Building Mobile Tourist Guide Applications using Different Development Mobile Platforms

Sawsan Alshattnawi

Jordan, Irbid, Yarmouk University,
Faculty of Information Technology and Computer Science
sawsan kh@yu.edu.jo

Abstract

The rapid proliferation of mobile computing technology has massive potential for providing access to different services at any time and from anywhere. The mobile telephone is more than just making calls. It allows accessing several applications and services via the internet connection or by building stand-alone applications. The mobile telephone has a considerable effect in tourism by allowing the user to access the contents from Internet or from an install application over the mobile devices. The existing tourist guide applications use the latest technologies to enhance the application quality by satisfying the user's requirements. These applications encounter great challenges because of limited mobile resources. Several development platforms for mobile applications are used to design tourist guide applications a caused to mobile devices incompatibility. Some of these applications are designed using, for example, J2ME which represents an effective language for small and portable devices. But, the ideal platform, now, for this type of applications is Android, it is designed to improve efficiency and reduce resource consumption compared to J2ME. The most used technology is Java programming languages but Android is more powerful, flexible and Android phones have more functionality than J2ME phones. In this paper, we build electronic tourist guide systems using the two technologies to confirm the powerful of Android. For this reason, a comparison between the two systems is made. The applications are tailored to user preferences, so that the user can access the application from a simple interface or automatically; the information may be displayed according to GPS positioning system. The system is built to be a tourist guide for Jordan, but it is flexible and easy to be for any country. To do that, the administrator need to know the GPS coordinates of each point of interest, and then the desired information may be added to database.

Keywords: *Smart phones, Tourist guide systems, J2ME, Android, XML*

1. Introduction

Smartphones are growing exponentially and become popular to offer greater services to users like GPS navigation, web navigation, games and more. This encourages the developers to build applications that tailored to mobile phone specifications and could be used easily through this hand-held device.

Mobile tourism is a term that starts to appear in the last two decades. It involves using mobile device as electronic tourist guide [1]. The tourist needs to search information about a Point of Interest (POI) from his mobile. He can get the information from the internet via a web browser but this requires a constant wireless connection to the internet. The continuous connection is not possible and some locations have not an internet coverage. Therefore, building a stand-alone application that could be installed over the mobile device to be used in off-line case was a good solution. The wireless connection in this case is not a necessary. However, building this type of application brings a big challenge because of the limited mobile resources such as the memory size, the CPU cycle, the battery life, *etc*. Therefore, the approach that is used to solve this problem is to develop

applications according to the resources capacity of each mobile device.

Some of existing tourist guide applications are designed for multiple mobile platforms a caused to incompatibility of mobile devices. The last recently used technology is Java 2 Mico Edition (J2ME). J2ME has rolled out in 1998, it represents a platform for small, portable with specific user interface and hardware constraints mobile devices. J2ME allows developer building applications that are executing on any device supporting CLDC/MIDP regardless of their underlying software/hardware architecture [2].

Most mobile phones support J2ME, and therefore, the existing applications have been built using this technology. But Android now is the largest installed base of any mobile platform. It is Linux-based system developed by Google. It is a world-class platform that is growing vastly [3] (comparing with J2ME). Android has penetrated the digital world easily and it has rapidly become the fastest-growing mobile OS. The final statistics point that more than one million new Android devices are activated everyday [3].

In this paper, we build a tourist guide application customized to user preferences using J2ME and Android programming language. A comparison between the two applications is made at the end of this paper. The system is built to be a tourist guide for Jordan and, at the same time, it is built to be flexible by allowing to change the information at any time according to a desired city. For example, the system can be a tourist guide for another city by allowing the administrator to define from the web site the GPS dimensions of this city and the correspondent information to the tourist locations, hotels, restaurants, *etc* cloud be stored in the database.

The paper is organized as follows: the rest of this section explores the J2ME and Android architecture and some of the existing electronic tourist guide applications, Section 2 presents the system design, its implementation and the system functionality, Section 3 presents a comparison between the two applications, and finally, Section 4 concludes the paper and presents the future work.

