International Journal of Education and Social Science Research

ISSN 2581-5148

Vol. 7, Issue.1, Jan-Feb 2024, page no. 22-30

To cite this article: Luay Abdulwahid Shihab, Sura Saad Basher and Manhal Mohamad Basher (2024). STUDY OF THE DEVELOPMENT ENVIRONMENT FOR MOBILE IN (JAVA, ANDROID, WINDOWS), International Journal of Education and Social Science Research (IJESSR) 7 (1): 22-30 Article No. 886, Sub Id 1383

STUDY OF THE DEVELOPMENT ENVIRONMENT FOR MOBILE IN (JAVA, ANDROID, WINDOWS)

Luay Abdulwahid Shihab[1], Sura Saad Basher[2] and Manhal Mohamad Basher[3]

[1]Assist Prof. College of Nursing, University of Basrah, Basrah, Iraq.
[2]Northern Technical University, Mosul, Iraq.
[3]Northern Technical University, Mosul, Iraq

DOI: https://doi.org/10.37500/IJESSR.2024.7103

ABSTRACT

In this research we will look at three types of development environment for mobile are Windows Mobile and Android and Java Mobile, by comparing the basic base of the programs, different environments examined tightly points of strength and weakness, many of the attributes or features that will be used in the process of comparison such as implementation, Performance, quality in order to have a deep view of the different environments used in the comparison process.

Our study showed that although the three environments are similar in some characteristics, they still represent three distinct fields, each with its own characteristics. Through code examples and our platform comparison, we conclude that Windows Mobile and Android provide a better development environment, while Java ME still suffers from poor support for incompatible applications. The main features common to them appear in trends that appear directly in programming languages such as programming section testing and language features. Large differences are distinguished when examining community environments, hardware capabilities, and hardware. This will have a major impact on choosing a development platform to create new applications for the environment.

KEYWORDS: Android, Windows Mobile, Java Mobile, development environment mobile, programming languages.

1. INTRODUCTION

That the proliferation of widespread technologies for livelihood assistance has undoubtedly been a factor in their success, relatively little work has been done, in addition to the study of code words in such technologies. Accordingly, in this paper, we will consider everything related to the development of the three platforms widely available for widespread applications and comparison between the three main platforms:

1. Java ME is a platform that supports a wide range of mobile devices from low-cost, low-cost phones.[1]

2. Windows Mobile has an alternative for professional users who can take advantage of the Microsoft .Net Compact framework in high-end mobile devices.[2]

3. In 2007, Google released what is known as Android, a Linux-based operating system that targets the smartphone market.[3]

We will compare platforms in several different categories, including aspects of implementation, performance, developer support and quality assurance, with some examples of code. Also features are commonly used including permanent storage and open network connection as well as highlighting some differences between platforms.[5]

The various development tools will also be discussed, as they are an essential part of modern software development. This section contains a brief explanation of key features including debugging and automated testing support. This is critical to developers especially when using test-based development. These are the main topics of research in mobile software development. the computer and find out local area networks has become a major tool to many companies and factories, universities, hospitals and standardization of data and security services that protect the data and means of communication [14]. that any information must be correct, accurate, and available to be stored, retrieved, processed, and made available safely and reliably [15].

Look at platforms

Our research focuses on three major mobile platforms: Java ME, Windows Mobile and Android. Languages such as C ++, C are not included. Although we have not covered all major mobile development environments and platforms, this research makes a significant contribution to the research currently available in this area.

1. JAVA ME

Java ME was created to provide a shared platform, where the source code itself must operate on a variety of different platforms and devices according to Sun Microsystems, which has produced a robust and flexible Java ME environment for applications running on a wide range of embedded devices. In addition to mobile phones, Java ME works on TV, printers, remote communication systems, and PDAs. The Java ME platform is not one option but a set of different profiles [4]. One of the most common platforms for resource-limited devices is the CLDC1.1 / MIDP2.0 CLDC (limited-connection devices) is designed for down-end devices with 160KB or more memory and 16/32 bit slow processors.

Only a small portion of the Java SE core libraries are supported [6]. Although the vision behind the Java ME is to be cross-platform, this is not always, what happens in the real world. Java ME development often suffers from many complications, including the inconsistent virtual machine and package implementation. [7] The standard APIs were originally created to work on all types of devices including limited cell phones resources.

