GeoDatings - a Proximity Based Social Application for Android

Mihai Surdeanu
e-Government
University Politehnica of Bucharest
contact@mihaisurdeanu.ro

***Abstract*—Current social applications can offer a lot of functionalities capable of improving user experience and define the right way of having an easier coomunication system through the entire world. The importance of establishing a communication channel between two or more users using low resources and of course, a lower price leads to the need of smartphones. These devices also cover the user's mobility demands. That is the reason why we develop more and more mobile application and why, in this paper, I present you another one - a proximity based social application.**

I. Introduction

Nowadays, the concept of mobile phones suffers a metamorphosis, you are no more using them mainly for voice communication purposes. The mobile phones become powerful devices capable to do advanced computing and multimedia processing, a lot of users start to use smartphones for other things like gaming or social relations. Due to their low cost, their small size, the smartphones are growing in popularity and become a must for everyone who wants to stay connected with others as long as possible.

Over the last decade, technology evolved a lot and plenty of devices are equipped with sensors capable to maintain the environment interaction and to set a connection between user and his current location. At this time, this improvement reduces the effort done by the user to adopt smartphones and also gives to developers the ability to create smart mobile applications. As you probably already know, the software layer was always bounded with the hardware one. When no hardware support is provided, the software capability is limited. Today, our phones let us:

- take photos everywhere we want
- tell us where we are
- recognize our voice commands
- process data in real time

Having a powerful device represents the future, since most valuable actions should be taken with a minimum delay time. It's obvious that taking a photo with one second delay, can alter our desires and in fact, our final result.

In the last years, more experiments were done and all of them concluded that combining elements of gaming, socializing or collecting and sharing information are the key of success to create interesting and profitable mobile applications. So, this paper aims to present an overview and analysis of another research social application for smartphones based on Android as operating system. The main purpose of this software is to help in connecting people closed together and by closeness we refer to both terms of geographical neighborhood and personal sympathy.

II. Related Work

It is a difference between a *proximity social based application* and *a specific social application - like Facebook*. Instead of following people (as we currently do using Facebook application), a proximity application gives you the ability to be followed. Your location is shared to other devices, which will assign you to an ad-hoc network. Following this approach, you are able to select the network by filling in more criterias used to filter your partners.

One example of a proximity application is represented by *Proxxi* that lets you communicate with other users either publicly by commenting on a post, or privately using a safe, anonymous email relay service. Anonymous emails are exchanged in order to protect real user identities.

The Proxxi application is built on following client-side technologies:

- Cordova
- Backbone.js
- RequireJS
- Topcoa

On server side, *Proxxi* is using the following technologies:

- Node.js
- Express
- MongoDB
- CloudFront

The high-level architecture diagram fill in with all those technologies is illustrated in Fig. 1.

Another application that activates in the same domain with *Proxxi* is *Nudge Nudge*. Between these two applications are a lot of similarities but also some important differences. If *Nudge Nudge* requires only Android platforms, *Proxxi* can be used over multiple platforms - iOS and Android. However, *Nudge Nudge* provides more features like profile retrieval (using Facebook), chat session management or remote server interaction. Both applications have some performance issues, because the geographic coordinates are updated every time the user position is changed. This fact associated with a client - server architectural model (where server is not replicated)

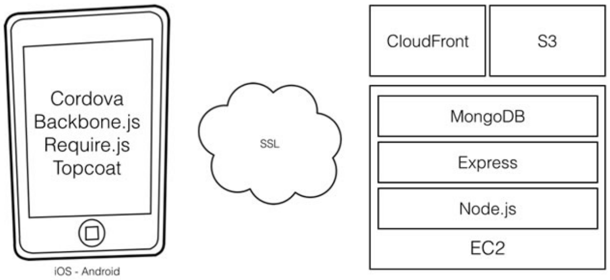


Fig. 1. Proxxi's architecture

means that our server will be fully loaded almost every time. The simplest solution to fix this is to have more server entities capable to stay synchronized and to work together.

Besides another issue with these kind of applications is the users privacy. Sometimes application protocol does not ensure the users identity to remain unrevealed.