1.1. J2ME and Android Architecture

Java 2 micro Edition (J2ME) and Android applications are written in Java. J2ME uses the JVM [4] and Android has its own VM which is called Dalivik. It has a special byte code format that corresponds to Android devices requirements [5]. The generated .dex files are smaller than the generated using usual Java byte code because the .dex files contain unique data. If several classes share the same string, the string exists only one time in the .dex file and next occurrences will be just a pointer to this string.

Figure 1 shows the software stack for Android. It consists of five layers [3]. The basic layer is the Linux kernel which interacts with the hardware and contains all the essential hardware drivers. Drivers are programs that control and communicate with the hardware. The Linux kernel also acts as an abstraction layer between the hardware and other software layers. Android uses the Linux for all its core functionality such as memory management, process management, networking, security settings *etc*.

The next layer is the Androids native libraries. This layer enables the device to handle different types of data. These libraries are written in c or c++ language and are specific for a particular hardware. Android Runtime consists of Dalvik Virtual machine and Core Java libraries.

Application Framework layer consist of blocks that user's applications directly interacts with. These programs manage the basic functions of phone like resource management, voice call management *etc*. The upper layer is Applications layer where user's applications are going to fit. Several standard applications come pre-installed with every device, such as: SMS client app, Dialer, Web browser, Contact manager.

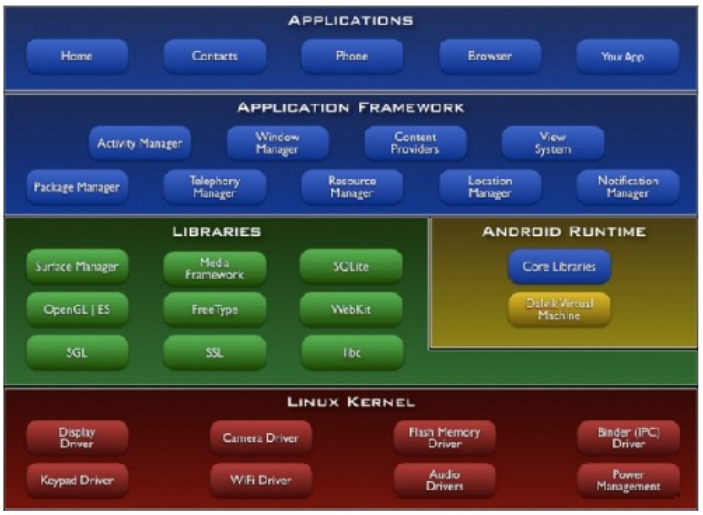

Figure 1. Android Software Stack [3]

In Android, the virtual machine that runs the program is called Dalvik, Dalvik virtual machine is a register-based architecture [6]. J2ME is a program that is run in virtual machine, without a need to recompile it [7]. Java Virtual machine is a stack machine which use instructions to load data on the stack. It needs more instruction to implement the same high level code than register machine [6].

J2ME applications are called MIDlets; MIDlets are usually packaged in *.jar files, down- loaded on-the-fly from the Web server and executed as stand-alone application with no requirement for a constant connection to a wireless network [2]. J2ME Architecture is shown in Figure 2. J2ME Architecture is consisting from five layers: optional packages (Top Most Layer) which contains Java APIs for user network connections, persistence storage, and the user interface. It also has access to CLDC libraries and MIDP libraries. The second layer is J2ME APIs (Profiles) which consists of the minimum set of application programming interfaces for the small computing device. The third layer is configurations which handles interactions between the profile and the JVM. The next layer is JVM and the bottom layer is the operating system.

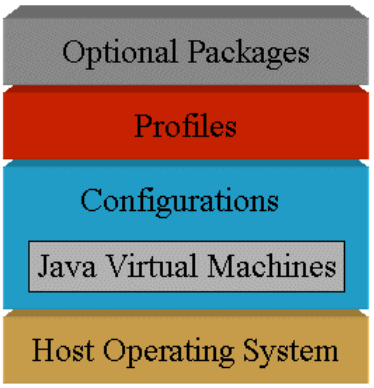

Figure 2. J2ME Architecture [8]

1.2. Related Work

In this section, the existing mobile tourist guide applications will be explored. These applications are specific and tailored to specific platforms. They are categorized according to Kenteris [9], into two categories:

1. Pre-installed applications: the application must be installed and run on the device.
2. Web applications: using web to browse the tourist information through a thin client on the device. These applications are considered as platform-independence.

