Doctor Search Service. Using Geo2tag LBS Platform for Mobile Healthcare Applications

Ivan Bezyazychnyy, Kirill Krinkin,
Open Source and Linux Lab
Saint-Peterburg, Russia
ivan.bezyazychnyy@gmail.com, kirill.krinkin@fruct.org

Abstract

Mobile healthcare is a rapidly developing area. The significance of solutions in this area cannot be underrated because they directly influence the health of people that is undoubtedly the most important thing in their life. This article presents the idea of one of the solutions in this area: doctor search service. The main idea of the service is that the patient can send SOS message from his mobile device in the emergency situation. A doctor who happens to be not far from this patient will receive the signal about this event. He is able to look at the location of this event on the map using his mobile device. The article describes the service from user point of view as well as technical details of service implementation.

Index Terms: Mobile Healthcare, Geo2tag, Location Based Services

I. Introduction

Mobile phones are widespread very much. And their functions are not limited by the possibility to make calls and send and receive SMS. A lot of phones nowadays have the ability to get access to the Internet and to determine its location. Almost every person in developed countries has at least one cell phone and a lot of these phones have both of the mentioned functions. A lot of resources show that mobile phones are becoming popular in developing countries also.

Mobile phones find their application in different areas of our life. One of these areas is healthcare. There are a lot of solutions that include but not limited by telemetry systems (the ability to treat or make consulting over distance), mobile health monitoring systems, geospatial information systems for public health emergency etc [1, 2, 3]. The main purpose of this paper is to present one more mobile healthcare solution that utilizes network and positioning capabilities of mobile devices. It is a service for searching doctor who can be occasionally found not far from the suffering patient. This solution is not the replacement of emergency aid. The main purpose is to increase the probability that the first aid will be provided as soon as possible. The situation can be imagined that it takes about ten minutes or even more for ambulance to get to the patient but the doctor with appropriate skills might be walking round the corner.

A. Use cases

As soon as the main idea is formulated let’s point the use cases that determine the functions of the service more precisely. The use cases are depicted in Fig. 1 and described below.

The most important use case is “SOS sending”. It describes the sending of request for help. The use case implementation should be as simple as possible for the patient so the patient is able to send request for help very fast. The patient only clicks on “SOS” button and the mobile client does all the work: it determines position and sends the message to server. While mobile client handles the SOS sending it shows the message that it is performing the sending. After message is sent mobile client shows the corresponding message. From now it is the work of the server to handle this SOS message and of other mobile clients to retrieve the new message from the server. It is evident that the patient might be able to click SOS button only once so the mobile client should not ask the patient whether to send message again in case of network or other types of errors and should start sending all over again until message is sent successfully.

All SOS messages sent to the server represent events. Each event keeps geographical position, time and may keep other information. So the second important use case of the service is to deliver the information about events to doctors who are in the predetermined radius from the patient having sent the request. The actors of this use case are doctors and other persons who can give first aid before the arrival of an ambulance. The actors might even be ordinary people who can help the patient to call ambulance. For example, the patient may appear in other country and may not know how to call it. When the information about new event is delivered to doctor's mobile client should notify about event using mobile device capabilities such as speakers, vibrator, light, etc. When the doctor gets the notification he opens the client and sees the map. The event is displayed on this map. He can also click on the image of the event and get extra information about the patient.

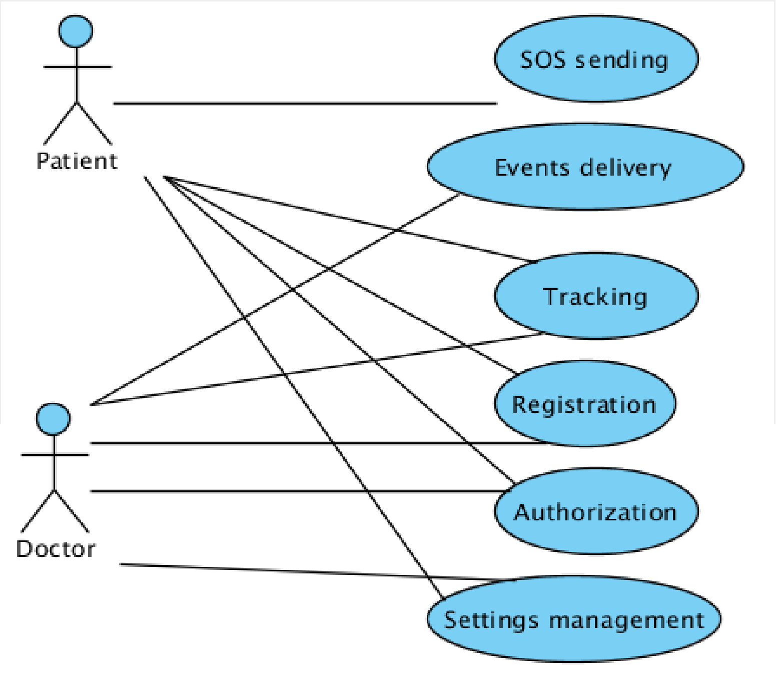


Fig. 1. Use cases

This two use cases describe the main idea of the service: patient sends SOS message as a request for help and the nearest doctor is notified about the event. Another important use case is tracking users' positions. The tracking of patient may help to find them if something happened with them and they were not able to send the SOS message. The tracking of doctors may be used on server side to find the nearest doctor for the patient.

