International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 2
Article Received: *25 January 2021* ***Revised:*** *02 February 2021* ***Accepted:*** *26 February 2021*

Responsive Web Design for Cross-Platform Healthcare Portals

Manasa Talluri

Independent Researcher, Usa.

Abstract

The importance of corporate integration and cross-platform engineering in transforming healthcare information systems is examined in this article. Apps for mobile health (mHealth) have drawn more attention lately because of their potential to help patients with a range of ailments. In recent years, the mobile industry has grown significantly. The emergence of new methods allows developers to address issues that arise with the multitude of devices, screen form-factors, and platforms that the devices are operating on. Web developers may now create web apps that adapt to the screen form-factor of the device they are operating on thanks to the prevalence of responsive and adaptable designs in web application and website development. With wearables on one side and PC-tablet hybrids on the other, the lines between devices are becoming less distinct. A variety of app-enabled devices with varying capabilities and respective software ecosystems must be supported by future applications. Previous research on cross-platform app development focused on ideas and prototypes, comparing methods that are aimed at smartphones. In order to accomplish these goals, researchers must develop mHealth app prototypes using specialised software architectures. This research presents a notion for developing cloud-based mHealth applications for supportive care for chronic patients. The idea makes it easier to create apps that can be utilised for various target applications by integrating pre-existing software platforms and services. Additionally, by employing similar components across several mobile platforms, this development strategy facilitates app portability and scalability via loose coupling of services. In a case study showcasing a mHealth solution for endocrine hormone treatment (EHT), the outcomes are shown via the creation of native Android and cross-platform web applications. Results from alpha and pre-beta testing, a performance analysis technique, and an app usability assessment based on focus group comments are presented.

Keywords: - Cross-Platform Engineering, (mhealth), Patients, Responsive, PC-Tablet Hybrids, Software Architectures, Endocrine Hormone Therapy, App-Enabled Devices, Chronic Patient, Pre-Beta Testing.

I. INTRODUCTION

Our modern civilisation is distinguished not only by an ever-increasing quantity of information sources, but also by the availability of these sources via a variety of client platforms and devices. Particularly in the field of mobile devices with one or more purposes, there has been a significant increase [1]. There are an endless number of mobile devices that connect to information servers via various platforms. Despite a modest dip in growth in 2015, sales of smartphones have outpaced those of traditional laptops.

The delivery, accessibility, and management of healthcare services and information are being profoundly changed by mobile health (mHealth) applications, or apps. By 2025, the healthcare app industry is predicted to produce 111.1 billion dollars, having peaked in 2017 [1, 2]. Currently, about 83% of doctors use cell phones and medical applications to keep an eye on their patients' long-term health. Using mobile applications to disseminate, provide, or gather medical data is the foundation of mHealth. Any health services or information pertaining to health that is provided or improved via the Internet and associated technologies is referred to as mHealth, a developing nexus of medical informatics, public health, and commerce. The objective is to use information and communication technology (ICT) to enhance healthcare on a global scale. Research has assessed mHealth app use from the viewpoints of patients and medical professionals.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 2
Article Received: *25 January 2021* ***Revised:*** *02 February 2021* ***Accepted:*** *26 February 2021*

The digital revolution in the healthcare industry has resulted in the growth of specialised software programs, each of which is intended to handle certain facets of hospital administration and patient care. Modern healthcare practices now rely heavily on Electronic Health Records (EHRs), Picture Archiving and Communication Systems (PACS), Laboratory Information Systems (LIS), and Remote Patient Monitoring (RPM) systems [3, 5]. Nevertheless, these systems often function independently, resulting in data silos that hinder the exchange of vital patient data across various departments and care environments.

The necessity to handle two more or less incompatible platforms is not the only factor contributing to the difficulty of app development. Vendor-specific changes and device fragmentation make development, especially for Android, inconsistent. Furthermore, wearables and PC-tablet hybrids are erasing the distinctions between devices by extending computers into previously unconnected electronic assistants, watches, and even clothes [3, 5]. App-enabled gadgets are everywhere, and each one has unique features and quirks. Additionally, the various gadget types each have their own ecosystems and use situations. It is simple to see how difficult it would be to create an app that would work on a smartphone, in a car's system, and on a screenless smart home or Internet of Things (IoT) device. Future development methodologies will need to take into account this confluence of intelligent, user-targeted devices and hitherto unnoticed small-scale IT.