The first approach needs some mobile resources such as storage and CPU computation power to be installed and run on the device. Also, the tourist content is already defined and installed in the user device.

These applications restrict the mobile devices to have some specific platform, they cannot be portable to other platforms, the information is static and limited in size a caused to the limited mobile storage, and they need time to be installed in client side. In addition, periodical synchronization must be done between the client and the system when required if possible.

The applications presented in Cyberguide [10] and Mytilene E-guide [1, 9, 11] are belonging to this category. Cyberguide [10] is mobile context-aware tourist guide; provides more services depending on the current location as well as history of previous location. A caused to incompatibility of mobile devices, many applications are designed for multiple mobile platforms: Delphi prototype and Visual Basic prototype. Cyberguide composed of several separate modules for indoor and outdoor use. The first prototype is designed for indoor use with static maps and infrared (IR) for indoor positioning. The extended version of prototype was for the outdoor use by replacing different maps without any problems and using a GPS as positioning system.

Mytilene E-guide [1, 9, 11] is designed for multiplatform: web application for PC needs network connectivity and stand-alone application to be installed on mobile phones and operate without network connectivity. The programming is done in J2ME. The visitor must connect to the internet from his PC or tablet computer to provide his preferences, the application is built and must be installed on the mobile device. In case of update the server checks always for any update and tell the user by SMS containing the new contents description.

The second approach overcomes some disadvantages of the pre- installed applications because web applications can be accessed from any platform and they are accessible from anywhere. But they need constant connectivity to the network. LoL@ [12], GUIDE [13] and Hippie [14] are Web tourist guide applications. These systems use the mobile computing technologies and wireless infrastructure to help the tourist to get information about the city adapted to their personal and environmental contexts. LoL@ [12] is a mobile electronic tour guide build over the UTMS for powerful devices and network connectivity. End-user devices are continuously connected to the application servers via UMTS or GPRS.

GUIDE application is a web-based application [13], it requires a continuous connectivity. Tourists can get good and adapted information to their locations and preferences. Portable GUIDE units obtain positioning information by receiving location messages that are trans- mitted from strategically positioned base stations. In case of disconnection GUIDE units cache some of information locally, but during the period of disconnection, as mentioned by the authors of GUIDE, the out of date information being presented to tourists. To overcome this problem, the user interface to GUIDE has been designed to encourage the user to form a suitable mental model of the system by providing visitors with feedback regarding the current state of connectivity and also encourage them to associate this with available functionality.

From our readings, we noted that the application is designed for multiple platforms and the dominate technology was J2ME because it is supported in several mobile devices. But now, the most mobile devices that try to dominant the whole word is Android. Several article have shown the superiority of Android over J2ME such as [15-17] for this reason, we like to build this application and at the end an usability study will be made. Just now we design and implement the first prototype for this application.

Before we begin, an investigation is made arbitrarily between 70 tourists about the telephone models they use currently and the model they want to buy if they have desire to change the model. The result is shown in Table 1.

Table 1.

OS	Percentage
Android	33
Sambian	21
Blackberry	9
Windows Phone	18
Others	19

2. Proposed Mobile Tourist Guide

The proposed mobile tourist guide (ETGS) design consists of two steps, the first step is designing a web site that allows the user to navigate a web site, discover the locations and the interesting points before the real visit. The second step is designing the mobile applications.

When the user visits the web site, selects the POIs that he wants to visit and specifies some preferences such as the visit duration, the mobile model, and the language, then the information about these locations are kept over the user space and an application is being built. The application must be now installed in his mobile phone. The user runs the application while visiting. He accesses the information from a simple interface or the information about any location is displayed without the user intervention according the GPS positioning system, where the matching is done between the GPS coordination and the GPS dimensions stored for each location over the device. If so, the information about this location is displayed.

