Department of Electrical and Computer Engineering
University of Puerto Rico Mayagüez Campus
CIIC 4060/ICOM 5016 – Introduction to Database Systems

Spring 2018 Term Project
Social Messaging Application Phase II:
Conceptual Design

Gladymar O'Neill
Kristalys Ruiz
Cristian Duque

May 1, 2018

User: Refers to every user in the UI.

Attributes: uid: user id, primary key. [serial]

name: first_name and last_name composite attribute. [varchar(30)]

phone: phone number a user uses to register in the system. [varchar(11)]

email: email used to register to the system. [varchar(40)]

password: password for the user login [varchar(10)]

is_active: will determine if the user is currently active. [boolean]

Relationships of **User:**

Owns: A user is mapped to **Chats**. A user can be the administrator of a chat

Participates: A user is mapped to **Chats.** A user participates in a particular chat.

Is Contact: A user is mapped to **User**. A user is contact of another users.

Sends: A user is mapped to **Messages**. A user sends a message to a chat.

Liked by: A user is mapped to **Messages**. A user can like a message.

Disliked by: A user is mapped to **Messages**. A user can dislike a message.

Chats: Relation to maintain a record of all of the chats that have been created in the system

Attributes: cid: chat id. [serial]

chat_name: name of the chat. [varchar(30)]

owner: refers to the user id in the User table of the person that created the chat [integer]

Relationships of **Chats**

Contains: A chat list is mapped with **Messages** to refer to the number of messages the chat has.

Message: refers to every message send in any chat

Attributes: mid: message id. [serial]

text: the body of the message. [varchar(200)]

sender: refers to the user id who sent the message. [integer]

date: the date the messages was sent. [date]

time: the time the message was sent. [time]

chat_sent_to_id: refers to the chat the message was sent to. [integer]

Relationships of **Message**:

Reply: Relationship between two messages, the original message and its reply.

Attributes: reply_mid: id of the reply message. [serial]

replying_to_mid: id of the message that is being replied. [integer]

Contains: A message is mapped to **Hashtags.** A message can contain a particular hashtag.

Hashtags: Relation that states all the hashtags used in every message on the system in any chat. This will be used to manage trending topics in the web application

Attributes: hashid: hashtag id [serial]

hashname: the text after the # in a hashtag [varchar(200)]

Tables Mapping:

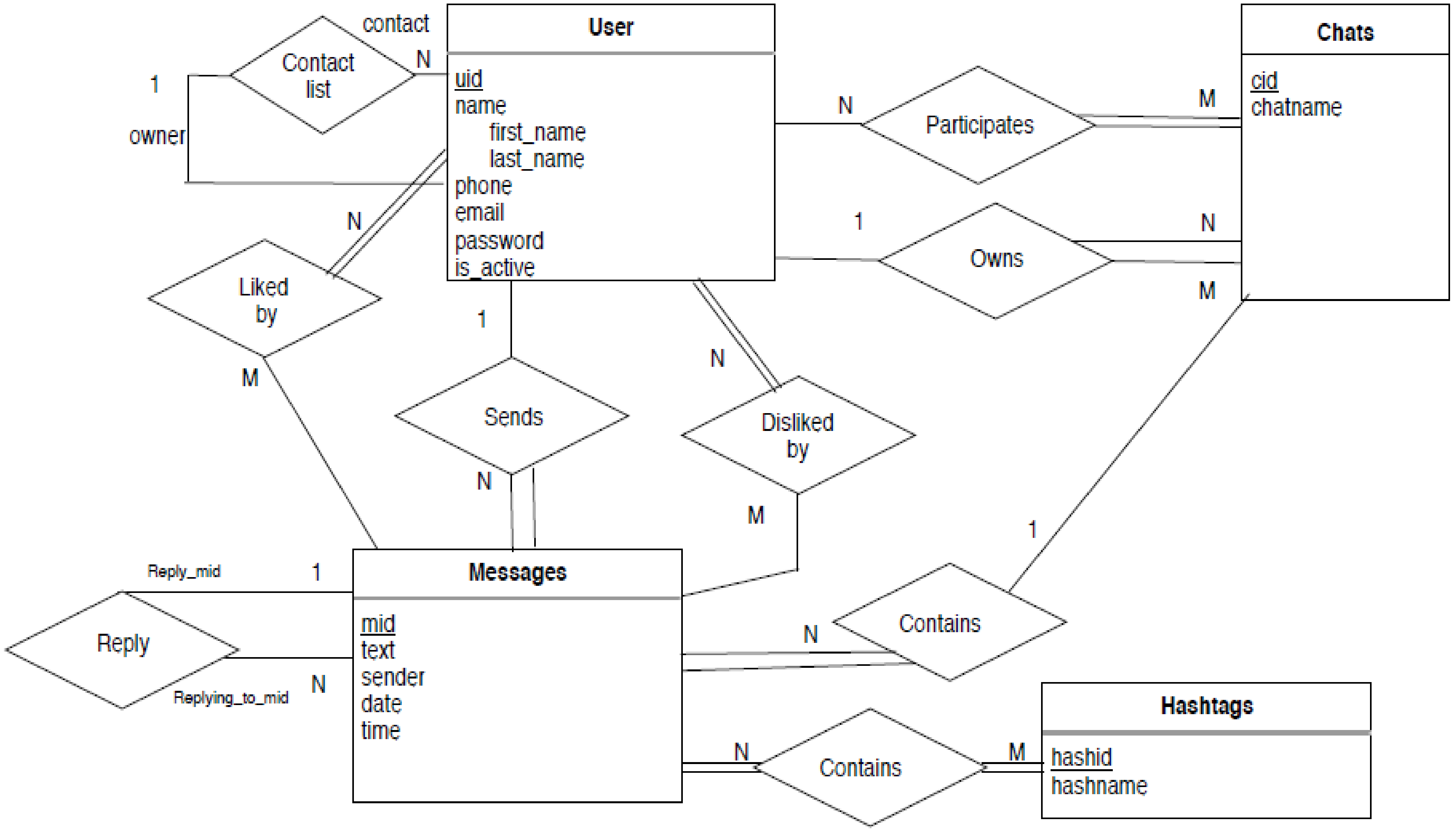
- Entities:

```
CREATE TABLE "user" ( uid SERIAL NOT NULL CONSTRAINT users_pkey PRIMARY KEY, firstname VARCHAR(30)
NOT NULL, lastname VARCHAR(30) NOT NULL, phone BIGINT NOT NULL, email VARCHAR(30) NOT NULL,
password VARCHAR(20) NOT NULL, is_active BIT, username VARCHAR(30) NOT NULL );

CREATE TABLE hashtag ( hid SERIAL NOT NULL CONSTRAINT hashtags_pkey PRIMARY KEY, hashname
VARCHAR(30) );

CREATE TABLE chat ( cid SERIAL NOT NULL CONSTRAINT chat_pkey PRIMARY KEY, chatname VARCHAR(30),
owner INTEGER CONSTRAINT chat_owner_fkey REFERENCES "user" );

CREATE TABLE message ( mid SERIAL NOT NULL CONSTRAINT message_pkey PRIMARY KEY, cid INTEGER
CONSTRAINT cid REFERENCES chat, uid INTEGER CONSTRAINT uid REFERENCES "user", time TIMESTAMP, text
VARCHAR(200) [] );
```

- Relations:

```
 CREATE TABLE contactlist ( uid INTEGER NOT NULL CONSTRAINT uid REFERENCES "user", contact INTEGER
NOT NULL CONSTRAINT contact REFERENCES "user", CONSTRAINT clid PRIMARY KEY (uid, contact) );

CREATE TABLE containhash ( hid INTEGER NOT NULL CONSTRAINT hid REFERENCES hashtag, mid INTEGER NOT
NULL CONSTRAINT mid REFERENCES message, CONSTRAINT chid PRIMARY KEY (hid, mid) );

 CREATE TABLE dislike ( uid INTEGER NOT NULL CONSTRAINT uid REFERENCES "user", mid INTEGER NOT NULL
CONSTRAINT mid REFERENCES message, CONSTRAINT did PRIMARY KEY (uid, mid) );

CREATE TABLE "like" ( uid INTEGER NOT NULL CONSTRAINT uid REFERENCES "user", mid INTEGER NOT NULL
CONSTRAINT mid REFERENCES message, CONSTRAINT lid PRIMARY KEY (uid, mid) );

CREATE TABLE participateschat ( uid INTEGER NOT NULL CONSTRAINT participates_uid_fkey REFERENCES
"user", cid INTEGER NOT NULL CONSTRAINT participates_cid_fkey REFERENCES chat, CONSTRAINT
participates_pkey PRIMARY KEY (uid, cid) );

 CREATE TABLE reply ( mid INTEGER NOT NULL CONSTRAINT mid REFERENCES message, reply INTEGER NOT
NULL CONSTRAINT reply REFERENCES message, CONSTRAINT rid PRIMARY KEY (mid, reply) );
```

Messaging App-ER


User
uid
name
first_name
last_name
phone
email
password
is_active
Chats
cid
chatname
Messages
mid
text
sender
date
time
Hashtags
hashid
hashname
Contact list
contact
N
1
owner
Participates
N
M
Owns
1
N
M
Liked by
N
M
Sends
1
N
Disliked by
N
M
Reply
Reply_mid
1
Replying_to_mid
N
Contains
N
1
Contains
N
M