

Literature Review: Starting Mobile Application Development for E-Sports Portal

Hayoung Noh

*Computer Science Honours Programme
University of Cape Town*

13 May 2014

Abstract

Constant demand for mobile applications and relatively easy access to the market have been drawing developers into mobile application development. Derivco, a game development company initiated a website project E-Sports Portal. In this project, a mobile application is needed to be developed. Before jumping into developing, background research should be done in the area of software engineering, different platforms and design principles. This literature review aims to cover these areas in general and provide a starting point in developing mobile application.

1. Introduction

E-Sports Portal is a website project that is being developed to facilitate and organize e-sports leagues. In order for an e-sports team to upload their match results to the website, a mobile application is needed to capture the results and upload to the website server.

When starting a mobile development, few questions may be raised. What differentiates mobile application development and traditional application development? What options are available? What are the important design principles for mobile application? These questions are answered as each literature is reviewed in the following sections: software engineering issues in mobile development; comparison between native application and mobile web application; iOS and Android operating system; and human-computer interaction in mobile.

The following papers and a book have been chosen to be reviewed in each section:

Paper 1 - Software Engineering Issues for Mobile Application Development [12]

Paper 2 - Mobile Application Development: Web vs. Native [1]

Paper 3 - Smart Smartphone Development: iOS versus Android [4]

Book - Mobile Interaction Design [9]

2. Software Engineering Issues in Mobile Development

Software engineering for mobile application shares similar practices with traditional application. However, paper 1 pointed out some of the issues that needed to be addressed specifically in mobile development.

Paper 1 - Software Engineering Issues for Mobile Application Development [12]

As an introduction to going into the issues, the paper introduced several points that are different in mobile applications from traditional applications.

The first point was the potential interaction of applications between each other. Mobile devices have many applications from various sources with the possibility of interaction between them. Secondly, the sensor handling was pointed out. The accelerometers that respond to device movements, numerous touch screen gestures, global positioning system, microphones that are usable in applications other than voice calls, cameras, and multiple networking protocols are all in a single device, allowing many feature options for the application.

Another point was families of hardware and software platforms. Mobile applications are required to support multiple devices with various screen sizes and different hardware. In addition, different versions of the operation systems are released much more frequently than the embedded devices complicating the support. It was further pointed out that, the various devices added to the complexity of testing.

Lastly, the power consumption and battery life was brought up. The author mentioned that many aspects of an application may affect the device's power, draining the battery life of the device.

According to the survey and research done in the paper, mobile development tools and framework are mostly focused on the individual developer with the intention of creating an application as quickly as possible. The author argued that mobile application will become more complex for business critical uses and it will be essential to apply software engineering processes to assure the development of secure, high-quality mobile applications.

The following issues were raised by the author:

- Determining which functions should be present in a mobile version of a traditional application.
- Providing techniques that can assure maximum reuse of code among different versions.
- Determining comparable effort in building a native mobile application to a mobile web application.
- Determining if the mobile user interface requires different contextual design process to support a different set of use cases.
- Integrating the various forms of input and sensor data in application design.
- Determining if the synchronization techniques from traditional client-server computing would suffice in the potential loss of connectivity or battery power running out.
- Different application design depending on the speed of the network.
- Creating an application that can maximize battery life and resource usage.

The best practice suggested by the author was to follow agile methodology that can quickly adapt to changing user requirements and to follow development guidelines published by various platforms.

In conclusion, the author stated that there is a mobile “angle” to almost every aspect of software engineering research.

3. Web vs. Native Application

Before going into different mobile operating systems, it would be worth considering mobile web application versus native application. Depending on the type of application, mobile web application could simplify the development in terms of time and cost.

Paper 2 - Mobile Application Development: Web vs. Native [1]

The article started with the argument “developers cannot develop for every platform”. The authors further argued that the performance of native application will only be noticed for high image processing or 3D games. It was also pointed out that the native application languages

are generally known to be more complicated than Web application languages. One of the disadvantages for Web applications was the user interface code. The authors stated that the standard APIs for Web application interfaces are much weaker than the native applications. The scaling of Web interfaces on different platforms and devices was also raised as an issue.

