# Developing a location based service for mobile phones

Henrik Back, Rickard Engberg, Johan Herdegard, Mikael Laaksonen, Rickard Lejon, Daniel Nibon, Viveka Sjöblom, Gustav Tano, Jiying Tian, Ming Zhao, Hao Zhang, Song Yuan

Department of Information Technology, Uppsala University
P.O. Box 337, S-751 05 Uppsala, Sweden.
Email: {heba7429,rien7181,johe0261,mila7681,rile6711,dani1903, visj8680,guta8921,jiti4019,mizh0695,hazh1255,soyu6003}@student.uu.se

**Abstract.** Mobile POI (Point Of Interest) is a location based service which is community driven. The service consist of a server system and a client application. The client application are developed for mobile phones. The main feature is the ability for users to share information about geographical positions. The project started September 2006 and finished Januari 2007. In this paper we describe the general design, environment and the result of Project Mobile Point of Interest.

## 1 Introduction

There are lots of positioning services on the market. However non of them has become really popular. The goal of this project was to create a community driven positioning service where information about geographical positions are shared. The users submit information by adding new POIs or commenting existing ones on their mobile phone. The POIs consist of text, images and user comments. There is also a web interface where the users can add, edit and remove POIs or some information about them.

The Mobile POI system consists of a server that is serving a client with user maps and location information. The client application are targeted for mobile phones and is programmed for the J2ME (Java 2 Micro Edition) environment. The majority of the mobile phones sold today support this technology. The server are Linux based and run Erlang [3].

The Mobile POI system is designed to allow multiple technologies to find a location. Among these technologies are GPS (Global Positioning System), operator assisted positioning and manual user positioning. For the communication between the server and the client a new protocol has been defined.

Mobile POI (Point Of Interest) is student project by fourth year computer science students at Uppsala University. This paper will describe the idea behind this project and give an overview of the system. The rest of the paper are organized as follows: In section 2 there is a short introduction to some keywords in this project. Section 3 presents the idea of Mobile POI and section 4 gives an overview of the system. Testing and Evaluation are described in section 5 and the paper is concluded in section 6.

## 2 Preliminaries

### J2ME

Java 2 Micro Edition (J2ME) [2] is a set of technologies and specifications developed for small device like mobile phones. J2ME uses a subset of J2SE (Java 2 Standard Edition) [4] components such as a smaller virtual machine and leaner APIs. These are similar to J2SE but with highly reduced functionality. Although it has some unique features not present in J2SE.

### Erlang

Erlang [3] is a concurrent functional programming language and runtime system. The main strength of Erlang is its ability of concurrency. It has small but powerful set of primitives to create processes and communicate between them. Erlang needs a run-time environment since it is by default an interpreted language. Open Telecom Platform [2] is a library developed by Ericsson AB that defines a large portion of Erlangs behavior, since much of Erlang development is based on this library.

### Point of Interest

A Point of Interest (POI) is a geograpical position that is holding some information. A POI can hold information about different kind of objects, for example restaurants, pubs or parks. The information consist of position, descriptions, images and user comments. The POIs are added and edited by the users of this system.

### GPRS

General Packet Radio Services [1] is a standard for packet data transmission over GSM. It makes the mobile devices able to access the Internet from any location as long as it has a sufficient signal strength from the GSM Base Station. The speed of the connectivity is relatively low, but in the 3G network the transfer rate is much higher. The communication between a mobile phone and server on the Internet can be done by using a socket connection.

### Cell-ID

Cell-ID is an identifier for the base stations in the GSM network. It can be used for mobile positioning, i.e. finding a users location. The location of the user can be approximated if the base stations position is known. This method is not very precise and it is depending on the number of base stations in the search area. The accuracy of approximation the users position vary. If the base stations are densely concentrated a more accurate position can be calculated.

# 3 Mobile Point of Interest

## 3.1 Idea about POI

The purpose of the Mobile POI project is to create a platform where a set of users can share information about geographical positions between each other. The POIs hold information that consist of descriptions, images, an integer rating and user comments. Each POI belongs to a community and a category. The community is a group of users. Every user can start a new community and choose which other users that are members. Each community has a set of POIs that only its members can see. The category distinguish between different types of POIs (i.e. Restaurants, Pubs, Shops). A category is chosen when the user adds a POI. The purpose of the categories is to let the users filter a set of POIs.

The system has two different interfaces for the user, the mobile phone and the web interface.

