

Location-based Services in Public Buildings: LIST for N900

**Denis Bepalov, Boris Vasilev, Iliya Mironov,
Gregory Serebryakov, Sergey Sosnin**

Nizhny Novgorod State University
Gagarin Ave, 23, Nizhny Novgorod, Russia
wl.maemo@yandex.ru

Abstract

The LIST mobile service is based on the idea that the existing (already formed) offline local community, united by a common territory, common affairs, common procedures and common interests needs tools that make it possible for their members to interact effectively with the infrastructure object (i.e. the university campus) and with each other. The service allows community members to get interesting local content and to exchange location-based information connected with off-line services and facilities. Also, this tool can simultaneously be an interface between the providers of online services and products aimed exactly at the target audience and this audience.

Spatial localization of the community allows to organize provision of services, sales of goods from suppliers of these services and sellers of good sufficiently with minimal cost, for example, in the form of local shops and delivery services.

The first version of the service is intended for students, staff and visitors of university campuses. It allows them to get on the mobile phone almost any information regarding to the University:

- Orientation on campus and inside buildings
- Find places and people on campus
- Find and order books in the library
- Access to the schedule, the ads, etc.

Users of the service have an opportunity to participate in the mobile social network, uniting them in the local community and allowing them to generate relevant and interesting local content. For local information service at the university campus the target audience is primarily youths and target vendors - vendors that provide goods and services targeted at young people.

The service will provide access to diverse local content associated with the campus UNN (University of Nizhny Novgorod). Most of the content is unique and not available through other channels (UNN sites, social networks).

Index Terms: Location-based, Mobile service, LBS, Social networks.

I. INTRODUCTION

Users of the service have an opportunity to participate in the mobile social network, uniting them in the local community and allowing them to generate relevant and interesting local content. For local information service at the university campus the target audience is primarily youths and target vendors - vendors that provide goods and services targeted at young people.

The service will provide access to diverse local content associated with the campus UNN (University of Nizhni Novgorod). Most of the content is unique and not available through other channels (UNN sites, social networks).

The content available through the service consists of two main parts: content created by administrators of service (interactive map of the campus and plans for buildings, information about places and people in the UNN, timetables, official announcements from the university

administration and the deans' offices) and the user-generated contents. Users have an opportunity to comment on any information placed in the service, to post messages on their mobile mini-sites and message boards, to communicate in thematic chats, to create location-linked notes. In this model we'll get self-updating and self-growing service.

The concept of service: at the university, business center, shopping mall, park etc. it is often needed:

- To find the right room, the person object or service;
- To understand where you are in the building (especially if you come to this place for the first time),
- To obtain the information on the objects' local geography,
- Just want to talk to someone who, like you, fated a long list with many links lead to information to find associated with this place.

Find answers online, using Yandex and Google via your mobile browser is not a good idea, because you will get an information that is not needed in a particular place and at the current time, or which is scattered across many documents and Web sites to download and view. So the user will waste a lot of time and money. A lot of the information in modern web was designed for leisurely analysis at your home or office and only occasionally can be helpful in field conditions. Ironically, the old paper guidebooks, handbooks, and, simply, a notebook in a pocket in these situations can be beneficial to the Internet.

II. MAIN PART

A. *System Architecture Fundamentals*

LIST is built on the basis of client-server architecture. The client is implemented for J2ME and Qt platform, the main server (LIST) is implemented with Ruby on Rails. Client and server exchange data is carried out via XML-files.

The main server (LIST) contains all the local content and user information. The interaction with the server the client behaves like a browser - displays the resulting page, which are described by using a specific XML markup language.

The functionality of client:

- Displaying XML pages generated on the server as an application screens
- Support for instant messaging protocol IRC (Internet Relay Chat)
- Navigation and positioning with the use of raster maps on the basis OpenStreetMaps
- Display maps of the campus and plans of premises using vector maps
- Creating notes linked with the specified position on the map and with the date
- Using the camera phone for snapshots and sending them to the server
- Including additional services into the application

B. *Processing XML pages on the client*

The application uses a markup language screens based on XML. When the user produces some event (choice of menu items, etc.) the client sends a HTTP request to the server, receives a response with a description of the next screen and displays it. Client supports both GET and POST methods, latter is used for transmitting photos.