User experience was another area to have an effect on both native and web application development. Users may have a different expectation for a native application to the web application. Web applications must be connected to the Internet the entire time the application is running but native applications can work offline as well as online.

In the area of performance, the size of the payload and the interpreting of code were discussed to determine how fast the web application can run. The conclusion was to favour data over decoration in the web application. The advantage of a native application was that performance issues are not related to the size of payload and the code is already compiled.

4. iOS and Android Operating System

Today, there are at least five important platforms (iPhone, Android, Blackberry, Windows Phone, Symbian) [12], but detailed examination of all the platforms will be impractical. Also, a large portion of the mobile markets in the world are currently iPhone and Android [13]. Thus, only the two platforms were reviewed.

Paper 3 - Smart Smartphone Development: iOS versus Android [4]

The aim of the paper was to compare two of the leading platforms in the world to contrast the difference in developing environments.

Comparisons of the two platforms are summarized in Table 1:

Table 1. Comparison of iOS versus Android

	iOS	Android
<i>Minimum Operating System Requirement for Development</i>	<ul style="list-style-type: none"> • Macintosh computers running Mac OS X 10.6 (Snow Leopard) 	<ul style="list-style-type: none"> • Windows XP • Linux • Mac OS X 10.5.8
<i>Language</i>	<ul style="list-style-type: none"> • Objective-C 	<ul style="list-style-type: none"> • Java (Dalvik VM) • Scripting (SL4A) • LogoBlocks (AppInventor)
<i>IDE</i>	<ul style="list-style-type: none"> • Xcode 	<ul style="list-style-type: none"> • Eclipse 3.5 (requires Android SDK plugin) • Can sign applications for release in the Android Market

<i>Garbage Collection</i>	<ul style="list-style-type: none"> ● Not available for performance reasons 	<ul style="list-style-type: none"> ● Available
<i>GUI Creation</i>	<ul style="list-style-type: none"> ● Xcode 	<ul style="list-style-type: none"> ● XML
<i>Simulator</i>	<ul style="list-style-type: none"> ● iOS SDK bundled ● GPS and Accelerometer not supported 	<ul style="list-style-type: none"> ● Android Virtual Devices (AVDs) that run on the Android Emulator ● GPS and phone/SMS interrupt signals that can be passed to the emulator through telnet connection ● Accelerometer, orientation and compass readings can be manipulated using OpenIntent's SensorSimulator
<i>Graphics</i>	<ul style="list-style-type: none"> ● OpenGL (support for 2D and 3D) 	<ul style="list-style-type: none"> ● OpenGL (support for 2D and 3D)
<i>Database</i>	<ul style="list-style-type: none"> ● SQLite 	<ul style="list-style-type: none"> ● SQLite
<i>Tutorial Resources</i>	<ul style="list-style-type: none"> ● Apple's technical documentation provides vast information 	<ul style="list-style-type: none"> ● Official tutorial development guide is available
<i>Reference Website</i>	http://developer.apple.com/iphone	http://developer.android.com

5. Human-Computer Interaction in Mobile

User interface is an important part of any software application that has user interaction. The book was reviewed for general human-computer interaction (HCI) and how HCI is different for mobile application.

Book - Mobile Interaction Design [9]

The main emphasis of the book was on the user satisfaction of the design and functionality. The focus of the book was on the user's needs, desires and capabilities. The authors argued that even if the industrial design and their aesthetic are appealing, if it does not address real user needs, it will be no use to the user.

Suggestion for capturing user needs was to apply disciplines such as psychology and sociology as well as looking at past successful designs. Also, constant testing and refinement of the design by engaging with the actual users were considered to be vital. User-centered design was emphasized because the general design principles may not work on a small screen size.

Two chapters were dedicated to understanding the users philosophically, and using various methods and activities to find out what matters for the users. Some of the methods included creating user scenarios, online inquiry and user self-reporting techniques. The most effective way was to get user feedback on the design through a prototype. Different forms of prototypes were discussed, from low-fidelity, paper-based sketches to complex pieces of software written for specialist hardware.