III. Architecture

The GeoDatings architecture model, exemplified in Fig. 2, is very similar with a classical server-client prototype. It is not a perfect correlation between these two models, more differences are highlighted by the fact that is an interposed entity - *Google Cloud Messaging*, which allows sending, receiving and keeping the data persistance between clients and servers. The following advantages lead us to integrate this service into our project:

- first of all, this service handles the message queue and delivers these messages to the target Android application
- secondly, is free of charge

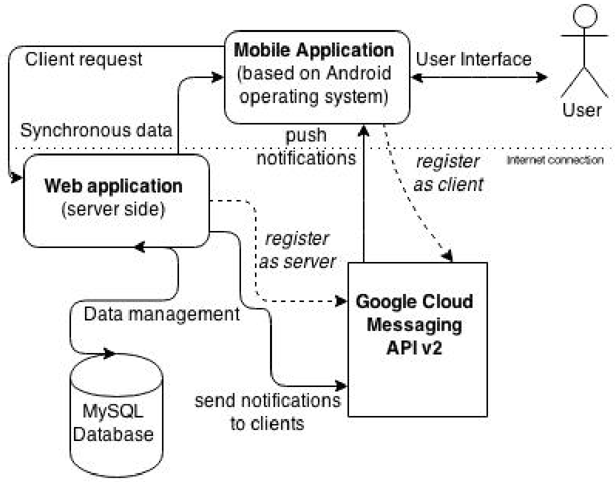


Fig. 2. System architecture

The client side is totally represented by a single mobile application, designed for devices based on Android platforms and the only purpose is to exchange data with the final customer. In turn of services this application is integrated with two other external online services Facebook and Google Maps.

Let's talk about the entity which is the brain of the whole project: the server. The importance of this entity is given due to the data's filtering and providing information that reaches to the customer.

Server is nothing else than a web product, which provides a RESTful type of interface, capable of granting access to MySQL database in an easy and secure way for our clients. This product is written in PHP programming language as back-end layer. Other functions included in this product are the mechanisms of authentication and hiding the user's information. The security it's a priority. The protocol ensure that user can select the message recipients and filters the received messages according to some given attributes. Two kind of messages can be received by the Android devices: synchronous and asynchronous. The synchronous messages are sent as a response to the client's request, and these are received directly from the server. The asynchronous messages are routed through the Google Cloud Messaging.

In the following paragraphs we are going to explain more thoroughly the process of interconnection described, the technologies used and the motivation behind their use.

A. Communication between entities

The client-server communication requires access to the internet and it's based on the HTTP protocol. The client can initiate the connection and interogate the database using GET requests, the feedback from the server to these requests will be an encoded JSON message.

B. Google Maps

Google Maps is another web service and technology provided by Google which is embedded in current application in order to have more information about dating event locations over a map. Setting up Google Maps is quite simple since there are guidelines about how to use and how to integrate *Google Maps Android API version 2*.

C. Facebook

Facebook is a social tool that connects people with friends and others who work, study and live around. Instead of having another custom login mechanism, *Facebook Login* is a real benefit since it's already a worldwide website. Also, having the opportunity to share information and get more details about friends made the choice to be easier.

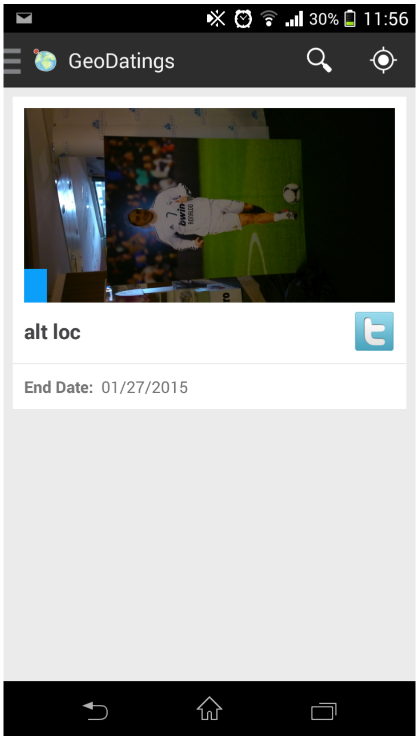
Returning back to the subject of this paper, the final goals of this product are to match the application requirement with system resources - you don't have to worry about your low system resources if you need a decent solution; and to assure the users privacy.

IV. Implementation

This section helps you to understand how Geodatings application was implemented and of course, what features are exposed to user. As you already know, there is a client-server architecture. The client is represented by a pure Android application targeted for Android version 4.3 (Jelly Bean) and the server side is maintained by a PHP application. At this point, the server consists in one class with multiple methods implemented to let user (through a standard interface) to obtain, add new events or to locate them using Google Maps.

To distinguish between two or more Android devices, each device manufacturer is identified by a unique vendor ID, called *device ID*. In our case, this is the element used to identify a client on server side. Also, if you are logged to Facebook, the username can be used as a secondary element after device ID. There is no security mechanism implemented to ensure a client authentication process on server. For a real application, no security means a big gap.

