

Design and Implementation of Positioning and Navigation System Based on Android

Yuan YuanYuan

School of Computer Science and Engineering
Beihang University
Beijing, China, 15011025247
yysmile109@gmail.com

Zhang LiJun

School of Computer Science and Engineering
Beihang University
Beijing, China, 010-82317611
ljzhang@buaa.edu.cn

Abstract—With the development of mobile Internet, more and more people begin to get convenient service by mobile phones. Obtaining one's current location by GPS positioning or network positioning has become one of the important foundations in most applications of location based service. In this paper, we designed and implemented a personalized positioning and navigation system based on the Android platform. With the combination of GPS positioning and network positioning, and using Google Map API, this system provides the following functions: view the current location, get the navigation route, address query and view historical location records.

Keywords—location; navigation; Android; Google Map

I. INTRODUCTION

In recent years, with the rapid development of mobile Internet, mobile phones are no longer just a tool for communication, more and more people hope to get more useful services by mobile phones. Location Based Service (LBS) is one of the popular applications in the field of mobile Internet. It obtains the location information (geographic coordinates or geodetic coordinates) of the mobile end-user through the wireless communication network (such as the GSM network, CDMA network) of communications carriers or external positioning (such as GPS or network), and then provides users with the appropriate service under the support of GIS (Geographic Information System) platform [1]. With the popularity of smart phones, obtaining one's current position by the GPS positioning or network positioning has become an important foundation of LBS applications. In this paper, we designed and implemented a personalized positioning and navigation system based on the Android platform. Users can get convenient services from this system as described below:

A. Current Location

Users can get real-time positioning results that will be displayed on the map. In addition, users can view the details of current location and upload the location information to the server.

B. Navigation Route

Users can see the navigation route information of the path between the start point and the end point. There are two navigation modes to choose: driving and walking. After

choosing the navigation mode, this system will provide users with the corresponding information of the distance and duration between the start point and end point, and display the navigation route on the map in the end.

C. Address Query

Users can query an address and the query results will be displayed on the map.

D. Historical Record

Users can view their own historical location information records, which are displayed in a list by default. Besides, users can view the historical location information records on the map as well.

II. ANDROID PLATFORM AND GOOGLE MAP

A. Android Platform

Android platform is of open system architecture, with versatile development and debugging environment, but also supports a variety of scalable user experience, which has optimized graphics systems, rich media support and a very powerful browser. It enables reuse and replacement of components and an efficient database support and support various wireless communication means. It uses a Dalvik virtual machine heavily optimized for mobile devices [2]. Android is a software stack for mobile devices that includes an operating system, middleware, and key applications. The Android SDK provides the tools and libraries necessary to begin developing applications that run on Android-powered devices [3]. As shown in Fig. 1 [4], Android architecture includes the following sections [5]:

1) *Applications*: A set of core applications are on the top level, including messages, web browser, contacts, etc. All apps are written using the Java programming language.

2) *Application Framework*: Developers have full access to the framework APIs used by the core applications. The application architecture is designed to simplify the reusing of all components. This mechanism allows every component to be replaced by the user. Underlying all applications is a set of services and systems, including a rich and extensible set of Activities that can be used to build an application, including

TextView, ListView, Spinner, Button, MapView and so on. Content Provider enables applications to access data from other applications (such as Contacts), or to share their own data. Resource Manager makes non-code resources accessible from code. Notification Manager enables all applications to show custom alerts in the upper status bar. Activity Manager manages the life of each application and provides a useful navigation backtrack.

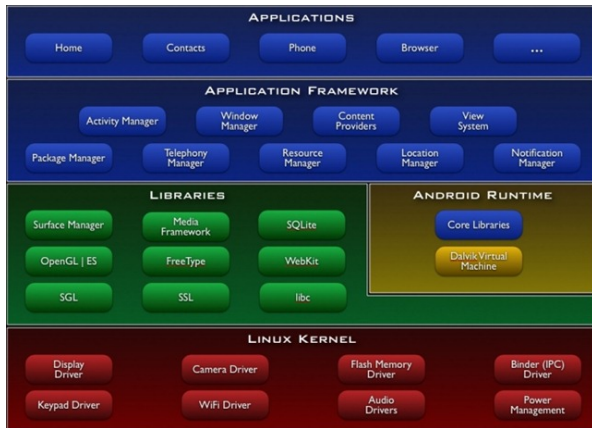


Figure 1. Android architecture.