In addition to the use cases described above mobile client should also support other secondary use cases. Each user should be able to register to the service using mobile client on his mobile device. User inputs login and password and clicks the registration button. The request is handled at server side and user receives the result of his registration request. To sign in to service through the mobile client the user inputs login and password and clicks the “Sign in” button. The information is sent to server which verifies the credentials and answers whether information is correct. If it is true user gets access to the main functions of the mobile client described above.

And the last use case is the settings management where user can specify personal information such as description of himself and service information such as address of the server.

II. Main part

A. Service architecture overview

The service is organized as a client-server architecture (Figure 2). The server part is based on Geo2tag LBS Platform [4] and has an open API for clients. The clients are mobile devices of patients and doctors.

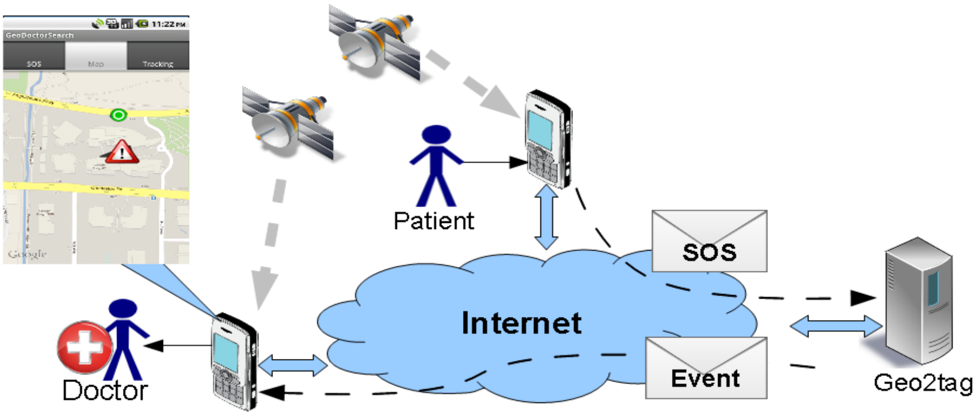


Fig. 2. Client-server architecture of the service

Geo2tag is an open source project. Geo2tag provides the API which can be applied for different Location Based services. A developer of the Location Based service installs the Geo2tag at the server side and decides how the platform will be used. He should understand the logic how the platform handles its content and decide how it will be mapped to data of concrete service. Also a developer should understand interface offered by the platform and implement clients that can operate with platforms’ content as it is supposed by the service logic.

The Geo2tag platform’s content are next entities: Users, Channels, Tags. Users describe the users of the service. Tags are the units of information which might have description, geographic coordinates, url, time. The channels are the way of classification of tags. Users subscribe to channels, read and write tags from them. Also Geo2tag supports time and spatial filters. More information about development applications for Geo2tag LBS Platform can be found in [5-7] and on the project wiki[1].

The first step of Location Based service developer is to decide how these entities will be used in the service. Almost always users entities are just users of the service. The platform supports registration and authentication. The usage of tags and channels entities is often not so clear. It is the task for service developer to decide how tags and channels will be used. Figure 3 depicts how they are used for the Doctor Search service purpose. As shown in the figure platform keeps one Events channel. Patients who are in an emergency situation send tags which represent SOS messages to this channel. Doctors request messages (tags) from this channel using filter and receive information about events near them. Also there are channels for each user. They have the same names as users' logins. Users periodically send messages with their location to their channels. That is how the tracking is implemented.