This means that basic libraries have very limited features. One classic example is the RMS (store management record), which provides the only way to permanently store information on many low-end phones. Moreover, RMS memory is very limited and difficult to use. To store anything in this database you will need to create your own sequence (since RMS only takes a byte array as an input) or uses an open-source library such as Floggy ie when the platform lacks functionality, one will likely find a third-party library that solves the problem or least some problems.[9]

2. WINDOWS MOBILE

Is a mobile operating system developed by Microsoft, It is based on Windows CE 5.0 and provides an operating system for a large number of PDAs, smart phones and touchscreen mobile devices [8]. The current version of Windows Mobile 6.5 enables the creation of custom written applications in both code (Visual C # / Visual Basic and .Net). More mobile phones are shipped with Windows Mobile.
The API is expandable and features rich features as well as a programmable layer. Unlike the limited Java ME system, Windows Mobile takes advantage of a rich environment of Microsoft .NET capabilities. The .Net Compact window provides access to almost all operating system features on the device. This allows sophisticated and feature-rich applications, but breaks Java ME's capabilities to run the same code on multiple operating systems.

The .NET Compact Framework shares a lot of capabilities with a full flexible desktop version such as Common Language Runtime (CLR) and Just-In-Time (JIT) to convert byte code into executable programs, Compact Framework is a C # Framework.net for application development The Compact Framework is used to create applications on mobile devices such as PDAs, smartphones and receivers, where the Compact Framework libraries act as a subset of the full Net framework with some additional libraries for mobile development. A rich application environment that makes it easy to run applications and databases O on the mobile device.

3. ANDROID

Google Android released Android in November 2007. [11] Aiming to be an open source platform for software development on the mobile platform. The Android platform was released under the so-called Open Handset Alliance. The goal of this alliance is to create open standards for mobile devices. Android is a Linux-based open source portable operating system. The operating system makes it easy for developers to write code in Java using Google lava. The Android platform not only provides the same mobile operating system as the development environment, but also provides a Dalvik Virtual Machine to run applications as well as being the middleware between the code and the operating system [10].

For application development, Android makes it easy to use 2D and 3D graphics libraries, a small custom SQL engine for persistent storage, and advanced network capabilities such as Edge, 3G, and WLAN. The API is constantly evolving and the current version (Eclair 2.0) is a huge step forward in terms of features available from version 1.0. Because Android is an open source operating system, the community welcomes it to collaborate in developing the programming environment, operating system, and API.

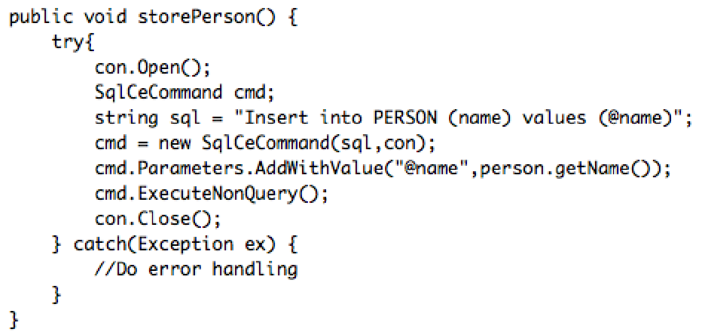
Comparison between the platforms of Java, Windows and Android

The following sections describe aspects of implementation, performance aspects, and developer support and quality assurance across the three platforms. These themes have been chosen because they represent important aspects for mobile developers and cover many of the commonly used features. It does not include comparing performance aspects in terms of run-time performance on the device because of the exact connection between software and hardware platforms. It is very difficult to find exactly the same hardware to run all the platforms, so our results will be greatly affected by the differences of the actual hardware. In addition, the simulator tests do not give the correct performance numbers as well. Performance comparison includes an example of network connectivity and file size. This comparison focuses on the development environment.

We have experimented on all three platforms to compare key factors in three mobile development areas. These applications were carried out in the IDEs described in this research, as well as platform simulators were used to test and verify examples of applications. Through code and discussion examples, we will highlight similarities and differences in the development of mobile software (environments).