By using middleware, microservices architectures, and standardised APIs, cross-platform engineering provides an organised method for bringing these disparate systems together [3, 4]. Cross-platform engineering establishes the groundwork for a more responsive and interconnected healthcare environment by facilitating smooth communication across various software platforms, irrespective of their underlying technological stack. Enterprise integration strategies aim to provide end-to-end connection across the healthcare IT ecosystem in order to support these initiatives [6].

Healthcare organisations may create a common language for data interchange by using industry standards such as DICOM and HL7 FHIR for medical imaging. Furthermore, the rise of cloud-native infrastructures and Integration Platforms as a Service (iPaaS) has created new opportunities for developing scalable, vendor-neutral solutions that enable real-time data synchronisation across apps.

Patient outcomes have significantly improved as a result of cross-platform engineering and enterprise integration used to integrate healthcare information systems. These integrated solutions provide better informed decision-making and individualised treatment plans by giving medical personnel thorough, real-time access to patient data. The decrease in medical mistakes is one of the main advantages [3, 5]. Integrated systems lower the possibility of unfavourable drug interactions or inappropriate procedures by giving medical professionals access to a patient's whole medical history, including prescription drugs, allergies, and prior treatments. According to a research, fewer adverse drug events and prescription mistakes were linked to the use of integrated health information systems [3, 5].

Healthcare professionals may monitor patients' symptoms in real-time and take pre-emptive measures when needed thanks to cross-platform connectivity, which makes it easy to integrate data from remote monitoring devices into patients' electronic health records. During virtual consultations, telehealth platforms that are connected with EHRs and other healthcare systems provide a more thorough picture of the patient [5]. This connection improves the effectiveness and quality of remote treatment by enabling medical professionals to access pertinent patient data, place test orders, write prescriptions, and update records—all inside a single system.

A number of new technologies have the potential to completely transform integration efforts as healthcare continues to change:

- In integrated healthcare systems, these technologies are being employed more and more to improve decision support, predictive modelling, and data analysis [6]. Large datasets may be analysed for trends using AI and ML, which can also increase diagnostic precision and allow for more individualised treatment regimens.
- This distributed ledger technology might simplify the interchange of health information, increase data security, and improve interoperability while protecting patient privacy.
- The development of wearables and linked medical devices is producing enormous volumes of patient data. There are potential and obstacles for enhancing patient care and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 2
Article Received: *25 January 2021* ***Revised:*** *02 February 2021* ***Accepted:*** *26 February 2021*

monitoring when integrating this data into current healthcare systems [4–7].

- 5G networks are expected to make data transfer quicker and more dependable, which might greatly improve telemedicine capabilities and allow healthcare practitioners to share data in real time.

II. RESPONSIVE WEB DESIGN (RWD)

The definition and delineation of the word responsive itself, the name's genesis, its primary objectives, and the most often used methods—fluid grids, [8, 9], flexible images, and media queries—are all covered in this part in order to describe the term and idea of RWD. Additionally, it outlines responsive design techniques before providing a brief overview of popular RWD frameworks and platforms [10].

Fig. 1 Website design: responsive design utilising fluid grids and media queries to suit the smartphone (right); desktop version on a smartphone using simply fluid grids to adjust to the screen-width (left). [11]

It is important to use the word "responsive" while defining RWD. The concept of web design is thus presented before the definition of the term responsive [5, 6]. To supplement this section, the history of RWD and the distinction between responsive and adaptive web design are explained.

2.1 Responsive

The English adjective responsive has two meanings, according the Oxford Dictionaries [6, 7]. The first definition—reacting promptly and favourably, such as a service that is adaptable and sensitive to shifting societal trends—can also mean reacting easily and genuinely, such as our most engaged and receptive pupils. Reactive, receptive, and fast to respond are some synonyms for responsive [12, 13].