## 3.2 Community based

The idea is to have a community where the users themselves share information about POIs between each other, instead of one centralized information provider. The system provides the users with maps, a centralized storage and tools for adding and removing information. Each user has the ability to start new communities or join existing ones. The user can only see POIs that belong to communities where they are members.

## 3.3 The mobile phone

The application on the client side is designed for mobile phones. The client application has a interface for showing maps with POIs. The user can step between different POIs shown at the current map (see Figure 1) and view information about each of them. The set of POIs shown are determined by which community the user belong to and which area of the map that is displayed. When the POIs are displayed the user can filter some of them out by choosing a category.

By clicking on one POI the user come to the information screen (see Figure 3). Users can choose to add a comment or see what other users written about that POI.

At the map interface the user are able to zoom the map, move the map or choose another location to be displayed. The users can also add their own POI, which other users in the community can see. This is done by choosing a location on the map (see Figure 2) with the cursor. Next step is to choose a category, enter a name and a description of the point on the map (see Figure 4). On some phone models there is also an option to add an image from the phones camera. After all information is entered it is sent to the server and the POI can instantly be displayed on the map.

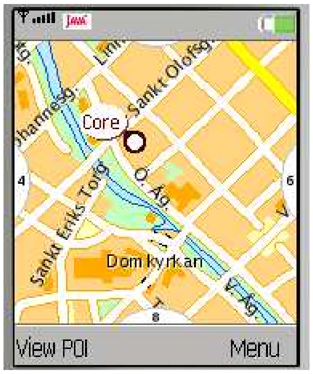


**Fig. 1.** Browse POI Mode

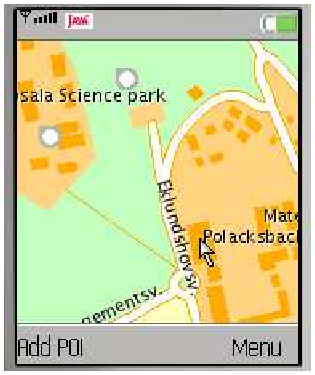
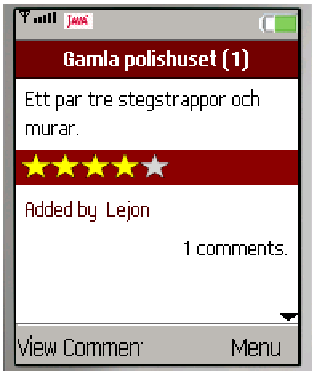


**Fig. 2.** Add POI Mode



Gamla polishuset (1)

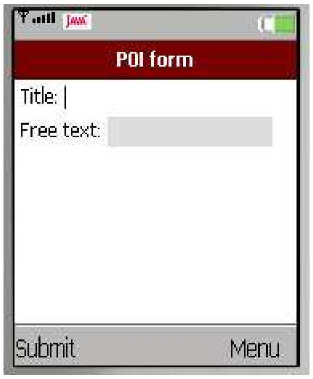
Ett par tre stegstrappor och murar.

★★★★☆

Added by Lejon

1 comments.

**Fig. 3.** View POI Info



POI form

Title: |

Free text:

**Fig. 4.** Add POI Form

### 3.4 The web interface

There are two separate web interfaces one for the end-user and one for the system provider. The two different interfaces are logically separated but use the same database to access common information. The separation of the interfaces facilitate that they run on different IP port numbers. This solution improves security due the possibility to hide the administration tools from public access if required.

The end-user web interface is intended to give the user feedback from the mobile phone community and also to view and edit available information in a simple fashion from the user's perspective. The web interface can be seen as a simple tool to edit posts and can therefore be seen as a complement to the mobile device where it is hard to overview much information (with the limited in- and output devices and traditionally low bandwidth). The prospective web interface is rich in possibilities to be used as a stable commercial platform.

Much underlying database functionality is already generated and more features are possible to implement in the web interface.

The administration interface is developed in parallel with the end-user interface and have initially got the same GUI and functionality plus the added administration functionality. An administrator have permissions to edit posts in the database more extensively than an ordinary end-user.

# 4 System Description

## 4.1 System Overview

The system consists of a mobile phone client and a server system (see Figure 5). The server side consists of a two major databases, communication interfaces to the mobile phone and external sources like the map server. As can be seen on figure 5 the server communicates with an external map server. The map server are provided by and external company. The server side and the client side are developed independently with the communication protocol linking them together. The communication protocol defines what kind of requests the client can send to the server.

