

Development and Implementation of Location Based Native Mobile Application

Case Study

Zlatko Čović

Subotica Tech – College of Applied Sciences,
Department of Informatics,
Marka Oreškovića 16, Subotica, Serbia
chole@vts.su.ac.rs

Dejan Babić

Devtech d.o.o,
Jevrejska 15, Novi Sad, Serbia
babicaja@gmail.com

Abstract – This paper presents the development and implementation of a location based native mobile application for smart phones with the Android operating system. It can be used in promotion of some product or brand in the way of mobile games. The application has the administration interface (back-end) for data manipulation of users, administrators, clients and items. The system distinguishes between three levels of users: the main administrator, administrators-customers and users. The paper also describes the concept of the division of mobile applications as well as the advantages and disadvantages of the implementation within each group. The application is tested in real conditions on more devices.

Keywords – GPS, LBMA, mobile application, web

1. INTRODUCTION

Access to the Internet using a mobile phone has made location-based services (LBS) a very popular part of the Web, from social network location-sharing, to driving directions or to finding interest points [2]. Based on the paper by Kölmel and Alexakis [1], location-based services are services that exploit knowledge about where a mobile device user is located. The technology to pinpoint the location of a mobile phone is available today and is of important commercial value to the market that wants to target more customers via mobile phones and mobile applications [1]. These services are often used in location based mobile applications (LBMA) which can be developed in various ways.

Students at Subotica Tech – College of Applied Sciences learn how to develop mobile applications in several courses with different technologies and programming languages. This paper presents a mobile application that was developed as a thesis of an Informatics student.

The second section deals with the term *mobile application* and its types. The third section briefly

describes the implementation of application. The next sections bring information of technologies used, the main activities of application, the administration panel, testing and results.

2. MOBILE APPLICATIONS

Usage of mobile internet and mobile applications is in progress and growing rapidly. People own two or more mobile devices and each of them contain pre-installed or installed applications. A mobile application is a software application designed to run on smartphones, tablets and other mobile devices. Mobile applications can be downloaded from app stores/markets that are distribution platforms for every operating system (OS). Based on the way they are developed and realized, mobile applications can be divided into three groups:

- Native applications,
- Web-based applications,
- Hybrid applications.

Native applications are developed in programming languages for certain OS, using certain IDEs (integrated development environment) for the chosen OS. In some

phases, development is harder and the programmer could create an application strictly for one operating system. With this approach, the application can access and take advantage of all hardware capabilities of a mobile device that are accessible for development. Native applications provide the best usability, the best features, and the best overall mobile experience.

Web-based mobile applications are developed by using several internet technologies and programming languages. These applications are basically a web page, or a series of web pages, that are created to work on a small screen. They can be opened with any modern mobile browser. The content of applications is on the web, it is searchable and it can be a great advantage for business and marketing purposes depending on the type of application. The benefit of these applications is that they are platform independent, the development process is faster and the programmer actually develops for more operating systems at the same time. The main disadvantages of web-based applications are that they need to be constantly connected to the Internet and they cannot access native hardware capabilities of mobile devices [3].

Hybrid mobile applications use some frameworks that allow programmers to develop native applications with the usage of Internet technologies such as a HTML5, CSS and JavaScript. Hybrid applications store their files on the device and on a server or use some communication techniques between client and server. The most popular framework for this kind of applications is PhoneGap. PhoneGap is a free and open source framework that allows programmers to create mobile applications using standardized web APIs (*application programming interface*) for the most popular platforms [4].

3. DEVELOPED APPLICATION

The developed application is a native mobile application for the Android operating system (OS) with some hybrid parts. The system distinguishes between three levels of users, i.e., the main administrator, administrators-customers and users. The main administrator is a user of highest priority and possibilities. He can control and manage all data. Administrators-customers are second-level administrators. They present customers which can put virtual hidden objects for their company or brand. They can also see who found the object. Users are persons using the mobile application. They search for hidden items, which can be divided into several parts, scanning the markers with the GPS in a real environment. Virtual markers are images of some products or company logo.