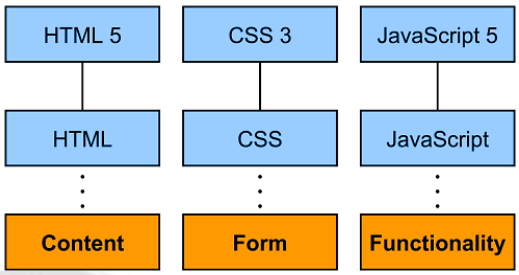

Fig. 2 In contemporary web design, the divides of information, shape, and functionality provided to the three most relevant standards—HTML, CSS, and JavaScript. [12]

2.2 Web Design

The first consideration when discussing web design in relation to RWD is the contrast between form, content, and functionality in website creation generally. Using HTML, the content—text, graphics, and other media—is specified [16, 17]. The graphical design and layout of the form are defined in distinct stylesheets, like CSS. The introduction of extra scripting languages allows for more functionality and involvement. Server-side scripting languages, such as PHP, Python, Perl, ASPNet, Cold Fusion, or JSP, are distinguished from commonly used client-side extensions, such as Flash, Silverlight, Java, and JavaScript. In today's web design, HTML 5, CSS 3, and JavaScript are often combined [18]. Figure 2 provides a visual representation of how these three criteria interact. These three standards are the most crucial in the context of RWD, while there are undoubtedly more accessible when considering web design.

2.3 HTML

W3C17 and WHATWG18 collaborated to produce HTML 5, a new standard that was designated as a Candidate Recommendation in December 2012. HTML 4, XHTML19, and HTML DOM20 Level 2 were all intended to be replaced by HTML 5 [16, 17]. Delivering rich information without requiring extra plugins was the goal. HTML 5 is also cross-platform as it is intended to function on PCs, tablets, smartphones, and smart TVs.

- The 2D drawing element
- The components for playing media
- Assistance with local storage

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 2
Article Received: *25 January 2021* ***Revised:*** *02 February 2021* ***Accepted:*** *26 February 2021*

- New components tailored to the content, such as, , , , [11]
- New form controls, such as search, email, URL, calendar, date, and time.

2.4 Cascading Style Sheets (CSS)

W3C is continuing working on the CSS 3 standard [14]. It is divided into modules. Although it has been divided into smaller sections, the previous CSS standard is still present [12, 13]. New modules have also been added. Among the most crucial components for CSS 3 are:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- UI

2.5 JavaScript

JavaScript is a popular programming language these days for adding functionality to webpages [14, 16]. JavaScript may validate data to verify user input and modify HTML components, properties, and styles (CSS) by manipulating the HTML DOM.

2.6 Responsive- versus Adaptive Web Design

A website may be represented on mobile devices in ways other than RWD. Adaptive Web Design (AWD) is one of the methods that is somewhat comparable to RWD. In his 2011 book of the same name, Aaron Gustafson first used the phrase "adaptive web design." The two approaches' greatest resemblance is that they both enable online content to be accessed across different screen form-factors and mobile browsers [15]. Enhancing the mobile user experience for visitors is the aim of both [15, 16]. The way they present the structures is where they diverge. In his post, "What is the difference between responsive vs. adaptive web design?" Ryan Boudreaux simplifies the distinctions into a single line. on April 11, 2013, on Web Designer:

"[...] adaptive design [...] will change to fit a predetermined set of screen and device sizes," whereas "[...] responsive web design [...] will fluidly change and respond to fit any screen or device size."

2.7 Importance of mHealth in chronic condition care

Comprehensive assessments of the status of mHealth services from a technology standpoint looked at the most important research papers and provided in-depth analyses of the new and current mHealth services and apps [16, 17]. To facilitate genuine and widespread patient-doctor interactions, the standard mHealth architecture makes advantage of the Internet and web-based services. Additionally, patient communication, access to medical education via online instructional materials, videos, disease-specific paperwork, and clinical trials for patients are all included in the mHealth applications now in use.

mHealth characteristics and challenges

Making sure that the features of mHealth applications are applicable to various mobile device platforms is becoming more and more crucial as mobile technology develops [16]. The following are the main features of mHealth:

(1) The uptake of applications by human populations;
(2) The form and accessibility of applications; [17],
(3) The setup and accessibility of network connections and Internet access, as well as the devices connected to people;

III. STATE-OF-THE-ART CROSS-PLATFORM DEVELOPMENT APPROACHES FOR MOBILE APPS

3.1 Mobile app development approaches

Generally speaking, mobile applications may be created as hybrid, web, or native apps [17]. Web applications are extremely portable across several platforms, operate on web-based servers, and may be accessed using desktop and mobile web browsers [18]. The time required for development and deployment cycles, as well as the expenses related to app creation, are decreased since these applications are compatible with a variety of platforms.