The system is designed to allow tourist to work in on-line from the web site or off-line mode from the installed application over the mobile device. It consists of the following components as shown in Figure 3.

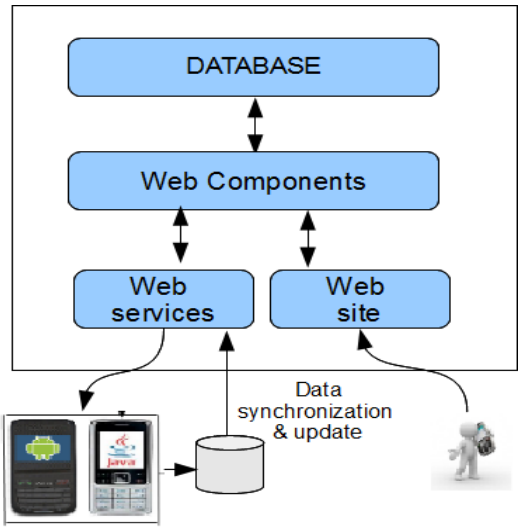

Figure 3. The System Architecture

1. The database that contains all the desired information about locations and users. The tourist can share experience to enrich the information in the database. It is built using MySQL database. MySQL database has become the world's most popular open source database because of its high performance, high reliability and ease of use [18].
2. The web site which allows the user to register and provide his preferences. The user can also access the information without registering. ETGS web site uses JQuery [19] to perform some functions at client side like parse xml data. The web site design is shown in Figure 4. It consists of the following components: a) Admin database contents: allow super users to add new data or update existing data in the database. b) Mobile application generator: this component is responsible of building the application that is tailored to the user preferences and device type. c) View database contents: explore information in the database. For example personal information, location, other user s comments or rating, photos, etc. d) XML generator: the information in the database is put into XML format to be used by JQuery. JQuery also parse XML format to be viewed on the map.

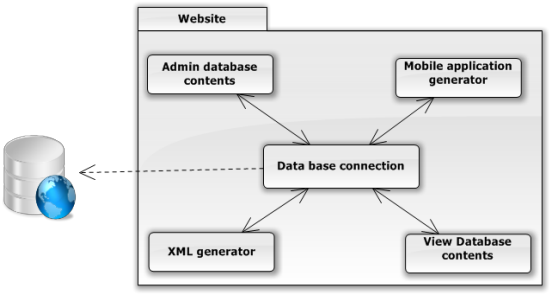

Figure 4. The Web Site Design

3. The web service is built to provide services for users who install the application over their mobile device. The user can access the information from the local storage in case of offline or update the local storage with information from the server on online case. This component helps the user to build an application in his mobile device to be used in case of disconnection or under the user demand. The amount of information over his mobile is determined by the mobile device storage capacity. The web service is designed using Restful [20] instead of WSDL. Restful depends on HTTP protocol and the services could be provided more quickly than in WSDL.

The system is built to be as much as possible flexible. It is built to allow some type of users changing and updating the database contents. To simplify this function we distinguish between many user levels and privileges given to each level. The user types are defined as follow: 1) Visitor: any user "not registered" visit the web site, he has initial privilege to access information and maps. 2) Tourist: the user that registered to plan his trip. He can access all information and all maps. After registering, he provides his preferences. The information will be presented according to user s preferences. In addition, an application will be built according to these preferences. He can install this application on his device to access the information via his mobile phone in the offline mode. 3) Groups admin: ETGS supports a group of tourist, the group admin is super user from the "Tourist" who has more privilege to manage a group of users such as add new groups, add new "Tourist "to group and managing some information between the members in a group. 4) Web admin: ETGS needs web site administrator to manage information in the database. 5) System admin: is the owner of the system or the manager of the whole system. He manages other users and their privileges.

The user can synchronize information between his space on the server and the mobile

device. The application contents in the device can be updated by the user. For example, if the user changes one of his preferences such as add new location in his trip outline, one component is responsible to update local storage with the related information about that location. On the other hand, he can upload his photos, his comments and evaluate places in terms of beauty and services quality.