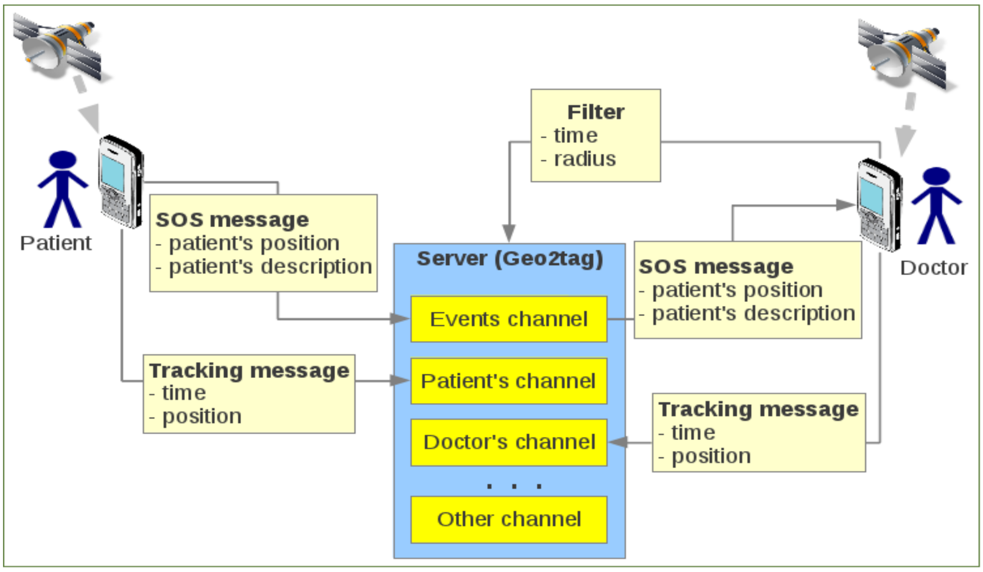


Fig. 3. Geo2tag adaptation

The Geo2tag offers RESTfull API. The interaction with the platform is occurred by means of HTTP requests and responses. They contain data in JSON format. Examples of requests are registration of new users and authentication, subscription to channels and unsubscription, sending tags and requesting tags from subscribed channels, adding of new channels. With request for tags from subscribed channels the concrete channel can be specified and the time interval of tags can be chosen. Also request for tags can be accompanied with spatial filters. In Doctor Search service case the Circle spatial filter is used that allows to request information about new events that are in specified radius from doctor requesting for them.

The open API allows to write clients for different mobile devices. Mobile devices should have network capabilities and the function of geographical position determination. Mobile application which is run on such device should use this capabilities and be able to perform HTTP requests, receive HTTP responses and handle data in JSON format to support the functionality described by use cases.

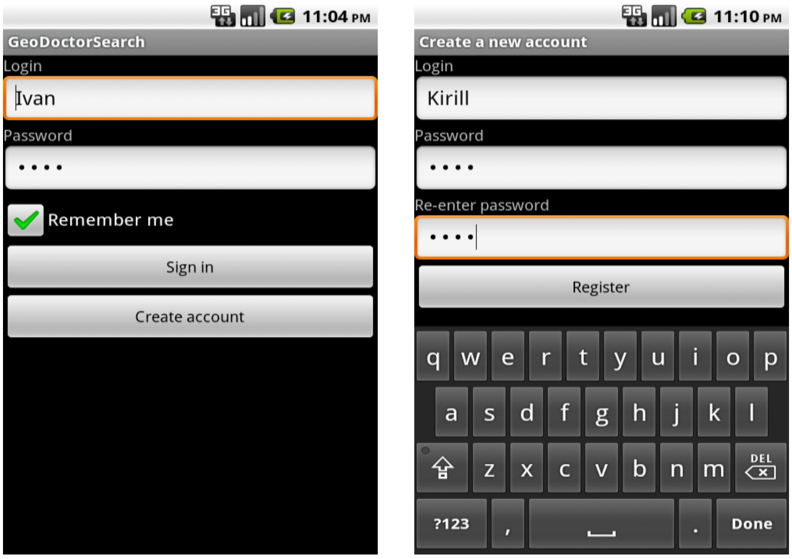
[1] http://geo2tag.org/index.php/Geo2tag:Open_Source_LBS_Platform

B. Current status

The architecture presented in previous section assumes creation of different clients for different mobile platforms. That is achievable by means of the API provided by the Geo2tag platform. The application should be able to handle the HTTP requests and responses that contain the data presented in JSON format. So the typical mobile application should include a module to interact with the Geo2tag platform and a set of UI forms that provide the functionality described in use cases.

Now two mobile client prototypes are implemented: Qt mobile client and Android mobile client. For the server part the test version of the Geo2tag is used that is located at http://tracks.osll.spb.ru.

When the mobile client is started the login form is shown (Figure 4, left screenshot). User is able to login or to create new account. If he chooses to create new account he clicks the corresponding button. During the registration (Figure 4, right screenshot) the account is created, the channel with his name is also create. User is subscribed to Events channel to receive events and to his channel to write tags with his position for tracking.



Login: Ivan
Password: ••••
☑ Remember me

Login: Kirill
Password: ••••
Re-enter password: ••••

Fig. 4. Login and registration UI in Android client.

After user is logged in he has an access to main use cases which are presented in corresponding tabs. One of this use cases is SOS sending. To send the SOS message user should click the SOS button in the first tab of the main form (Figure 5, left screenshot). The idea was to provide very simple implementation of this use case from user point of view. User just clicks one button and the application determines the position and sends the SOS message until message is delivered to server.

