ID of the message author, and a unique identifier for each message. All messages **must** have an author of the Users entity and **must** also belong to a chat of the GroupChats entity. Also contains the total number of likes and dislikes given to each message.
4. **Hashtags**: Contains simple indicator of whether hashtags are included in a message or not. It includes a Boolean value which is true if there is a hashtag in a related message and false if there is not. It also includes the ID of the message which the Hashtag tuple references and a unique identifier. All hashtags **must** have a related message referenced.
5. **Reactions**: Contains information related to all reactions to a message (likes/dislikes). It includes a Boolean value which indicates whether the reaction is a like (true) or dislike (false), the ID of the User which issued the reaction, the ID of the message which was reacted to and a unique identifier. All reactions **must** have **both** a related User which issued the reaction and a Message which was liked or disliked.   
     
   **Relational Tables**
6. **Contacts:** Contains a reference to the unique IDs of two Users in each tuple. The first ID listed is that of a user who owns a contact list. The second ID listed is that of a user who is a contact of the first user listed. This information is sufficient to generate contact lists for all users.
7. **Participates:** Contains references to User IDs and GroupChats IDs. Each pair of User IDs and GroupChats IDs is unique. Each tuple indicates that a User with the ID in the first column is an active member of the GroupChat with ID specified in the second column.
8. **IsReply**: Contains references to two Message IDs. The first Message ID references an original message while the second Message ID references a reply to said original message. These pairs are unique.

**ER Diagram: Mapping**

create table Users(UID serial primary key, UDispName varchar(120), UPassword varchar(120), UFirst\_name varchar(120), ULast\_name varchar(120), UPhone char(10), UEmail varchar(120));

create table Contacts(UID integer references Users(UID), CUID integer references Users(UID), primary key(UID,CUID));

create table GroupChats(GID serial primary key, GName varchar(120), GCDate char(8), UID integer references Users(UID));

create table Participates(UID integer references Users(UID), GID integer references GroupChats(GID), primary key(UID,GID));

create table Messages(MID serial primary key, Message varchar(200), MDate char(8), MHashtag boolean, UID integer references Users(UID), GID integer references GroupChats(GID), NoLikes integer, NoDislikes integer);

create table IsReply(Or\_msg\_ID integer references Messages(MID), R\_msg\_ID integer references Messages(MID), primary key(Or\_msg\_ID, R\_msg\_ID));

create table Reactions(RID serial primary key, MReaction boolean, UID integer references Users(UID), MID integer references Messages(MID));

create table Hashtags(HTID serial primary key, HText varchar(120));

create table HasHashtags(MID integer references Messages(MID), HTID integer references Hashtags(HTID), primary key (MID,HTID));