Markup language supports several types of screens:

- menu screen
- entry forms' screen
- map screen

- chat screen

These screen can content different types of content elements, e.g. buttons, text fields, groups of radiobuttons and so on. When the client recieves a new page it tries to math it's type with suitable translator, which will produce a list of data elements, storing all information of the screen. Then this list is sended to widget builder which generates widgets associated with each data element. So, the system in this part is very flexible and appearance of the application can be changed very easily.

C. Maps

Client supports raster maps. We are not using vector maps due to rendering of large maps (plots of cities) on the client side is not possible due to the lack of performance of the platform, so we use raster maps server CloudMade, obtained on the basis of vector maps OpenStreetMaps ([2]), nevertheless it can easily be changed to to another sevice, using Mercator projection of the globe ([1], [3]) and working with ellipsoidal approxiamtion of the Earth form (e.g. GoogleMaps).

Raster maps have a smaller number of interactive elements (in our implementation they are layer of LIST objects and points with attached information, e.g. photos, notes and so on) compared with vector maps, and increase the amount of network traffic, but nevertheless they are a reasonable solution for this platform.

On Fig. 1 you can see how maps look in work:

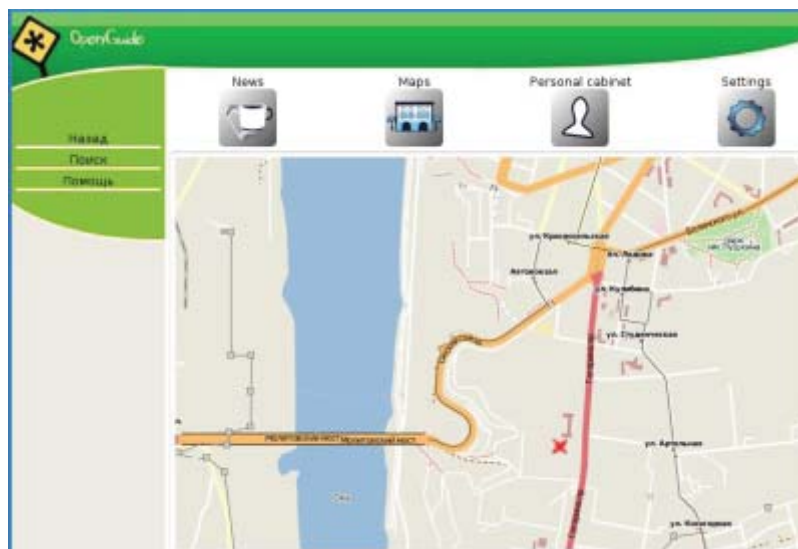


Fig. 1. Raster OpenStreetMaps

To reduce the amount of network traffic the following features were realised:

- parts of maps are downloaded asynchronously
- parts of map are cached
- when the map is moved client "updates" it — only new parts are downloaded

D. Caching screens

In this implementation, to reduce the amount of traffic screens are cached at the session level (until you restart the application). Screens' XML downloaded from server are wrote into

the local filesystem with creating a record in cache database(we use SQLite for it) storing the name of the file and it's validity mark. On closing connection with the database client checks all records and remove each one with appropriate flags.

E. Interface

For this version the interface has been changed dramatically due to be more convenient with the user. Also exists a desktop version of LIST, so the UI is the similar on both desktop and N900 internet tablet. On Fig. 2 you can see a start screen for authorized users - "Personal cabinet".



Fig. 2. Personal cabinet

Unauthorized users begin the work with the application from the page, which is presented on Fig. 3.



Fig. 3. Main menu

III. CONCLUSION

Coming to the conclusion, I should say that we have reached almost all aims, which were set on the stage of developing the concept of the project. Now the service LIST successfully works on the base of the Nizny Novgorod's State University (UNN) and will be started at some more object soon.

What about nearest plans, we mean to convert LIST from simply mobile service to the platform for family of different applications which are united with location-based and personal-oriented abilities. Also we are intending to provide work with offline packages of information and GPS.

REFERENCES

- [1] Mercator projection http://en.wikipedia.org/wiki/Mercator_projection
- [2] CloudMade API <http://developers.cloudmade.com/projects/tiles/documents>
- [3] J.P. Snyder, "Map Projections - A Working Manual," *U.S. Geological Survey Professional Paper 1395*, United States Government Printing Office, Washington, D.C., 1987