2.1. EGTS Web Site

One of the most important parts of EGTS is the web site, this importance comes from many points:

1. The web site is the first place for the tourist to register and build mobile application.
2. The web site manages whole system services and users.

ETGS use Java as server side programming language. HTML5 is used to build compatible web site with most devices because the mobile web is growing up quickly, and tons of great HTML5 features are already supported on modern mobile browsers. Web developers can use the same set of technologies they know and to build rich web applications that work across different device types. The web site use JQuery to do some needs functions at client side like parse xml data at client side "on some pages and for some browsers "and AJAX to get data from web services.

The Website main functions are:

- View map: one of the main functions on the web site "View maps" and surfing all data attached to it. The user can customize the POIs on the map by determining the type of tourism. For example, religious tourism, education tourism, *etc*.
- Register: anyone can register to get more information, get mobile application and also add his profile in ETGS to be accessible from any device.
- Manage data: users must have the capability to add, update or remove data, but every user have his own privilege to manage data, for example tourist "user" can manage own profile and change his preferences and also he has access to add new information about the destinations, and other example the web admin user can do more than tourist like manage destinations, places and cities as shown in Figure 5. Also the "admin" can change and modify the information in DB.

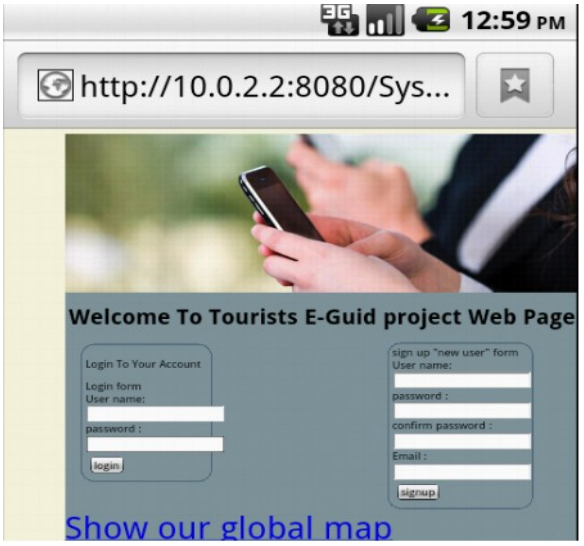

Figure 5. The User Login: Register Web Page

2.2. J2ME and Android Mobile Tourist Guide

When the user select locations that he want to visit from the web site, an XML file is created. This XML file will be used as input to J2ME and Android applications. It contains some information about selected destinations and some places (res, hotel, *etc*.) for every place. Another files will be generated that contain the map images, since this application will be run in an offline mode, we have to get all needed information on the user device. So, when he selects destinations to visit we generate (using Google Map API) static image for every destination and install these images to be used in the generated applications.

The web application generates .jar or .apk file according to the user mobile model. This information must be provided while registering. The user access the application during the visit from an easy interface. A list of locations is displayed when the user is in a specific location he can get information about this location, hotels or restaurants in that location. Also, the user can display the map to see his location. The figures show the run to one application. Figure 6(a) shows the information about a location, the map, and a list contains additional information that the user may access such as the hotels or restaurants in this area. The places that are given are those that have the highest rank. Figure 6(b) gives some information when the user selects the hotel name from the list.