3.2 State-of-the-art cross-platform development approaches

The portability problems brought on by the fragmentation of mobile devices, platforms, and apps have been addressed in recent years [18, 19]. These initiatives have focused on frameworks and tools for CP

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 2
Article Received: *25 January 2021* ***Revised:*** *02 February 2021* ***Accepted:*** *26 February 2021*

development [18, 19]. By maximising code reuse and lowering the cost of necessary resources, these frameworks enable programs to be created and shared across several platforms. A comprehensive analysis first categorised a number of CP techniques that have surfaced by comparing web applications created with natively designed mobile apps and assessing web apps created using online-technology-based PhoneGap and Titanium Mobile [20].

IV. CLOUD-BASED MHEALTH SYSTEM ARCHITECTURE

As seen in Fig. 1, this section describes the use of a cloud-based mHealth infrastructure to enable native and CP web applications for supportive treatment of chronic conditions [21, 22]. In addition to offering services that can adapt to shifting customer preferences and demands, the architecture that is being described aims to lessen the effects of fragmentation brought on by the variety of devices (from tablets to smartphones) and operating systems. A private database (SQL or NoSQL) unique to each component or service makes up the cloud storage-based system. All system components may alternatively utilise Google Cloud Platform (GCP) or Amazon Web Services (AWS) as central databases (DBs) [20].

4.1 Evaluation metrics and results

In addition to presenting the measurement techniques, functionality testing, [20,22], and usability assessment findings, this section outlines the parameters that were used to evaluate the performance of the implementations that made use of the suggested mHealth design. The specs of the test devices used in the performance study are listed in Table 1, which was done to ensure that the applications worked on various platforms. All test devices were returned to their original factory settings, and we conducted our research in a separate location without any background programs.

Table 1 Specifications of the Test Device.

	Android	**iOS**
Device	Moto G5 plus	iPhone XsGHz
Operating System	Android 7.0	iOS 12.2
Ram Memory	4 GB	2 GB
CPU	Octa-core, 2 GHz	Hexa-Core (2×2.5 GHz Vortex + 4 × 1.6 Tempest)

outlines the instruments used to gauge each test device's native Android applications' and web apps' performance characteristics. Table 2. Using the Android Debug Bridge (ADB) shell, we ran the "dumpsys meminfo," [22], and "top" commands to monitor the native Android app's CPU and memory utilisation, respectively, on the Android test device.

Table 2 List of Performance Analysis Measuring Instruments. [21]

Metrics	**Web App**	**Android App**
CPU Usage	Key-metrics PM2, Android ADB, iOS Instruments, and Chrome Dev Tools and Node.js	Profiler for Android Studio, ADB (top)
Memory Usage	(Allocations and Time Profiler)	ADB (meminfo dumpsys)
Installation Size	Size of the PWA web app when it is added to the Home Screen on the test device (iOS Safari and Android Chrome browsers)	The Android test device's size may be seen in the settings.
Response Times	JMeter for Apache and Chrome Dev Tools	Custom Java application when loading on Firebase, Android device monitor (DDMS)
Loading Firebase Cloud DB	A shell script and custom Java application for loading Firebase	

The sizes of the installed applications and the installer APK Android bundle are shown in Table 3 [21, 23]. When viewed using the Chrome browser on Android devices and the Safari browser on iOS devices, the web application's size indicates its memory footprint.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 9 Issue: 2
Article Received: *25 January 2021* ***Revised:*** *02 February 2021* ***Accepted:*** *26 February 2021*

Table 3 Android and Web App Installation Sizes (in MB). [23, 24]

	Android Test Device	**iOS Test Device**
Native Android APK Installer (Download size)	7.89	-
Native android Installer App	36.96	-
Web app	1.09	1

The CPU utilisation for the online and native Android applications during video retrieval is shown in Table 4 as a proportion of the overall CPU capacity [25]. Like the native Android app, the web app server logic runs on Amazon EC2, but it retrieves movies from the movie's node via the Firebase API.

Table 4 CPU Usage of Web Apps and Android. [21]

	Android Wi-Fi	**iOS Wi-Fi**	**Android LTE**	**iOS LTE**
Native Android App	9	-	11	-
Web App	14.26	12.59	19.65	14.14

Based on the feature order, Table 5 displays the measured memory consumption figures, expressed in MB, when moving through the programs.