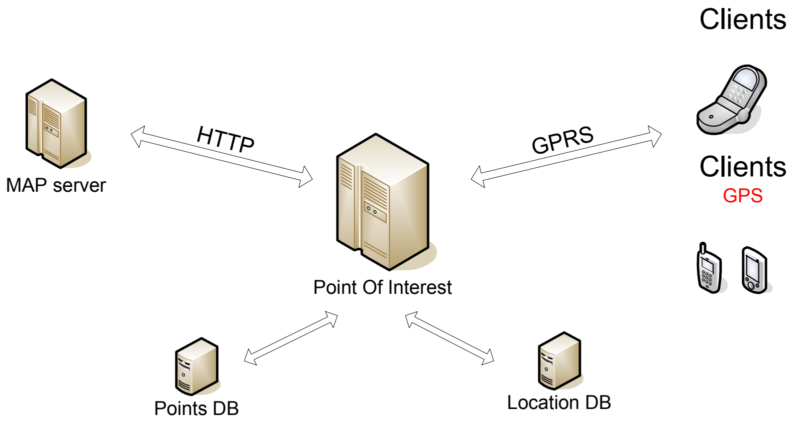


**Fig. 5.** System overview

## 4.2 Server

The server system consists of a database, a web server and a POI server. It is running on a single computer under Ubuntu Linux. There are two ways to access the server, either from a handset through GPRS-socket connection or from a web-browser through HTTP-connection.

The system uses a native database called Mnesia. Mnesia is a distributed Database Management System, appropriate for telecommunication applications and other Erlang applications which require continuous operation and soft real-time properties. What sets Mnesia apart from most other DBMSs is that it is designed with the typical data management problems of telecommunications applications in mind. Mnesia combines many concepts found in traditional databases with concepts found in data management systems for telecommunications applications. Mnesia take the advantages of database transactions, queries and very fast real-time operations. It is possible to configure the degree of fault tolerance (by means of replication) and the ability to reconfigure the system without stopping or suspending it. All the data are also stored in a database.

The web server is running Yaws (Yet Another Web Server), which is written in Erlang. It is easy to configure and run under OTP (Open Telecommunication Platform). One of the advantages of Yaws is its ability to handle many concurrent connections. The web portal contains information both for the administrator and for public users. The database is accessible from the web interface and both the user and the administrator can do operations (i.e. modifying the user profiles, creating communities, categories et cetera).

The POI Server is responsible for handling requests from handsets by GPRS-socket connection. It also has a connection to the remote map server. The connection between the mobile phone and POI Server is stateless and all actions at the server are based on each individual request. All the essential information is therefore built into each packet that the server receives.

The system has two kinds of clients. The mobile phone with GPS and the ones without GPS. Mobil phones with GPS can provide the POI server with an exact GPS position. In this case the POI server talks to the Map server directly and can offer a map around the precise GPS location of client. The second type of client does not have a GPS device. This type mainly relies on the self-learning system we provide or a manual input from the user.

The self-learning system is based on Cell-ID. The Cell-ID are sent to the server by each client where it is available. The idea is that the server shall be able to map each individual Cell-ID to a location. This approximation is not done by triangulation, it is a one dimension positioning algorithm. Clients that know its position and Cell-Id help to make the precision better by sending both. However this feature is not fully developed and not tested in our system due to hardware problems. It is only a small fraction of all mobile phones that support this type of positioning.

### 4.3 Client

The client application is developed for a mobile phone. A typical phone model is the Sony Ericsson W810i. The client application is written in J2ME. Java technologies that are used are MIDP 2.0 [5], CLDC 1.1 [6], J2ME Polish [7]. The communication between server and client is made using sockets over GPRS. We have also defined a protocol for communication to and from the server which is described below.

The system is designed to allow different techniques to find the clients geographical position. Technologies that can be used are GPS, operator assisted location, Cell-ID positioning and manual user input. The GPS can be built in to the mobile phone or can be an external device connected via Bluetooth. If the mobile network operator permits the positioning can be requested from the network. The last option is that the user find his position by zooming in and moving the map. By leaving the positioning option open it allows the system to run on several kinds of hardware on the client side.

The data requested from the server are maps, POIs, categories and comments. The data sent to the server are new POIs, comments and different kind of requests for maps, POIs, comments et cetera.

Portability is one of the major difficulties when programming mobile devices. The main problem is that the handset software has bugs in the implementation of the Java Runtime Engine. Different screen sizes, processor power and memory are also factors to adapt to. A helpful tool in this area is J2ME Polish. With Polish it is possible to select the devices for which the application should be built for.