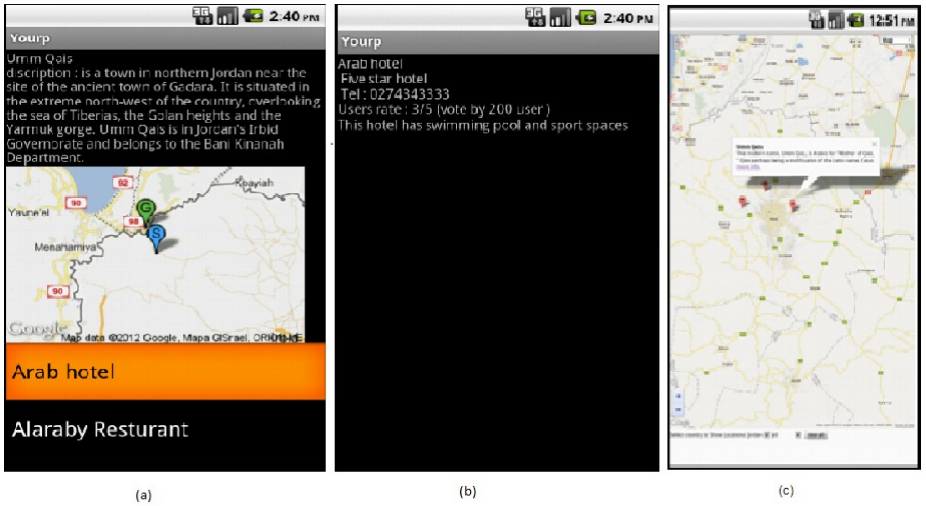

Figure 6. Screenshots of the Application Execution

2.3. System Functionality

We tried to make the system function in many cases depends on the user preferences or the internet status. Three usage scenarios are considered:

1. A tourist plans his trip at home and navigates information through web site from his PC or Mobile phone. The tourists see the interesting locations; he has to register through the web site and provide the necessary information about his trip (user preferences).
   When the tourist begins his trip and connects to his personal space, an execution of the application begins and the information is provided according to tourist location. The tourist can choose to upload additional information and pictures he takes to his space. This allows tourist to take any number of pictures without need to big data storage in his mobile. He can share his trip with others by upload all his comments and pictures to his page in any social network. This is left as future work.
2. Off-line functionality: a tourist can download an application which will be installed in his mobile device. In this scenario, the tourist chooses to work totally in offline mode.

3. The third scenario is for a tourist that does not register and want to navigate the web site to get the necessary information. The tourist has some buttons in the web site to determine some preferences like the language. ETGS in this case is like any web application where the continuous connection to the internet is a necessary. In this scenario, the mobile device may have the minimum requirements such as Wi-Fi or any type of connection with the internet.

3. Comparison

The implementation of the two identical systems is made. We begin with Android implementation and then the implementation of the same system is done in J2ME. During this process, we noticed that the some functions are difficult to code in J2ME than in Android and it desires more instructions to be done as in Android. First of all, the interface design in Android is easier than in J2ME.

Android API is much more powerful than the J2ME. Although Both J2ME and Android Java applications are recognizable Java programs, made from assemblies of Java classes packaged into archives. The creation of complex form can be done very quickly, and software package for the Android SDK is easy to install. Google re-designed Java for Android to improve efficiency and reduce resource consumption.

When user start application the XML file is parsed and the information is saved on a SQLite in Android application and in memory RAM in J2ME application. Android use SQLite which is more efficient because it need small memory because the data is loaded as require to memory.

When J2ME application is added on any phone this means a new layer (virtual machine VM) is added. Unfortunately this VM cannot access the OS directly, every time we run the application we have to get the access to OS, but when the user stop the application this application cannot access the OS directly. Android is modern phone OS and has embedded Java VM which help developer to coding his application on Java and then run the application as part of OS and with any access at any time.

4. Conclusion

This paper presents the advantages of new popular technology, Android, that will be the dominate technology. We presented already the reasons that make Android popular. To confirm the effectiveness of this technology, we build our first prototype using J2ME and Android and compare between them from development point of view and efficiency of using resources from another point of view.

These applications are still restricted to mobile resources and to the connection of the internet. In addition they may suffer from reliability and scalability. The user can share his experience and contribute to enrich the information in the database. For this reason, the number of updates may be overloaded the server if the number of users is grown quickly. To make the system scalable, the new promising technology that will help to alleviate several problems encounter the mobile applications is cloud computing. This technology will conquer the poverty of mobile resources by using different services of cloud computing, it provides also flexibility and scalability to use the computing resources on-demand. Therefore, we will modify the system design and implementation to be cloudy.

Acknowledgment

I would to greatly thank the Ministry of Higher Education & Scientific Research in Jordan for the financial support that they granted to accomplish this work.

