Fancy title here

Harsh Damania, Christophe Faucon, Nicolas Salhuana, Tsung-Yen Yu

***Abstract*—With today's technology, there are a number of different smartphone apps out there that try to be an all in one social network, and many that play around with the users location to provide data. We take this concept of combining niche location based social networking services, and add a twist with dynamic location data. Users are classified in to zones based on where other app users are right now and can thus communicate with those around, which could end up being for special events.**

***Keywords*—social networking, location services, clustering**

I. Introduction

IN today's day and age where technology keeps changing and new ideas are born on a daily basis, there are still a lot of holes left to plug in various areas of computing. There has been a tremendous success of mobile apps in the areas of social networking and location based services, and there have been quite a few that combine these 2 disciplines. Our project provides a little a twist to these 2 disciplines to make it different from other apps, and make it better for people on the go.

A. Project Description

Our project is location based social network. There can a lot be done with a users location, and thus we chose to use this location data to group a user within a zone with other app users. The size of the zone is determined by the number of other app users around. While in a zone, the user can then send messages to others within the same zone and discuss and what's happening near by. This app would help users understand if there is something special on nearby, and also help understand the events in the zone.

B. The Need

The main use of the app would be for the social networking and getting to know what is going on around you at a given time. There have been a number of other mobile apps that have gained traction lately that relate to social networking and location based services, but none quite have a good blend of the 2 that can be used on the go.

The use of a location service where a user is automatically allocated to a zone they are in based on "hotspots". This way a user who is new to the area can instantly tell what is going on at the moment based on where other app users are and see if there is anything special going on.

Another noteworthy use of the app is for getting a live feed of the current event in a "hotspot". For example, for a college football game, there could be a large number of users in around the stadium that form a zone and everyone within that zone can talk and discuss about the game on that thread. Users are stored within a zone 24 hours after they have left it if they wanted to discuss the thread for a little while after the event too.

II. Related Work

III. Goals

For our project, we had 3 main goals that we wanted to accomplish

- Ability for the user to be assigned a zone automatically based on location: We wanted to make sure that the user would be allocated into their zone based on where the other app users were. This further determined zones for future users to connect to. This was the heart of the project.
- Users should be able send messages to others in their zones: The entire point of being allocated in a zone is for the users to see what others around them are discussing. Since this is a part social network, it is integral for the user to communicate with others.
- Setup a centralized server with Zone messages: Our project will rely on a centralized service that a user can GET and POST to. Joining a zone, downloads the messages on to the users device.

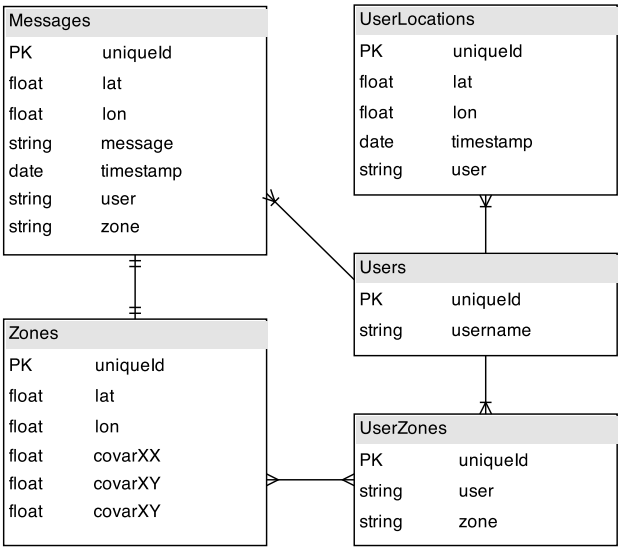

Fig. 1. GeoChat Database structure. The user pings the server regularly with their location which is used to determine which zones they should be a part of, and which messages they can see. Zones behave much like geo-spatial conversation threads, where all messages are public within them. Zone covariance defines the size and shape of the zones.

IV. System Model and Problem Statement

A. System Model

Our system was implemented with a relatively simple database model shown in figure 1. Firebase allows developers to watch one key on a table for instantaneous updates, and for this reason it is critical to ensure that we can indicate whether a message will be relevant to a user with a single key. To this end we reorganized our code so that instead of watching a pair of lat/long co-ordinates we instead watch a particular zone. The zones are calculated by expectation maximization of a gaussian mixture model, discussed in the implementation section.

B. Problem Statement

Terms, definitions, components