The symmetrical use case of SOS sending is events delivery. When the device gets information about event it plays the alert and shows the event on the map (Figure 5, right screenshot). The doctor finds event on the map and can get extra information about the patient clicking on the image of the event.

One of this use case is tracking of the position. User can see whether tracking is active and is able to start/stop the tracking. Also he can see the log of coordinates sent to the server which approves the service is running correctly.

Using menu the user can get settings widget. In the current implementation settings include server address and user's description. User description is transmitted with the SOS message and may keep user's address and health information, for example, his diagnosis.

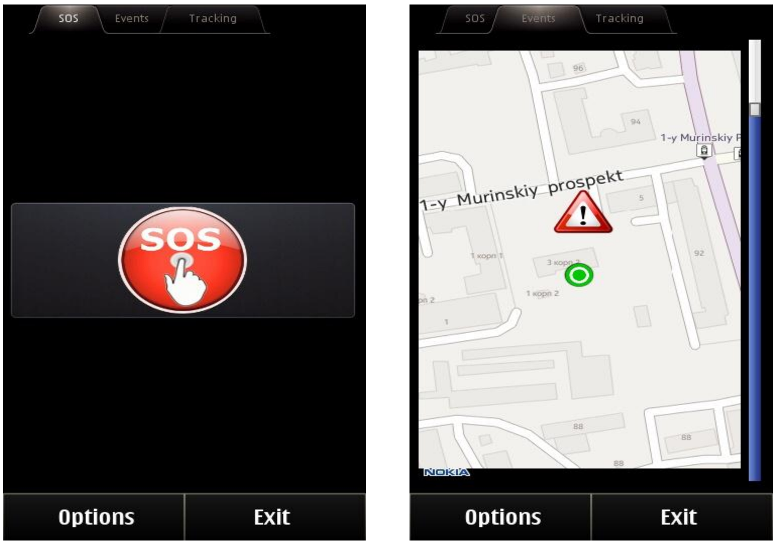


Fig. 5. SOS button and map with events in Qt client

The Qt application had been test on Nokia E7 device and the Android application has been tested in the Android emulator.

III. Conclusion

The service for searching doctors near the suffering patient using mobile devices has been presented. The main use cases of this service have been described. The technical implementation details have been briefly discussed. Our future work is connected with the growth of the service in two directions: new functionality and new mobile platforms support.

In the first case our plans concern the additional functions of the service. One of them is improving the notification system. Server part should be smarter and be able to notify doctors also by SMS. Also the server part should handle SOS requests and should be able to call an ambulance to the scene. Another idea of service extending is differentiation of doctors and patients. Patients may have different diagnosis and doctors may have different specializations. So the task of service is to analyze these factors and to notify the most appropriate doctor.

In the second case plans include further extending of the supported mobile clients. For example, implementing the Java ME mobile client will allow supporting low price Nokia devices which are widespread and affordable.

Acknowledgment

The authors would like to thank Finnish Russian University Cooperation in Telecommunication Program and especially Mobile Healthcare (m-health) working group for provided support, greatest ideas and fruitful cooperation.

REFERENCES

[1] “An overview of Recent End-to-End Wireless Medical Video Telemedicine Systems using 3G” / A. Panayides [and oth.] // *32nd Annual International Conference of the IEEE EMBS* Buenos Aires, Argentina, August 31 - September 4, 2010 . - Buenos Aires, Argentina, 2010, pp. 1045-1048.

[2] “Yu-Chi Wu. A mobile health monitoring system using RFID ring-type pulse sensor “// Yu-C hi Wu, Pei-Fen Chen, Zhi-Huang Hu // *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing* , 2009, pp. 317-322.

[3] “Design and Implementation of Mobile GeoSpatial Information System for Public Health Emergency” / Weixing Wang [and oth.], *Wireless Communications, Networking and Mobile Computing*, 2009. WiCom '09. *5th International Conference*. - Beijing, China, 2009.

[4] Geo2tag LBS Platform – http://geo2tag.org

[5] V. Romanikhin, M. Zaslavsky “Spacial Filters For Geo2tag LBS Platform,” *The 11th Conference of Open Innovations Association FRUCT*, Saint-Petersburg, Russia 23-27 April 2012.

[6] Zaslavsky “Location Based Platform for Public Transport Monitoring System”, *9th Conference of Open Innovations Framework Program FRUCT*, Petrozavodsk 25-29 April 2011.

[7] Bezyazychnyy, K. Krinkin, M. Zaslavskiy, S. Balandin, Y. Koucheravy “Geo2Tag Implementation for MAEMO”, *7th Conference of Open Innovations Framework Program FRUCT*, Saint-Petersburg, Russia 26-30 April 2010.