UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Phase Report

By:
Kelvin García Muñiz (kelvin.garcia5@upr.edu)
Carlos J. Ayala Amorós (carlos.ayala11@upr.edu)
Melvin Malavé Sánchez (melvin.malave@upr.edu)

Dr. Manuel Rodríguez Martínez
ICOM 5016 – Introduction to Database Systems

Due Date
April 8, 2019

GramChat

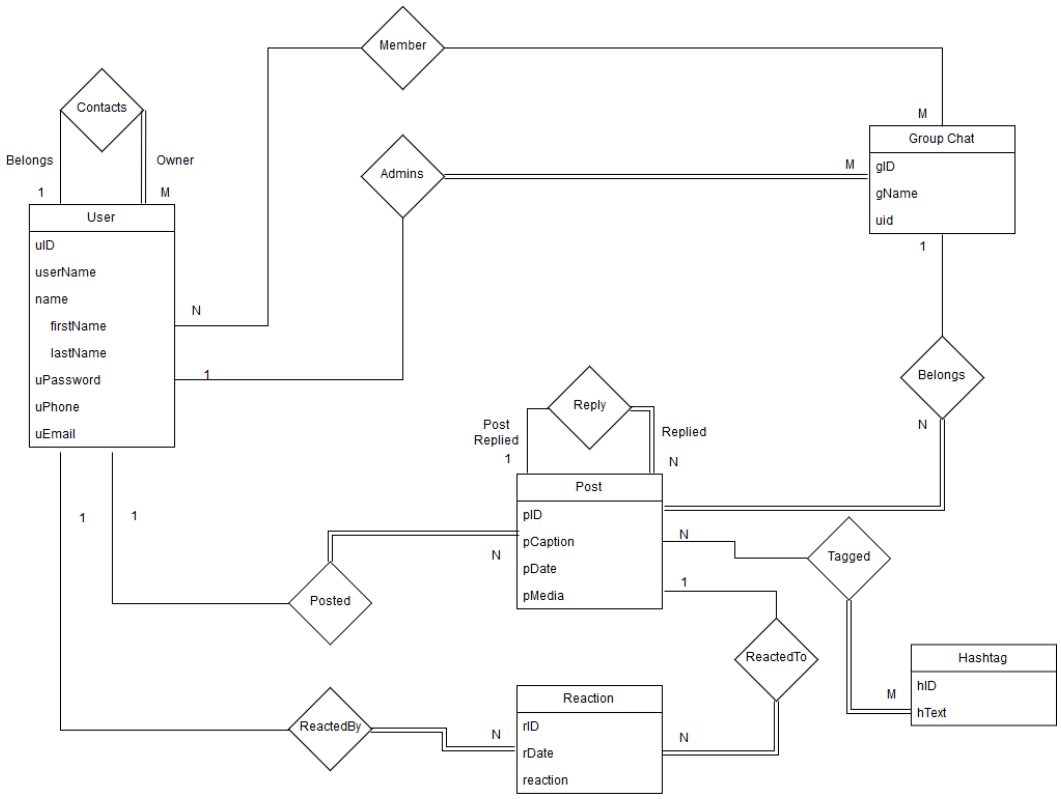
The GramChat API is based on a database application for photo messaging in a social context, like Instagram but based on chat groups. The data in the application is managed by a relational database system and exposed to the client applications through a REST API.

Project Packages

The application is organized in three broad layers:

1. Main - the main app module takes care to setup the routes for the REST API and calling the proper handler objects to process the request.
2. Handlers - the handler modules take care of implementing the logic of each REST call. In this sense, a handler is a Facade for accessing a given operation on a data collection. Each object handles a particular type of request for a data collection (e.g. Parts). The handlers rely upon the Data Access Objects (DAOs) to exact data from the database. The handlers encode the responses to the client with JSON and provide the appropriate HTTP response code.
3. DAOs - the Data Access Objects (DAOs) take care of moving data in and out of the database engine by making SQL queries and wrapping the results in the objects and object list of appropriate types.

Project ER



Understanding the ER

User

The user of the application. After registration, the user will be able to access different groups, post and messages as well as create its own as well as access to a contact list. Registration consists of username, name (first and last), password, phone, email, birthday.

- uID - unique integer used to identify a user (to be used by the system)
- userName - unique name proposed by the user to identify (to be used by the user/system)
- name - first name and last name of each user
- uPassword - variable to log in the user's account
- uPhone - user's phone number
- uEmail - user's email Users possess a contacts list in which they can add or remove other users by their phone or email.

Group Chat

In order to see posts and messages, a user must be part of at least 1 group. A group can only have 1 admin but is not limited in the number of members. An admin must be part of the group it administrates. Inside a group a user will be able to post an see the posts of every member in the group.

- gID - unique integer used to identify a group chat
- gName - group chat's name
- uid - id corresponding to the group chat's administrator

Post

A post is a multimedia file (image or video) with a caption and hashtags that will be shared within a group. A post can only have 1 user but a user can post multiple times. A post can be tagged with hashtags. A reply is a post without multimedia. A reply to a post is a comment. A comment can be replied, but a reply cannot be replied.

- pID - unique integer used to identify a post
- pCaption - text that the user wants to share with the other members of a targeted group chat
- pDate - date of the post
- pMedia - picture that the user wants to share with the other members of a targeted group chat (exclusive to posts)

Reaction

A like or dislike in a post. A user can only react once per post.

- rID - unique integer used to identify a reaction
- rDate - date of the reaction
- reaction - like or dislike of a post

Hashtag

A tag on a post. A message can have many tags. These hashtags will be displayed on the trending tab (to be implemented) if used by many users within the group.

- hID - unique integer used to identify a hashtag
- hText - the hashtag used to tag the post

Mapping

Entity tables

Users table

- *create table users(uid serial primary key, username varchar(25), first_name varchar(15), last_name varchar(20), upassword varchar(20), uphone varchar(10), uemail varchar(30));*

Chats table

- *create table chats(cid serial primary key, cname varchar(50), user_id integer references users(uid));*

Posts table

- *create table posts(pid serial primary key, pcaption varchar(140), pdate char(10), pmedia varchar(200), uid integer references users(uid), cid integer references chats(cid));*

Hashtags table

- *create table hashtags(hid serial primary key, htext varchar(50));*

Reactions table

- *create table reactions(rid serial primary key, rdate char(10), reaction varchar(7), pid integer references posts(pid), uid integer references users(uid));*

Relational Tables

Tagged table

- *create table tagged(pid integer references posts(pid), hid integer references hashtags(hid), primary key(pid, hid));*

Contacts table

- *create table contacts(user_id integer references users(uid), contact integer references users(uid), *primary key(user_id, contact));*

Member table

- *create table member(uid integer references users(uid), cid integer references chats(cid), primary key(uid, cid));*

Reply table

- *create table reply(post_id integer references posts(pid), rid integer references posts(pid), primary key(post_id, rid));*