4. USED TECHNOLOGIES

Several technologies were used in the development process. Java programming language was used for mobile application, JavaScript, HTML, CSS, PHP and MySQL for server-side web application. Marker location

was defined by using Google Maps API and data synchronization was realized with JSON (JavaScript Object Notation).

This version of application was developed in the API 15 (ICE CREAM SANDWICH) version of Android and the minimal version of Android for using this application is API 10 (GINGERBREAD).

The application consists of 12 activities and three auxiliary classes. Each activity represents a single screen, a graphical interface with more or less options [3].

Auxiliary classes are used for the purpose of some of the activities and they are separated from them for better visibility of the code.

A list of activities:

1. *IntroScreenActivity* – startup screen with logo
2. *SetupActivity* – basic checks and account selection
3. *GetDataActivity* – download of data and images for read location
4. *MainMenuActivity* – main navigation menu
5. *SearchActivity* – search of environment
6. *ListActivity* – list of items to search
7. *FoundActivity* – display of founded objects
8. *RankActivity* – user rank list
9. *ProfileActivity* – nickname change
10. *UpgradeActivity* – upgrade of account to access all of items
11. *HelpActivity* – guide
12. *InfoActivity* – contact information

A list of auxiliary classes:

1. *DbModule* – manipulation with database
2. *ItemsAdapter* – modified adapter of array (existing element)
3. *ItemsHolder* – class tailored to the structure of hidden object

4.1. COMMUNICATION WITH SERVER

The application needs a stable internet connection for receiving and sending data to the web server. All communication is done with JSON. JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and a web application, as an alternative to XML [4].

On the server side, PHP scripts are executed, and based on HTTP request, they send a JSON array of data in response.

5. DESCRIPTION OF ACTIVITIES

Applications or activities, as parts of application, are triggered by *Intent*. *Intent* is an asynchronous message which activates a single component and it carries the information about what should be done and how. Management of *Intents* is done by the Android OS. After starting, every activity goes through the life cycle [3].

It is necessary to assign specific permissions within the *AndroidManifest.xml* file. This application requires the following permissions.

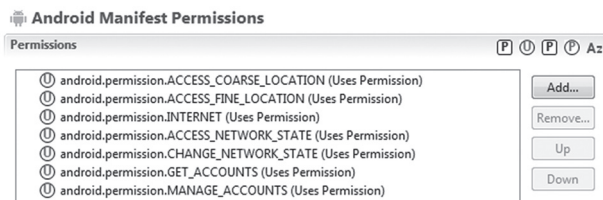


Fig.1. AndroidManifest.xml file

5.1. INTROSCREENACTIVITY

After launching the application, this activity shows the initial screen in the period of 3.5 seconds and then the next activity starts.



Fig.2. Welcome screen of the application

5.2. SETUPACTIVITY

This activity carries out basic checks of certain functions of the phone and activates all necessary services. If everything is correct, and if it succeeds in determination of phone location, selecting one of the active Google accounts, a user may push the start button to download necessary data about hidden objects.



Fig.3. Setup screen

Checking is done in linear steps, and when a singular item is checked, a user gets information about that with the change of icon in a graphic interface or through an alert window. The following is checked:

1. Database integrity (provides a place to store information about hidden objects),
2. Internet availability (provides communication between device and server),
3. Is a GPS enabled? (GPS provides a major tool in the whole application),
4. Is the location successfully determined? (check of location restricts access to objects in a given location),
5. Is there an active Google account? (provides user access to the system).

5.3. GETDATAACTIVITY

In this activity, a list of hidden objects for a predetermined location is received in communication with the server. Based on the list from the server, images of the object are downloaded only if they have not been downloaded yet. If this is the first launch of the application, the user registers on the server. All operations are divided into two *threads* [3].

