SOCIAL MESSAGING APPLICATION

PHASE I – CONCEPTUAL DESIGN

Team Members:
Sujeily P. Fonseca González
Samuel G. González Ortiz
Lianne Sánchez Rodríguez

Professor Manuel Rodríguez
Database Management Systems
ICOM 5016, Section 096
Wednesday, March 28, 2018

TABLE OF CONTENT

SOCIAL MESSAGING APPLICATION

PHASE I – CONCEPT DESIGN

OBJECTIVES

1. Understand the design, implementation and use of an application backed by a database system.
2. Understand the use of the E-R model for database application design.
3. Gain experience by implementing applications using layers of increasing complexity and fairly complex data structures.
4. Gain further experience with Web programming concepts including REST and HTTP.

GENERAL OVERVIEW

The project presented in this document, titled Social Messaging Application, consists in the design and development of an application used for messaging in a social context. It will callow the following operations:

- ☐ Register a user with the application
- ☐ Create a chat group with a given name
- ☐ Add a user to your contacts list based on name, last name, and either phone or email address
- ☐ Add a contact to a chat group
- ☐ Remove a user from a chat group
- ☐ Remove a user from the contacts list
- ☐ Remove a chat group (only the owner)
- ☐ Post a message to a chat group
- ☐ Read the messages from a chat group
- ☐ Like a message posted on a chat group
- ☐ Dislike a message posted on a chat group
- ☐ Reply to a message posted on a chat group

- ☐ Post an image to a group (optionally)
- ☐ Post a video to a group (optionally)

Furthermore, the application will contain a web-based dashboard to provide statistics, such as the mentioned below:

- ☐ Trending topics via #hashtags
- ☐ Number of message per day
- ☐ Number of replies per day
- ☐ Number of likes
- ☐ Number of dislikes
- ☐ Active users

The data in the application is managed by a relational database system and exposed to client applications through a REST API.

E-R MODEL

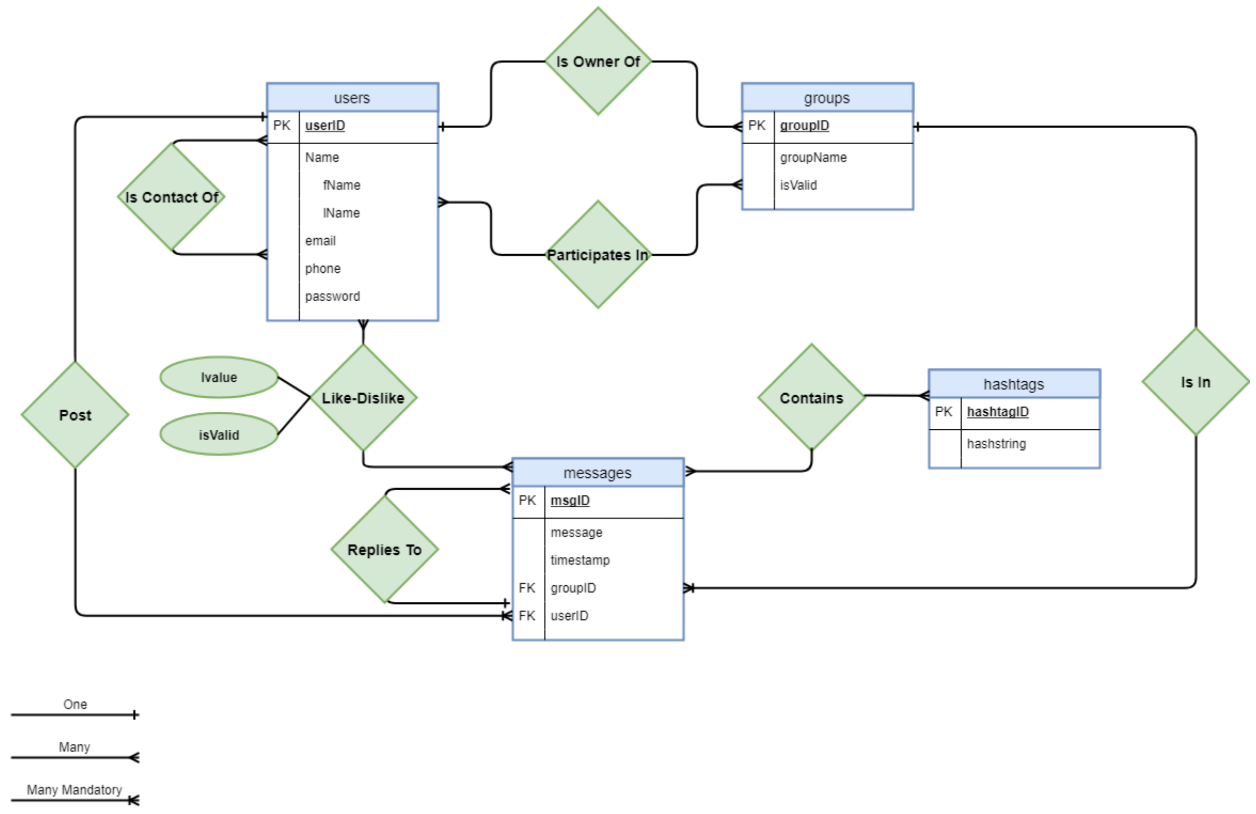

Figure 1: *E-R Diagram*

The Entity Relationship Diagram shows the relationships of entity sets stored in a database. Relationships can be described as follows:

- ☐ **Users:** Users table is related with itself with many to many relationships with users, messages and groups tables. Furthermore, it is related with groups or chats with a one to one relationship based on ownership. Finally, since a user can post multiple messages, the user table is related with messages table with a one to many relationship, having total partition of messages.
- ☐ **Messages:** Messages table have a role relationship with itself because a message can have multiple responses provided by users.
- ☐ **Groups:** Groups table is related with the messages table with a one to many relationship, since a group can contain multiple messages. Furthermore, total partition is present in the messages side.

- **Hashtags:** This table is related with the messages table with a many to many relationship.

Other relationships can be appreciated in the E-R Diagram presented in Figure 1.

MAPPING FROM E-R TO SQL TABLES

create table users(userID serial primary key, fName varchar(50), lname varchar(50), email varchar(50), phone char(10), password varchar(30));

create table groups(groupID serial primary key, groupName varchar(100), isValid bit);

create table ParticipatesIn(ownerID integer references users(userID), groupID integer references groups(groupID), primary key(ownerID, groupID));

create table messages(msgID serial primary key, message varchar(350), userID integer references users(userID), groupID integer references groups(groupID), timestamp);

create table RepliesTo(msgID integer references messages(msgID), replyID integer references messages(msgID), primary key (msgID, replyID));

create table Reactions(reactionID serial primary key, lvalue bit, isValid bit, userID integer references users(userID), msgID integer references messages(msgID));

create table Hashtags(hashtagID serial primary key, hashstring varchar(80));

create table Contains(msgID integer references messages(msgID), hashtag integer references hashtags(hashtagID), primary key (msgID, hashtagID);