References

[1] M. Kenteris, D. Gavalas and D. Economou, "An innovative mobile electronic tourist guide application", Personal Ubiquitous Comput, vol. 13, **(2009)**, pp. 103-118.
[2] S. Kagstrom, H. Grahn and L. Lundberg, "Cibyl: an environment for language diversity on mobile devices", Proceedings of the 3rd international conference on Virtual execution environments. VEE '07, New York, NY, USA, ACM, **(2007)**, pp. 75-82.
[3] Developer: developer. (http://developer.android.com) [4] Johnson, S.: Java in a teacup. Queue, vol. 4, **(2006)**, pp. 36-41.
[5] H. S. Oh, B. J. Kim, H. K. Choi and S. M. Moon, "Evaluation of android dalvik virtual machine", Proceedings of the 10th International Workshop on Java Technologies for Real-time and Embedded Systems. JTRES '12, New York, NY, USA, ACM, **(2012)**, pp. 115-124.
[6] Y. Shi, D. Gregg, A. Beatty and M. A. Ertl, "Virtual machine showdown: Stack versus registers", In Proc. ACM/USENIX Int. Conf. on Virtual Execution Environments, ACM Press, **(2005)**, pp. 153-163.
[7] T. Lindholm and F. Yellin, "Java Virtual Machine Specification", 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, **(1999)**.
[8] S. Microsystems, "Java 2 platform micro edition (j2me) technology for creating mobile devices", **(2000)**.
[9] M. Kenteris, D. Gavalas and D. Economou, "A novel method for the development of personalized mobile tourist applications", Communication Systems and Networks, pp. 208-212.
[10] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper and M. Pinkerton, "Cyber- guide: a mobile context-aware tour guide", Wirel. Netw., vol. 3, **(1997)**, pp. 421-433.
[11] M. Kenteris, D. Gavalas and D. Economou, "Mytilene e-guide: a multiplatform mobile application tourist guide exemplar", Multimedia Tools and Applications, **(2011)**.
[12] G. Pospischil, M. Umlauft and E. Michlmayr, "Designing lol@, a mobile tourist guide for umts", Proceedings of the 4th International Symposium on Mobile Human- Computer Interaction. Mobile HCI '02, London, UK, Springer-Verlag, **(2002)**, pp. 140-154.
[13] K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou, "Developing a context- aware electronic tourist guide: some issues and experiences", Proceedings of the SIGCHI conference on Human factors in computing systems. CHI '00, New York, NY, USA, ACM, **(2000)**, pp. 17-24.
[14] R. Oppermann and M. Specht, "A context-sensitive nomadic information system as an exhibition guide", Proceedings of the Handheld and Ubiquitous Computing Second International Symposium, HUC 2000, **(2000)**, pp. 127-142.
[15] D. Gavalas and D. Economou, "Development platforms for mobile applications: Status and trends", IEEE Software, vol. 28, **(2011)**, pp. 77-86.
[16] T. M. Grønli, J. Hansen and G. Ghinea, "Android vs windows mobile vs java me: a comparative study of mobile development environments", Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments. PETRA '10, New York, NY, USA, ACM, vol. 45, no. 8, **(2010)**, pp. 1-45.
[17] X. Meng and Y. Ou, "A solution for application transplantation from j2me to android", Future Information Technology and Management Engineering (FITME), 2010 International Conference, vol. 3, October, pp. 60-63.
[18] Oracle: mysql The world's most popular open source database. (http://www.mysql.com/why-mysql/)
[19] JQuery: JQuery. http://rungringjung.files.wordpress.com/2010/10/why-jquery1.pdf, **(2010)**.
[20] C. Pautasso, O. Zimmermann and F. Leymann, "Restful web services vs. "big"' web services: making the right architectural decision", Proceedings of the 17th international conference on World Wide Web. WWW '08, New York, NY, USA, ACM, **(2008)**, pp. 805-814.

Author

Sawsan Alshattnawi is an Assistant Professor in the Department of Computer Science at Yarmouk University (Jordan) since June 2009. She has received her PhD degree in Computer Science from Henri Poincar´e University (France) in 2008, her M.Sc. degree in Computer Science from Yarmouk University in 2003, and her B.Sc. degree in Computer Science from Yarmouk University in 1994. Her research interests include: Distributed systems, Cloud Computing, and Mobile Computing.