An internet connection is required for everyone who wants to use this application. If someone launches the application without a Wi-Fi or 3G/4G connection then a proper message notification will appear on screen, otherwise user will be able to see main activity - similar with the one illustrated in Fig. 3. From main activy you can easily access other important sections by using the main menu, implemented as a navigation drawer. The user can bring the navigation drawer onto the screen by swiping from the left edge of the screen or by touching the application icon on the action bar.



alt loc

End Date: 01/27/2015

Fig. 3. Main Android activity

As you can see, on the main screen are displayed latest Geodatings events which are still available (are not expired). Those events are not filtred by some other rules like: hide events that are not near current user location (obtained by a GPS module). This behavior is a little bit different from the one provided by another activity - *Around You*. Around You, uses a Google Maps map to show all events centered in a point - current user location. By default, the radius is set to 10 kilometers and is editable from a dedicated setting.

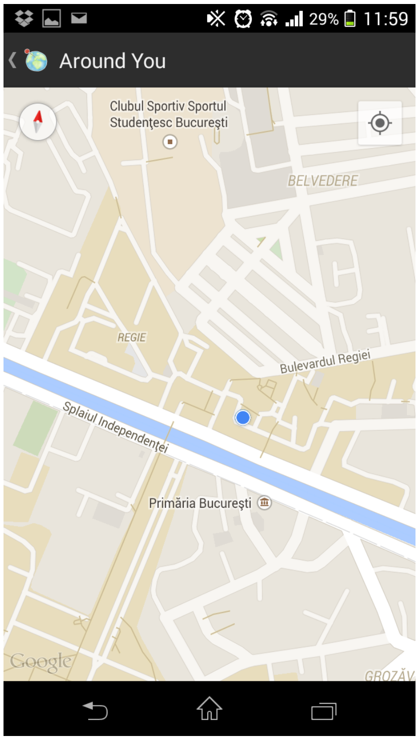
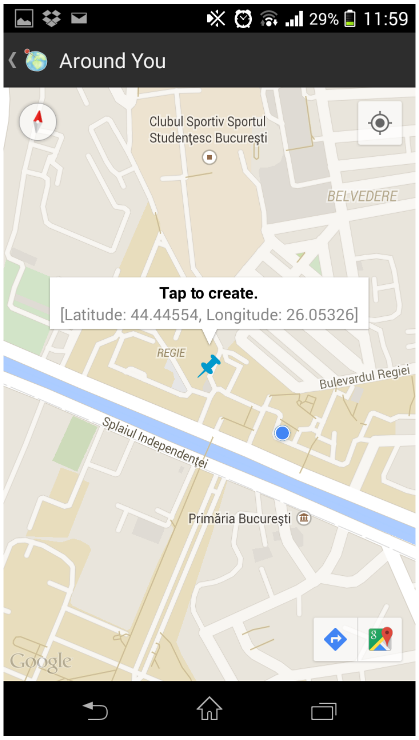


Fig. 4. Around You activity

Around activity is accesible after clicking on the right icon found in Android action bar (near search button). When no events can be found around your location, you'll be able to see only your current location represented with a blue circle - Fig. 4. From here, you can add a custom event by having a long finger tap over the location desired. After doing this, a new event will appear on the screen (Fig. 5 is an example). At this point, the event is not created yet, the server doesn't anything about it. The marker added can be moved to another location using drag & drop. When you established the event position, you can proceed to the next step. Please tap inside the information box and wait a moment, until a special form will became visible.

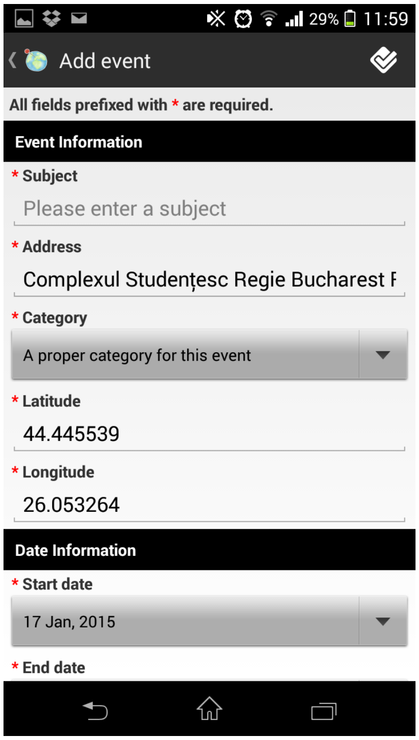
On the new form seen, there are more input fields filled automatically - address, latitude and longitude are some examples. Immediately after filling all fields required an user can submit his event by pressing the button from top-right side. Due to the fact that the form let you to upload an image and this image is not compressed or no size restrictions are defined, the submitting process will take a while. Congrats! A new event was added (Fig. 7). Using Google Cloud Messaging layer all clients registered in system will get notifications about the new entry.

Moreover, Android application include settings that allow users to modify application features and behaviors. For example, some of them allow users to login using Facebook as platform, or specify whether notifications are enabled or



Tap to create.
[Latitude: 44.44554, Longitude: 26.05326]

Fig. 5. Creating a new event - Step 1



All fields prefixed with * are required.