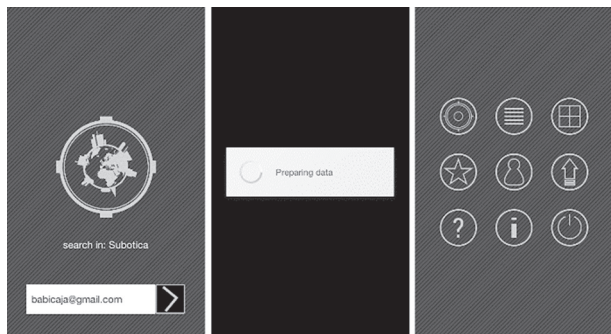


Fig.4. Receiving data

5.4. MAINMENUACTIVITY

This activity presents the main navigation menu of the application. It contains nine options

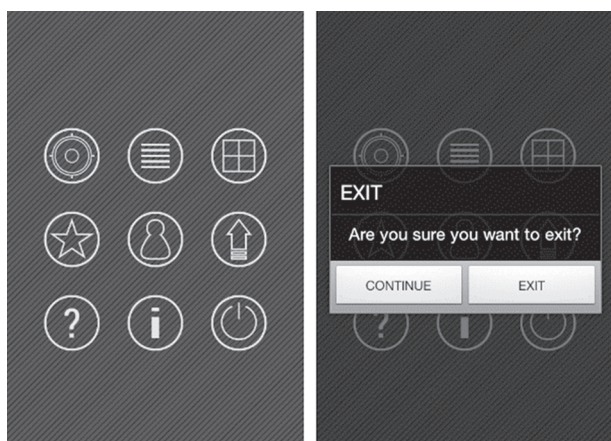


Fig.5. Navigation menu

5.5 SEARCHACTIVITY

Through the controls within the application, a user starts to scan the environment on the basis of its location obtained by GPS receivers. In this activity, by pressing the search button, a user compares its position with the marker data from the server. In case the object is not within 20m in diameter from a user, but within 300m, a user will be informed what direction to go.

For every downloaded marker, the application checks if they are already found. If they have not been found, they are compared with the current location. The result is the distance of the current point to the marker in meters. If the object is found, its data are read from the database and sent to the server. If the object is not found, parameters are set for feedback.

The whole process begins by clicking on the START button that calls for the *startScanThread()* and *startScan()* method, respectively. This method works as follows:

1. Animation starts.
2. Opens database.
3. Requires the determination of the location.
4. Records the current location.
5. Requires markers from the server to compare.
6. Requires found markers from the server.
7. Compares the current location with markers.
8. Based on the comparison result, notifies the user.

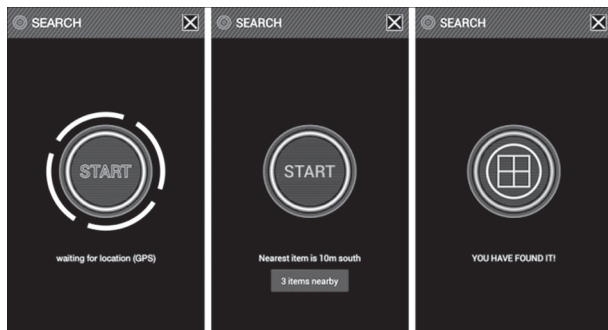


Fig.6. Search activity

5.6. ITEMSLISTACTIVITY

This activity graphically presents objects that are searched by the application. Basic information about the object are displayed in the figure below. Each time the activity starts, the database is updated and synchronized with data from the server.

5.7. FOUNDACTIVITY

In this activity, a user can see found items and the information if anyone else has completed a given object in the meantime. Next to the bigger picture, a user can see how many parts are needed for completing the current object. If the item was found and completed, the user receives a voucher – an email confirmation that the object is found [3].

