Impartial Distance Calculation using Android

Rahul Kulhalli
B.E. Computer
MMCOE, Pune

Tejas Belvalkar
B.E. Computer
MMCOE, Pune

Tejas Khajanchee
B.E. Computer
MMCOE, Pune

ABSTRACT

Often, conventional methods for deciding a common meeting place are quite time consuming, as it involves individual confirmation, argument about deciding the common meeting place and the impartial distance factor. Psychology indicates that a majority of people tend to cancel group meetings due to the feeling that 'others have chosen a meeting place that is comparatively closer to their own homes'. In other words, the meeting place seems 'partial' to some. We hence coin this feeling as the ***partiality factor***.

The motivation of this project, thus, is to introduce the 'impartial factor' in meetings, so as to eradicate partiality whilst choosing a meeting spot. (Please note that this term is self-coined, so as to denote a certain 'degree' of impartiality.)

We shall be creating an Android application that effectively calculates the real time coordinates of all the members of a group, and calculates a smart, scalable and reliable centroid. The meeting places in this restricted vicinity can then be chosen on the basis of voting. Since the locality of the meeting place is strictly restricted by the application's algorithm itself, it will most likely completely eradicate the need of constant compromises and cancellations.

General Terms

Android, Distance calculation, Client-server, Google APIs

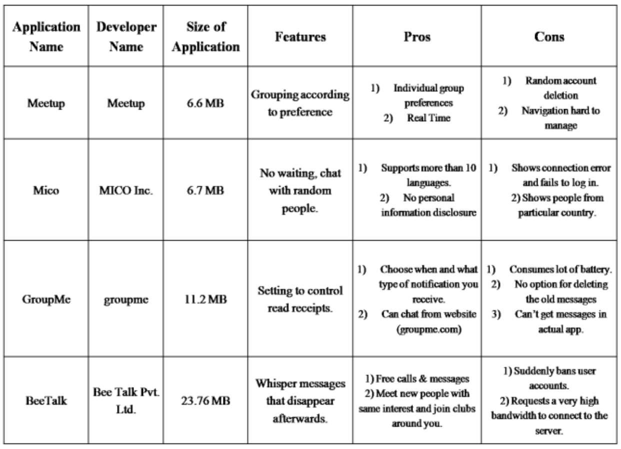
Keywords

Google Cloud Messaging, Google Maps, Google Places, Radar Search, Impartiality, Google APIs, Real-time, Android.

1. INTRODUCTION

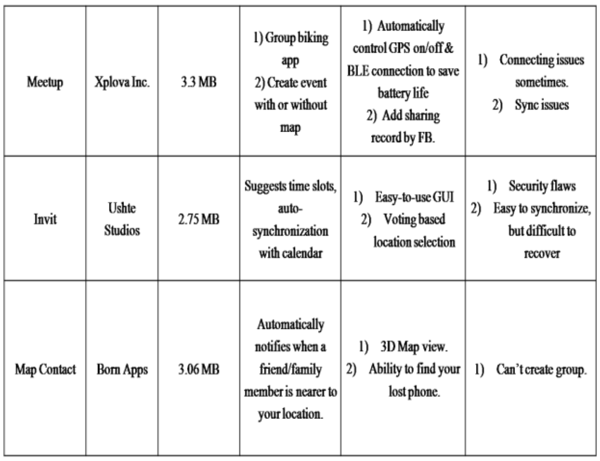
It is a very commonly observed phenomenon when people decide a meeting place, and it is time consuming, if not slightly irritating, as it involves individual confirmation, argument about the personal preferences of meeting places, and the partial distance factor. Psychology has proven that the most common reason that people who cancel meetings give is that of a feeling which somehow makes them believe that the meeting place is partially decided,i.e., they feel that the meeting place is somehow biased. This fact alone goes a long way to prove that if, by any chance, a meeting place which is completely impartial (in terms of distance, not personal preferences) is decided, the rate of people canceling out on social meetingswould decrease to some extent or the other, thus alleviating the stress of not having to attend these meetings. This fact can further prove that the partiality factor plays a direct and major role in the person's decision to attend said meetings.

2. LITERATURE SURVEY



Application Name	Developer Name	Size of Application	Features	Pros	Cons
Meetup	Meetup	6.6 MB	Grouping according to preference	1) Individual group preferences 2) Real Time	1) Random account deletion 2) Navigation hard to manage
Mico	MICO Inc.	6.7 MB	No waiting, chat with random people.	1) Supports more than 10 languages. 2) No personal information disclosure	1) Shows connection error and fails to log in. 2) Shows people from particular country.
GroupMe	groupme	11.2 MB	Setting to control read receipts.	1) Choose when and what type of notification you receive. 2) Can chat from website (groupme.com)	1) Consumes lot of battery. 2) No option for deleting the old messages 3) Can't get messages in actual app.
BeeTalk	Bee Talk Pvt. Ltd.	23.76 MB	Whisper messages that disappear afterwards.	1) Free calls & messages 2) Meet new people with same interest and join clubs around you.	1) Suddenly bans user accounts. 2) Requests a very high bandwidth to connect to the server.