3) *Libraries*: Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed in Figure 1.

4) *Android Runtime*: Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process given by the OS, and owns its own instance of the Dalvik virtual machine. The Dalvik VM executes files in the .dex (Dalvik Executable) format which was optimized for minimal CPU and memory usage. The Virtual Machine is register-based, and runs classes compiled by a Java language compiler that have been transformed at compile-time into the .dex format using the "dx" tool, that are shipped with the SDK.

5) *Linux Kernel*: Android relies on Linux (Kernel version 2.6) for core system services such as memory management, process management, network stack, security, and driver model. The core also acts as a hardware abstraction layer between the applications and all the hardware.

B. Google Map

Google Map is an electronic map service provided by Google. Before using the Google Map service in Android applications, applying for a Google Map Android API Key is pre-requisite to the development of the corresponding map function [6]. With class MapActivity, MapView and Google Map API, we can control the map easily on Android. Class LocationManager and LocationProvider are important in

Android location services. LocationManager provides methods of getting the system location, therefore, positioning, tracking and approaching prompts are easy to accomplish with LocationManager. LocationProvider defines a method of providing location service (for example, provided by GPS or provided through a network, etc.). In order to get a suitable LocationProvider, we can use class Criteria to set some conditions we preferred. In addition, class Geocoder can complete the conversion between the address information and the latitude and longitude coordinates [7].

Most of the smart mobile phones provide location services. Currently, Android platform supports GPS positioning and network positioning. GPS positioning is completed by GPS satellite, it has a high accuracy and wide coverage, however, it often fails when blocked by large buildings and always consumes more power, so it is suitable for outdoor positioning. Mobile network positioning is completed through the mobile operator's cellular mobile base station or WiFi access point, the location accuracy is lower but is suitable for indoor positioning.

III. SYSTEM DESIGN AND IMPLEMENTATION

A. System Design

The architecture of the entire system is based on C/S. The server is mainly responsible for the processing of corresponding data and responding to the requests of Android client. The Android client is primarily responsible for the implementations of UI and business processing logic. In this paper, we mainly describe the design and implementation of the Android client.

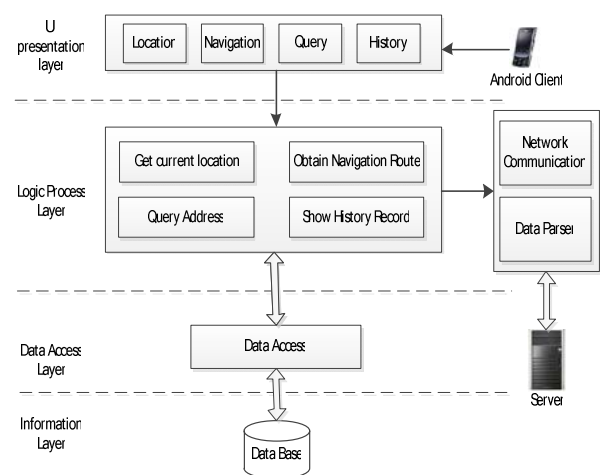


Figure 2. System architecture.

As is shown in Fig. 2, the system is divided into four layers as described below:

1) *UI Presentation Layer*: Be responsible for implementing friendly Android client user interface that including location, navigation, query, history and so on with

different kinds of Android view component and Activity component.

2) *Logic Process Layer*: Receive instructions and data from UI presentation layer, then process the logic business including obtaining current location, obtaining navigation route, querying address, showing history record, network communication, data parser and so on. Finally, return the results to UI presentation layer.

3) *Data Access Layer*: Be used by logic process layer to access data in the database.