4.1 In terms of implementation

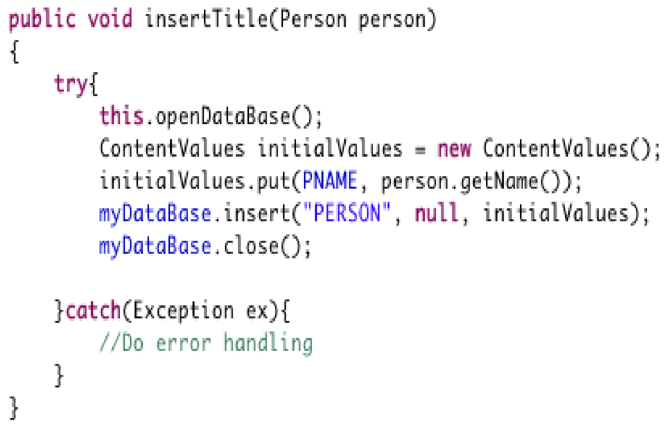
The storage technology varies between different environments. Both Android and Windows Mobile have the ability to use a small database on your device. This facilitates easy manipulation and extraction of data. On the other hand, Java ME uses a feature called the Record Management Store. This is a very simple database consisting of two columns, one for the record ID, and one for the actual data. When it comes to storing files locally, all three frames have the ability to use memory cards to provide additional storage space. The differences here lie in exploiting the storage area. Windows Mobile and Java ME have the option to install applications on the memory card while Android cannot do so. [12]



```
public void storePerson() {
    try{
        con.Open();
        SqlCeCommand cmd;
        string sql = "Insert into PERSON (name) values (@name)";
        cmd = new SqlCeCommand(sql,con);
        cmd.Parameters.AddWithValue("@name",person.getName());
        cmd.ExecuteNonQuery();
        con.Close();
    } catch(Exception ex) {
        //Do error handling
    }
}
```

Figure 1. Insert into database query in Compact Framework context

As shown in Windows Mobile (Figure 1) and Android example (Figure 2), both versions have similar methods. Both pass the steps to open access to the database before creating and issuing a query. Finally, the database is closed and access is stopped. Both examples have a high level of abstraction. This easily creates a readable and readable code, as well as it follows the general rules of programming oriented towards modern objects.



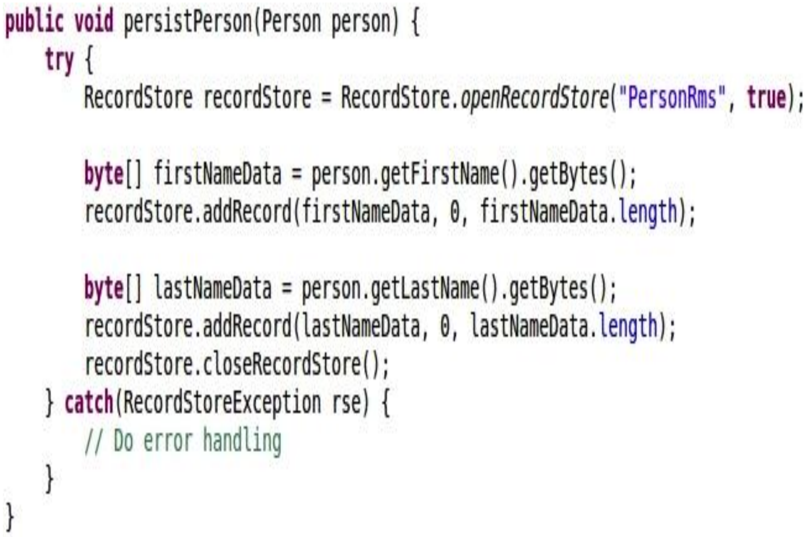
```
public void insertTitle(Person person)
{
    try{
        this.openDataBase();
        ContentValues initialValues = new ContentValues();
        initialValues.put(PNAME, person.getName());
        myDataBase.insert("PERSON", null, initialValues);
        myDataBase.close();

    }catch(Exception ex){
        //Do error handling
    }
}
```

Figure 2. Insert into database query in Android context.

In Java ME (Figure 3), Record Store provides a simple, user-friendly programming interface. However, they are very limited because you have to store byte arrays. The alternative offered by Windows Mobile and Android, using value objects and relational databases as it creates a better and more flexible development environment. Table (1) shows the number of lines for a static storage example per platform. The number of lines is similar to all platforms, and this indicates that the amount of modeling to open the database connection / RMS is somewhat similar.

Table 1. Lines of code needed for an insert into a database on the device.