### 4.4 Communication Protocol

The POI communication protocol was created in cooperation with both server-side and client-side in the POI-project. Due the very limited system resources involved with mobile computing the protocol is as small as possible, considering the data transfer overhead. Small overhead generally needs to compromise with future extensibility. In the POI project the communication protocol is extensively found upon using bit-masking and operating codes (OP codes). Bit-masking and opcodes can be used in subsets of a byte or overlapping several bytes. This technique optimizes the usage of individual bits within the header.

To prevent reinventing the wheel and to easily use existing techniques the POI protocol extends the TCP-protocol (RFC 793), and can benefit of the error correction and received packet ordering. The TCP protocol is supported by sockets in MIDP 2.0 which is supported by most new mobile telephone models. The extending of the TCP protocol is done by sending data as a ordinary TCP-datagram. Once received on the server or client the received data is extracted and the POI-header is read.

The minimal size of the header is eight bytes long, but might be larger depending on the type of message. The POI header always contain a one byte opcode in the first position of the header which telling the purpose of the rest of the package. The second to the fourth byte is defined as a sub-opcode which holding information which differs depending on the opcode set in the first byte. This characteristic give a highly scalable solution to express fourth coming information in only four bytes. The fifth to eight byte in the POI header is always an integer with user ID of a POI user.

# 5 Evaluation and Testing

## 5.1 Project evaluation

The Mobile POI system consists of POI client, POI server and web portal. We have tested its functionality and concluded that it operates functionally. In the requirements specification almost all requirements are met (with some exceptions).

We started out this project with the aim for using Cell-ID based positioning. A system that gets improved over time by a self learning algorithm. The Cell-ID extraction is supported by only a few mobile phones at this time. We only succeeded in extracting Cell-ID on one Nokia phone. Because of the small support for Cell-ID extraction this features became a side track in the project and lost some focus in the end. However we got both the position approximation and the self learning algorithm to work.

## 5.2 Project testing

To develop this application incremental development was used. From the start we planed three milestones, M1, M2 and M3. Each milestone contained an important subset of the requirements from the requirements specification. When the M1 was supposed to be tested there was not much to test, but the development had already started on several of the requirements of M2. Therefore the M1 was delayed and combined with M2. M3 testing went on as expected with about 80% pass. In the very end we added an additional test M3, to verify the latest updates. All tests were conducted after our test specification. The test specification was created to cover all non optional requirement stated in the requirement specification.

# 6 Conclusion

We have described the project Mobile POI - a location based service for mobile phones. The project has met most of the requirements. Some hardware related problems occured in the middle of the project with the Cell-ID extraction. The focus moved from the Cell-Id feature to concentrate on the community service instead. The community service will be the important feature if this project should continue and become a commercial product. The positioning is a secondary thing that could be solved by the mobile network operator or by the use of a GPS. There is also a possibility for the user to enter a position manually by clicking and zooming in on a map.

Developing a mobile application for the first time is very much trial and error programming. For each new feature implemented it requires testing on a mobile phone before continuing. Some features turned out to be hard to get working on some of the platforms. Among them are the networking on Nokia Series 60 phones and Cell-ID extraction on all phones. Since the hardware and the implementation of the J2ME environment differentiate on the phones, a lot of strange errors occur. A lot of these errors are not documented and you have to search for clues in internet forums about J2ME programming.

Despite the fact that our project took another course than first planned, we are very satisfied with the result. That is, a location based service which allows a community of users to share information about geographical locations.

## References

1. [GPRS] *General Packet Radio Service - Wikipedia, the free encyclopedia* http://en.wikipedia.org/wiki/General_Packet_Radio_Service , 2007-1-4
2. [J2ME] *Java 2 Platform, Micro Edition (J2ME) Overview* http://java.sun.com/j2me/overview.html , 2007-1-4
3. [Erlang] *Erlang Programming Language* http://www.erlang.org/doc.html , 2007-1-11
4. [J2SE] *Java 2 Standard Edition* http://java.sun.com/j2se/overview.html , 2007-1-11
5. [MIDP] *Mobile Information Device Profile* http://java.sun.com/products/midp/, 2007-1-11
6. [CLDC] *Connected Limited Device Configuration* http://java.sun.com/products/cldc/ , 2007-1-11
7. [J2ME Polish] *J2ME Polish* http://www.j2mepolish.org/documentation.html , 2007-1-11