4) *Information Layer*: Database. In order to save resources of Android client, user's data will be stored in the database of server.

B. Implementation

The implementation of each module is described in the following:

1) *Current Location Module*: The flow of current location is shown in Fig. 3. First, check whether the GPS of Android client is turned on, if not, a dialog box will be popped up to alert user to turn on the GPS. After that, system tries to get user's current location by GPS positioning with function `getLastKnownLocation()`, if the value of location is null, use network positioning instead. At the same time, add a `LocationListener` to monitor the change of location. When location is successfully obtained, get the latitude, longitude and detailed street address information by class `Geocoder`, then display location information on the map with `MapView` and `Overlay`. Finally, user can view the location details, and upload the location information to the server.

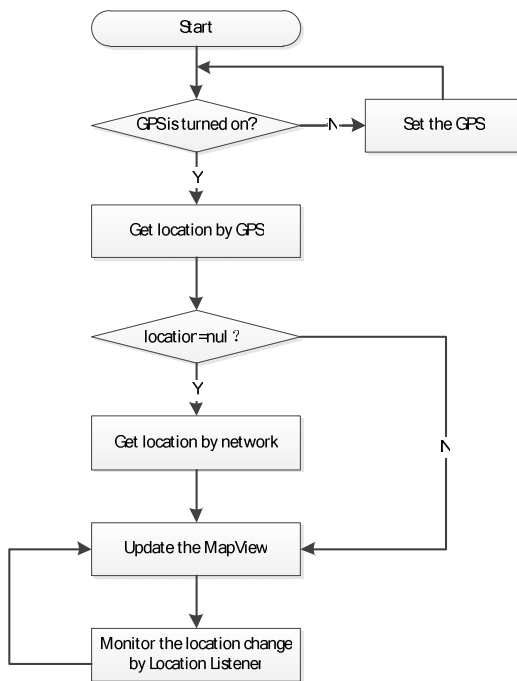


Figure 3. Flow of current location.

2) *Navigation Route Module*: The flow of navigation is shown in Fig. 4. First of all, gather start point, end point and navigation mode that user input, then send a Http request for JSON format response packet with these parameters to the Google Map server by network communication module. Afterwards, Google Map server will respond a corresponding JSON format packet [8]. Parse the response packet with JSON parser and get the navigation route information that including key points, distance and duration between the starting point and the end point. Finally display the distance and duration in a `TextView` and draw the navigation route on the `MapView` with custom `Overlay`.

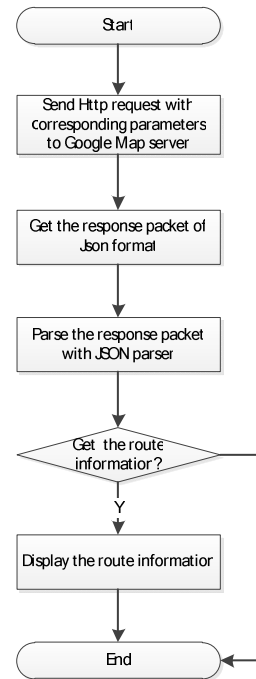


Figure 4. Flow of navigation.

3) *Address Query Module*: Convert the address information that users input into geographic coordinate information by class `Geocoder`, then mark the location on the `MapView` with custom `Overlay`.

4) *Historical Record Module*: Send Http request to the server for relevant historical location records information that previously uploaded through the network communication module, the historical records will be displayed in a `ListView`. In addition, user can also view the record item in the map mode.

IV. SYSTEM TEST RESULT

We test the function of this system on Google Nexus S of Android 4.0.3. The screenshots shown below are part of the test results.

Fig. 5 shows the result of current location on the map (left) and the corresponding detailed location information (right).

Besides, users can switch to satellite map by clicking the satellite button.