There were few design guidelines that stood out. The limiting of number of items in the menu to a maximum of seven was one of them. The studies presented in the book indicated that people can generally remember plus minus seven items in a menu. Use of icons (symbols), were suggested to replace text whenever possible to free-up space. Generally, pictures are easier to remember than text. Overviews for browsing were discussed in detail. The authors argued that well designed overviews are a powerful feature that can overcome restrictions of small screen sizes. Other design guidelines included quick navigation to frequent functions and limiting excessive scrolling or page-to-page navigation. Excessive scrolling is tiring on a small screen and users can get lost in a page-to-page navigation.

6. Conclusion

Four areas that can be helpful in starting mobile development for have been reviewed. Couple of issues were raised in the software engineering section without solutions. These issues should be considered for complex applications and there need to be further research in the future on the issues that were raised. However, given the requirement of capturing and uploading functionality for the E-Sports Portal application, traditional software engineering may suffice. In deciding whether to develop a native application or web application, the developer would have to consider user experience as well as time and cost constraints to make the best choice.

For native application platforms, Android and iOS were compared. Both platforms have advantages and the choice between the platforms will depend on the target market of the application. The HCI section provided insight in designing user-centered applications. User requirements and functionality should be the focus in the application design.

7. References

- [1] Charland, A., and Leroux, B. 2011. Mobile Application Development: Web vs. Native. *Communications of the ACM*. ACM, v.54. n.5. DOI=10.1145/1941487.1941504.
- [2] Church, K., and Oliver, N. 2011. Understanding Mobile Web and Mobile Search Use in Today's Dynamic Mobile Landscape. *MobileHCI '11: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM.
- [3] Forgue, M., and Hazael-Massieux, D. 2012. Mobile Web Applications: Bringing Mobile Apps and Web Together. *Proceedings of the 21st international conference companion on World Wide Web* (16-20 April). Lyon, France. DOI=10.1145/2187980.2188022.
- [4] Goadrich, M. H., and Rogers, M. P. 2011. Smart Smartphone Development: iOS versus Android. *SIGCSE '11: Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM.
- [5] Grønli, T., Hansen, J., and Ghinea, G. 2011. A Cloud on the Horizon: The Challenge of Developing Applications for Android and iPhone. *PETRA '11: Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM.
- [6] Grønli, T., Hansen, J., and Ghinea, G. 2010. Android vs Windows Mobile vs Java ME: A Comparative Study of Mobile Development Environments. *PETRA '10: Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments*. ACM.
- [7] Häkkinen, J., and Mäntyjärvi, J. 2006. Developing Design Guidelines for Context-Aware Mobile Applications. *Proceeding Mobility '06: Proceedings of the 3rd international conference on Mobile technology, applications and systems* (Article No. 24). ACM, New York. DOI=10.1145/1292331.1292358.
- [8] Huy, N. P., and Thanh, D. 2012. Evaluation of Mobile App Paradigms. *MoMM '12: Proceedings of the 10th International Conference on Advances in Mobile Computing and Multimedia*. ACM, 25-30. DOI=10.1145/2428955.2428968.
- [9] Jones, M., and Marsden, G. 2006. *Mobile Interaction Design*. John Wiley & Sons, Ltd.
- [10] Padley, R. 2011. HTML5 - Bridging the Mobile Platform Gap: Mobile Technologies in Scholarly Communication. *Serials: The Journal for the Serials Community* (0953-0460). UKSG, Volume 24, Supplement 3, S32-S39. DOI=10.1629/24S32.
- [11] Sletholt, M. T., Hannay, J., Pfahl, D., Benestad, H. C., and Langtangen, H.P. 2011. A Literature Review of Agile Practices and Their Effects in Scientific Software Development. *SECSE '11: Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering*. ACM.

- [12] Wasserman, A. I. 2010. Software Engineering Issues for Mobile Application Development. FoSER '10: *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM.
- [13] Market Share Statistics for Internet Technologies.
<http://www.netmarketshare.com/mobile-market-share> [Accessed on 12-05-2014]