Fig 1: Literature Survey (I)



Meetup	Xplova Inc.	3.3 MB	1) Group biking app 2) Create event with or without map	1) Automatically control GPS on/off & BLE connection to save battery life 2) Add sharing record by FB.	1) Connecting issues sometimes. 2) Sync issues
Invit	Ushte Studios	2.75 MB	Suggests time slots, auto-synchronization with calendar	1) Easy-to-use GUI 2) Voting based location selection	1) Security flaws 2) Easy to synchronize, but difficult to recover
Map Contact	Born Apps	3.06 MB	Automatically notifies when a friend/family member is nearer to your location.	1) 3D Map view. 2) Ability to find your lost phone.	1) Can't create group.

Fig 2: Literature Survey (II)

3. GOALS AND OBJECTIVES

3.1 Objectives

The objectives for this project are:

1) To determine an unbiased meeting location (assuming that cancellations are solely due to distance).

2) To eradicate (to some extent) the partial distance factor.

3) To reduce the time required to decide meeting places. (Almost no time will be required)

4) To maximize the ratio of the actual volume of people attending the meeting to the expected volume of people attending the meeting.

3.2 Objectives

1) After the algorithm finishes processing, it should find actual meeting places.
2) Each member of the group should get the individual route from his/her location to the final meeting place.
3) The members of the group should be able to decide the kind of meeting place. (For e.g., cafes, take-away joints, restaurants, etc.)

4. SYSTEM DESCRIPTION

4.1 Inputs and Outputs

4.1.1 Inputs

The major inputs to the system would be:

1) Set of user coordinates.
2) (Proposed) Vote for the shortlisted locations

4.1.2 Outputs

The major outputs of the system would be:

1) The calculated midpoint
2) The list of shortlisted meeting places
3) Details of the final place
4) Directions from current user location to final meeting place

4.2 Major Data Types

The major data types to be used in this are:

1) ArrayList
2) LatLng
3) HashMap
4) JSON
5) Bundle
6) String
7) Double

4.3 Bounds on I/O

The major bounds on I/O are:

1) Inputs should be of type 'double'.
2) Input co-ordinates should not contain alphabets or special symbols.
3) The ordering of the inputs should always be {Latitude, Longitude}
4) The Latitude value should always be in the range {-90, 90}
5) The Latitude value should always be in the range {-90, 90}

5. ALGORITHMIC DESIGN

The algorithm we have chosen to calculate the midpoint of the incoming Latitudes and Longitudes is that of the **Simple Midpoint algorithm**. The algorithm simply computes the sum of all the latitudes and longitudes, and then divides the sum by the number of Latitudes and Longitudes.

To illustrate:

```
Algorithm ComputeMidpoint returns Location
For i=0 to SIZE do
    Latitude_sum += getLatitude(i);
    Longitude_sum += getLongitude(i);
Latitude_mid = Latitude_sum/SIZE;
Longitude_mid = Longitude_sum/SIZE;
Return new Location(Latitude_mid, Longitude_mid);
```

6. OUTCOMES AND APPLICATIONS

6.1 Project outcomes:

1) Rapid suggestions for meeting points.
2) Hassle-free and autonomous decision making.
3) Removal of the partiality factor.
4) Wider variety of meeting places to choose from.

6.2 Applications

1) Hassle-free decision making.
2) Automated, real time processing.
3) Avoids unnecessary confusions and potential quarrels.
4) Elimination of the partiality factor.

7. ACKNOWLEDGMENTS

We would also like to take this opportunity and thank our internal guide**Prof. Shailaja Jadhav** for giving us all the help and guidance we needed. We are really grateful to her, for she kindly extended all kinds of valuable suggestions, criticisms, workarounds, and strictly monitored our progress, making surewe never derailed off-track.

We are also grateful to **Prof. Harmeet Khanuja**, Head of Computer Engineering Department, Marathwada Mitra Mandal's College of Engineering, for her indispensable support, criticisms and suggestions.

Lastly, we would like to offer our special thanks to **Mr. Vinayak Kulkarni** for making provisions for various resources such as workstations (with all the necessary softwares), a stable internet connection and much more, which proved to be extremely beneficial for our project.

8. REFERENCES

[1] Yaghmour, Karim. *Embedded Android: Porting, Extending, and Customizing*. "O'Reilly Media, Inc.", 2013.

[2] Rogers, Rick, et al. *Android application development: Programming with the Google SDK*. O'Reilly Media, Inc., 2009.

[3] Flores, Huber, and Satish Srirama. "Mobile cloud messaging supported by xmpp primitives." *Proceeding of the fourth ACM workshop on Mobile cloud computing and services*. ACM, 2013.

[4] Koufi, Vassiliki, et al. "An android-enabled mobile framework for ubiquitous access to cloud emergency medical services." *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*. IEEE, 2012.

[5] Jarle, Tor-Morten Grønli, and Gheorghita Ghinea. "Towards cloud to device push messaging on Android: technologies, possibilities and challenges." *International Journal of Communications, Network and System Sciences* 5.12 (2012): 839.

[6] Sappelli, Maya, Suzan Verberne, and Wessel Kraaij. "Recommending personalized touristic sights using google places." *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013.

[7] Van Canneyt, Steven, et al. "Detecting places of interest using social media." *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 2012.

[8] Svennerberg, Gabriel. *Beginning Google Maps API 3*. Apress, 2010.

[9] Mike, James Scott, and John Krumm. "Location-aware computing comes of age." *Computer* 2 (2004): 95-97.