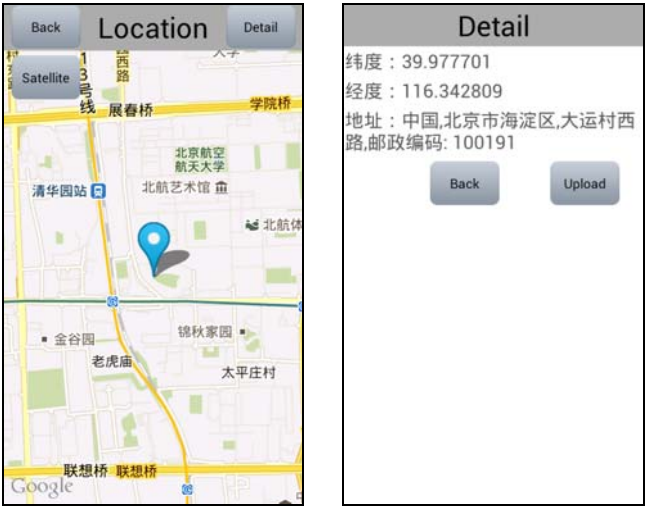


Figure 5. Current location.

Fig. 6 shows the navigation function. As the left screenshot shown, we input Beihang University and Beijing University of Technology as the start point and end point respectively, then select driving mode to get the navigation result. The screenshot on the right shows the navigation result that including distance, duration and navigation route which is drew on the map.

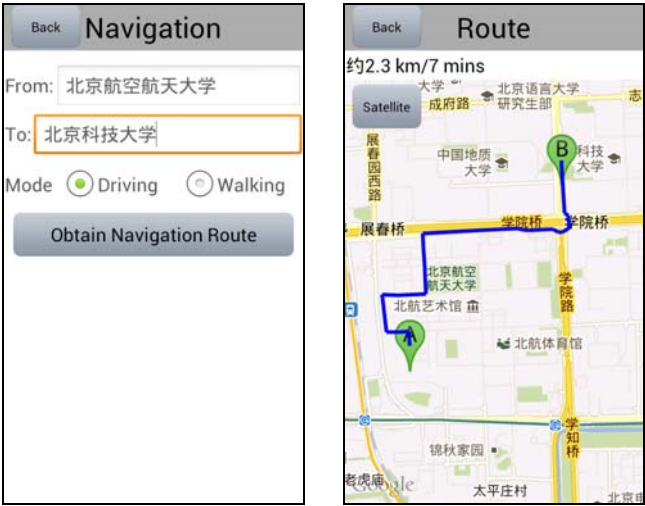


Figure 6. Navigation route.

The left screenshot of Fig. 7 shows the function of address query. We input an address and click query button, the result is shown on the map.

The right screenshot of Fig. 7 shows the historical records in a ListView. Each record includes the location and time

information. User can view the corresponding location on the map by clicking the record item.



Figure 7. Address query and historical records.

V. CONCLUSION

This positioning and navigation system consists of the following functions: viewing the current location, getting the navigation route, getting address query and viewing historical location records. The user interface is friendly and each module works well, various LBS applications can be implemented based on this system. We hope to extend the functionality of the system by adding some other features such as friends in the future.

ACKNOWLEDGMENT

This paper is funded by Graduate Innovative Practice Fund of Beihang University. The corresponding project number of this fund is YCSJ-02-8.

REFERENCES

- [1] Q Ren, M. H. Dunham, "Using Semantic Caching to -Manage Location Dependent Data in Mobile Computing", In the 6th Annual International Conference on Mobile Computing and Networking, Boston: ACM Press, 2000, pp.210-222
- [2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [3] Android - An Open Handset Alliance Project, <http://code.google.com/intl/zh-CN/android/>.
- [4] Android, <https://developers.google.com/android/>.
- [5] Android Developers, <http://developer.android.com/>.
- [6] Marko Gargenta, Learning Android. Machinery Industry Press, BeiJing, 2012.
- [7] Google Maps Android API, <https://developers.google.com/maps/documentation/android/index/>.
- [8] HuaiYu Fan, Android development essentials. Machinery Industry Press, BeiJing, 2012.
- [9] Google Direction API, <https://developers.google.com/maps/documentation/directions/>.