Event Information

* Subject
Please enter a subject

* Address
Complexul Studenţesc Regie Bucharest F

* Category
A proper category for this event

* Latitude
44.445539

* Longitude
26.053264

Date Information

* Start date
17 Jan, 2015

* End date

Fig. 6. Creating a new event - Step 2

specify which server and API version are used.

V. Results

The current development state don't let me to publish this application. As I already said, there are still more security mechanisms to be fullfied. In addition, current architecture defines a bootleneck between clients and our server. We can have multiple clients but only one server. If the number of server requests are huge than probably things will go crazy. The simplest solution here you be to replicate the server and to have more of them registered as Google Could Messaging servers.

Regarding the usability, after showing this application to my room colleagues, they had some issues on how to use. So, as a feature improvement, more overlays (with help instructions) should be added. This approach will avoid the necessity to read manuals which are annoying most of the time. They also told me that I have an intuitive application and relative easy to use once someone shows them all the features.

VI. Conclusions

In conclusion, I've conducted an interesting project where more elements were combined to create a proximity based social application over Android devices. Three important online services were integrated to ensure different capabilities. The first on is represented by Facebook - used to obtain the friend list of an user; the second one was Google Maps - useful to categorize events after their geo position; and the last one was Google Cloud Messaging - essential to ensure sending notifications without the intervention of the main server. There are still more improvements to be done before publishing it to Google Play Store. The most important one is related with the security side which is missing now. I'll continue to work on this and after few months I want everyone to be able to use it.

References

[1] T. Ravindran and A. Y.K Chua, *Whither Mobile Social Application Prototypes?*, Nanyang Technological University, Singapore.

[2] A. Bujari and N. Miotto, *Nudge Nudge: a Proximity Based Social Application*, Universit degli Studi di Padova, Italy.

[3] E. Novak and Q. Li, *Near-Pri: Private, Proximity Based Location Sharing*, College of William and Mary.

[4] C. Coenraets, *Proxxi - A Proximity-Based Social App*, 2014.

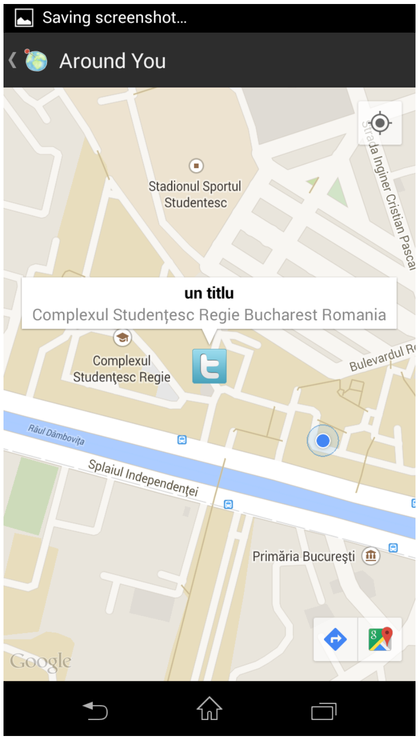
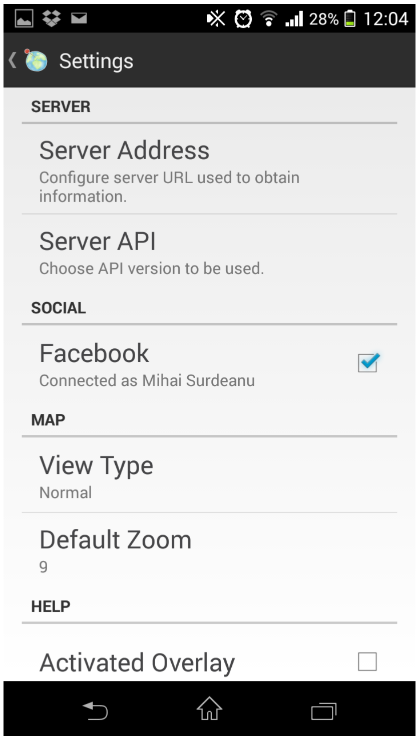

Saving screenshot...
Around You
Stadionul Sportul Studentesc
Strada Inginer Cristian Pascal
un titlu
Complexul Studențesc Regie Bucharest Romania
Complexul Studențesc Regie
Bulevardul R
Râul Dâmbovița
Splaiul Independenței
Primăria București
Google

Fig. 7. The event created



Settings

SERVER

Server Address
Configure server URL used to obtain information.

Server API
Choose API version to be used.

SOCIAL

Facebook ☑
Connected as Mihai Surdeanu

MAP

View Type
Normal

Default Zoom
9

HELP

Activated Overlay □

Fig. 8. Settings