Database connectivity, lines of code	Mobile platform
13	Windows Mobile
11	Java ME
11	Android



```
public void persistPerson(Person person) {
    try {
        RecordStore recordStore = RecordStore.openRecordStore("PersonRms", true);

        byte[] firstNameData = person.getFirstName().getBytes();
        recordStore.addRecord(firstNameData, 0, firstNameData.length);

        byte[] lastNameData = person.getLastName().getBytes();
        recordStore.addRecord(lastNameData, 0, lastNameData.length);
        recordStore.closeRecordStore();
    } catch(RecordStoreException rse) {
        // Do error handling
    }
}
```

Figure 3. Serializing a Person object and inserting the byte array into a Java ME Record Store.

4.2 In terms of performance

Performance and package size numbers are important for developers and users. To be able to compare this to the three development platforms, we chose to implement a basic application consisting only of the necessary code and files needed to create an executable program. The application was developed independently on all three platforms and we measured the required code lines in the application as well as the size of the package ready to be installed on separate devices. By calculating the lines of the code, we get an indication of how much serial code is available on each platform in order to run the application on the device.

We have chosen to calculate code lines only in the relevant code files and include configuration files and similar environment numbers. Although these files are excluded from the number of code lines, these files are still very important and the applications cannot do without them. We also measure the size of the publishable package that includes all the environment files as well as the actual icon files. The application sizes in this demo are so small that in most cases they do not add too much overhead when downloading and installing on the device.[13]

The purpose of measuring the file size was to get an idea of the extent to which configuration and playback dependencies were included in bundled applications. These combined numbers indicate how apps are grouped and what they effect on devices. Another important aspect of mobile phone development is network connectivity. This is especially important in mobile software because many applications rely on downloading information to the phone.

5- SUMMARY:

Although it is almost impossible to give a full picture of the features and challenges of each of the three platforms, we believe that we have presented in this paper an overview of their strengths and weaknesses. However, it is important to remember that the examples in this paper present only a very small fraction of the features of each platform. Examples were created to provide specific cases common to mobile software development. Most of the applications created today need to permanently save information, and can store user name or configuration settings. Network connectivity is critical to mobile software in particular. The typical case is where the application will download information from the Internet. These examples cover these examples.

One of the key strengths of Android and Windows Mobile is that the programming language is tightly integrated with the operating system. For Java ME, interest is the number of supported device types. However, different API applications cause compatibility issues between vendors and even mobile device models. There is ongoing work to improve this through a project called JATAF (Java Application Terminal Alignment Framework). JATAF is collaboration between many large and influential companies in the mobile software market including Sun Microsystems, Orange, Nokia, Samsung and Sony Ericsson.

The Android operating system can be tightly linked through standard language development through the Open Handset Alliance and Windows Mobile by the sole owner of Microsoft as a driving force behind the platform's development. Android and Windows Mobile are currently integrated mobile systems as well as a softener and application development interface, while Java ME is an additional feature that makes it possible to run Java ME on limited hardware resources.

In this research, we have examined many key parts of the mobile development areas of Android, Windows Mobile, and Java ME. Through careful examination of areas such as performance aspects, implementation aspects and quality assurance results emerged. An interesting feature to note is the difference in the behavior of the unit test. Here, the Android system is the system that delivers the most surprising results when there is no easy-to-access general interface to test unit results on the device.

6- CONCLUSION

Windows Mobile and Android have full integration with the mobile operating system. This leads to a good integration between the development environment and the actual hardware. For Java ME, the main advantage is that there are a large number of third-party libraries available that will solve everything you may need. Android also offers a new and interesting way to distribute your app with the Android Market. Windows Mobile has the power to connect seamlessly to one vendor, Microsoft. This makes the development environment very stable, and the features available to developers enjoy high quality assurance.

In general, Windows Mobile and Android platforms provide a better development environment in our opinion. Java ME is still struggling with weak emulator support and incompatible applications. The Windows Mobile platform is a Microsoft product, so it relies on Visual Studio. Of course, Visual Studio also depends on the Windows operating system.

While all development environments share many common features, there are still individual features that make them unique. In permanent storage, both Android and Windows Mobile provide a better solution than Java ME. When you open a stream connection to connect to the network, however, Java ME and Android have the easiest and most elegant solution. In support of module testing, the Android development environment still has some issues that need to be categorized. However, large differences are distinguished when examining community environments, hardware capabilities and platform maturity.