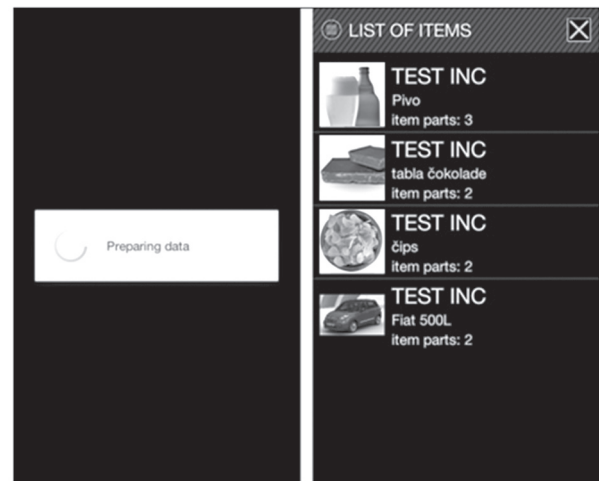


Fig.7. List of items

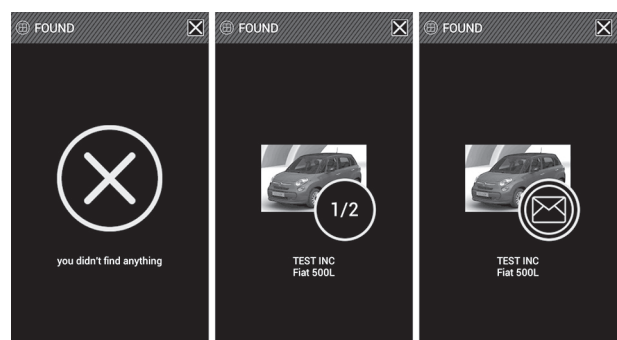


Fig.8. Information of found objects

6. ADMINISTRATION PANEL

The application has an administration interface (back-end) for manipulating data for users, administrators, clients and items. This is a web-based server-side application.

The main options in the administration panel are:

- management of administrators,
- management of hidden virtual objects (the location is determined by the Google Maps API),
- management of customers,
- management of users,
- additional functions.



Fig.9. Administration panel

opis	#	id	datum	komentar	predmet	lokacija	delova	zivo	izmenj	obrisi
TEST INC	1	9	28.03.13	TEST INC	tabla Piva	Subotica	3	da	✓	✗
TEST INC	2	10	28.03.13	TEST INC	tabla Zlatolada	Subotica	2	da	✓	✗
TEST INC	3	11	28.03.13	TEST INC	opis - demoeasli	Subotica	2	da	✓	✗
TEST INC	4	12	28.03.13	TEST INC	Flat 500L	Subotica	2	da	✓	✗

Fig.10. Data presented in tables

7. TESTING AND RESULTS

Through the testing process, the administration panel showed no problems with data management of the system. After installing and launching applications, the system showed the real situation; objects entered through the admin pages were visible in the application.

During the scan in several locations near hidden object, results were obtained that show the accuracy of a comparison of markers downloaded from the server and markers read from the location on the device itself. Mobile application was tested with two mobile devices: *Samsung GT-I9100 2* and *HTC - Desire X*.

The administration panel was tested in several web browsers.

Further development of the system requires adaptation of the application to newer versions of the Android OS and, if necessary, this should also include the previous one that was not covered by this development.

Since this application is a prototype, testing with a larger number of users and different mobile devices has not been performed, but is expected in the near future. Based on the obtained information, it will be easy to define additional guidelines for future development.

8. CONCLUSION

In this paper, the development and implementation of a mobile application were presented. This is a loca-

tion based native mobile application for smart phones with the Android operating system. The main activities are described. It can be used in promotion of some product or brand in the way of mobile games. The paper also describes the concept of division of mobile applications as well as the advantages and disadvantages of the implementation within each group.

The system is tested in real conditions on more devices and more web browsers. The application is available in Serbian and English.

9. REFERENCES:

- [1] B. Kölmel, S. Alexakis, "Location Based Advertising", Proceedings of the First International Conference on Mobile Business, Athens, Greece, 2002, pp. 1-7.
- [2] B. Pejić, A. Pejić, Z. Čović, "Uses of W3C's Geolocation API", 11th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 18-20 November 2010, pp. 319 – 322.
- [3] D. Babić, "Development and Implementation of Android Application", Thesis, Subotica Tech, Subotica, 2013.
- [4] D. Crockford, Google Tech Talks: JavaScript: The Good Parts; <http://googlecode.blogspot.com/2009/03/doug-crockford-javascript-good-parts.html>, (accessed: 7 February, 2009)
- [5] PhoneGap online documentation; <http://docs.phonegap.com/en/4.0.0/index.html> (accessed: 15 January, 2015)
- [6] R. Meier, Professional Android 4 Application Development, Wrox, 2012.