The ease of development in different regions is likely to have a significant impact on the choice of development platform to create new applications for the enabling environment.

One interesting future research goal for this comparison is to include the iPhone. This would give a more complete picture of mobile phone platforms. In addition, expanding examples, including a more comprehensive performance test would be an interesting addition. These, however, are difficult task because of a wide range of different devices and phones that apply different platforms. Are all worthwhile in the future.

7- REFERENCE

[1] Gartner, “Gartner Says Worldwide Mobile Phone Sales Declined 8.6 Per Cent and Smartphones Grew 12.7 Per Cent in First Quarter of 2009.”

[2] T. Yamabe, K. Takahashi, and T. Nakajima, “Design issues and an empirical study in mobility-oriented service development,” *Proc. 1st workshop on Mobile middleware: embracing the personal communication device*, Leuven, Belgium: ACM, 2008, pp. 1-6.

[3] N. Milic-Frayling, M. Hicks, R. Jones, and J. Costello, “On the design and evaluation of web augmented mobile applications,” *Proc. 9th Int. Conf. on Human computer interaction with mobile devices and services,* Singapore: ACM, 2007, pp. 226-233.

[4] W. Du and L. Wang, “Context-aware application programming for mobile devices,” *Proc. 2008 C3S2E conf.*, Montreal, Quebec, Canada: ACM, 2008, pp. 215-227.
[5] M.T. Heikkinen and J. Still, “Benefits and challenges of new mobile service development in R\&D network,” *Personal Ubiquitous Comput.*, vol. 12, 2008, pp. 85-94.
[6] E. Harjula, M. Ylianttila, J. Ala-Kurikka, J. Riekki, and J. Sauvola, “Plug-and-play application platform: towards mobile peer-to-peer,” *Proc. 3rd international conference on Mobile and ubiquitous multimedia*, College Park, Maryland: ACM, 2004, pp. 63-69.
[7] A. Koller, G. Foster, and M. Wright, “Java Micro Edition and Adobe Flash Lite for arcade-style mobile phone game development: a comparative study,” *Proc. SAICSIT 2008* Wilderness, South Africa: ACM, 2008, pp. 131-138.
[8] M. Huebscher, N. Pryce, N. Dulay, and P. Thompson, “Issues in Developing Ubicomp Applications on Symbian Phones,” P*roc. Int. Workshop on System Support for Future Mobile Computing Applications*, IEEE, 2006, pp. 51-56.
[9] J. DiMarzio, *Android A Programmers Guide*, McGraw-Hill Osborne Media, 2008.
[10] G. Dubochet, C. Exton, and J. Buckley, “Computer Code as a Medium for Human Communication: Are Programming Languages Improving?,” *Proc. 21st Working Conf. on the Psychology of Programmers Interest Group*, Limerick, Ireland: 2009, pp. 174-187.
[11] S. Bohonos, A. Lee, A. Malik, C. Thai, and R. Manduchi, “Universal real-time navigational assistance (URNA): an urban bluetooth beacon for the blind,” *Proc. 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, San Juan, Puerto Rico: ACM, 2007, pp. 83-88.
[12] C. Benjamin, J. Harris, A. Moncrief, G. Ramsberger, and C. Lewis, “Naming practice on an open platform for people with aphasia,” *Proc. 10th international ACM SIGACCESS Conf. on Computers and accessibility*, Halifax, Nova Scotia, Canada: ACM, 2008, pp. 265-266.
[13] C. Greenhalgh, S. Benford, A. Drozd, M. Flintham, A. Hampshire, L. Oppermann, K. Smith, and C. von Tycowicz, “Addressing Mobile Phone Diversity in Ubicomp Experience Development,” *UbiComp 2007: Ubiquitous Computing*, 2007, pp. 447-464.
[14] Manhal M. Basher Hasan Aga1 & Sura Saad Basher2 & luay abdulwahid shihab3, Increasing The Efficiency Of Solar Energy Systems Using The Sun Tracking Method (Design And Implementation) , Webology (ISSN: 1735-188X) Volume 19, Number 2, 2022.
[15] Shihab, L.A. (2020). Technological Tools for Data Security in the Treatment of Data Reliability in Big Data Environments. International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies,11(9),1-13. http://doi.org/10.14456/ITJEMAST.2020.17