Table 5 Android and Web App Memory Usage (in MB). [22]

	Android Launch	**Android Nav Sequences**	**Android Launch**	**iOS Nav Sequences**
Native Android App	35.9	81.92	-	-
Web App	61.29	145.96	56.93	132.09

The startup timings for the web app on both the Android and iOS test devices, as well as the native Android app for the Android test device, are shown in Table 6.

Table 7 Average response times (in MS) for the native Android app in Wi-Fi and LTE modes. [23]

Load	**Android LTE**	**Android Wi-Fi**
0%	122.36	65.98
25%	146.96	109.69
50%	170.96	144.59
75%	251.96	186.99
100%	396.88	241.96

The average response times for this situation are shown in Tables 7–9 for the web app on an Android device, the native Android app [11], and the web app on an iOS device, respectively.

Table 8 Average response times (in milliseconds) for the web application on an Android test smartphone in both LTE and Wi-Fi modes. [12, 13]

Load	**LTE**	**Wi-Fi**
0%	164.36	98.65
25%	231.49	203.96
50%	300.69	228.78
75%	416.96	359.47
100%	512.96	496.89

Table 9 Average response times (in milliseconds) for the Web application on an iOS test device in both LTE and Wi-Fi modes. [13]

Load	**LTE**	**Wi-Fi**
0%	141.59	87.49
25%	201.22	142.69
50%	256.96	219.98
75%	321.65	329.65
100%	489.69	412.96

Furthermore, data interoperability throughout various system components is critical, particularly during the collection, storage, and use of data. For these situations, the architecture must be standardised throughout deployment [22]. A stand-alone application for use in research on health promotion is the case study. Therefore, without implementing standards like HL7, the architecture that was described used REST APIs, JSON schema, and basic web technologies to standardise communications amongst component services that need to interact with the user.

In the absence of a load on the cloud database, the measured performance metrics from the quantitative data gathered during the technical system assessment did not reveal any notable variations between the native Android and web-based applications that may have an impact on user experience [20]. Despite the underlying architectural differences in their designs, this result shows that the Android and CP web applications performed equally well for the present level of complexity.

V. CONCLUSION

Presenting a cloud-based mHealth system with supportive care features for chronic conditions is the aim of this project. This solution's design makes it possible to add additional modules or services over time, which makes it easier to provide a variety of healthcare features.

Additionally, the mHealth applications' usability and functionality were evaluated for use in research projects that sought to enhance medication adherence for chronic illnesses. To ensure the creation of a scalable and evidence-based intervention, we specifically created a cloud-based mHealth architecture with multilingual, interactive CP mobile applications that were deployed in the EHT environment to enable patients to self-monitor and manage their condition and symptoms. The provided research technique is used to verify and customise our design. The look and functionality of the applications we created for PWA and Android are almost the same.

We want to assess the performance of the native iOS application that makes use of the mHealth architecture that has been described in subsequent research. A virtual assistant on a mobile device might be included as a future architectural element to meet the demands of users with varying learning styles and help them navigate applications.

VI. REFERENCES

[1] Amatya, S., Kurti, A. Cross-Platform Mobile Development: Challenges and Opportunities. Advances in Intelligent Systems and Computing. 2013, Vol. 231, p. 219-229.

[2] Nebeling, M., Norrie, M. C. Responsive Design and Development: Methods, Technologies and Current Issues. p. 510.

[3] Nir Menachemi, Saurabh Rahurkar. (28 April 2018). The benefits of health information exchange: an updated systematic review. Journal of the American Medical Informatics Association, 25(9), 1259-1265.

[4] Niam Yaraghi (2015). An empirical analysis of the financial benefits of health information exchange in emergency departments. Journal of Medical Internet Research, 17(12), e305.

[5] Marten Smits ,Ewout Krame et al.(January 20,2015). A comparison of two Detailed Clinical Model representations: FHIR and CDA. European Journal of Biomedical Informatics, 11(2), 24-31.

[6] Guoqian Jiang a, Richard C. Kiefer et al. (August 2016). Developing a data element repository to support EHR-driven phenotype algorithm authoring and execution. Journal of Biomedical Informatics, 62, 232-242.

[7] Ryan Boudreaux. What is the difference between responsive vs. adaptive web design? TechRepublic. Apr. 2013.

[8] Pete Cashmore. Why 2013 is the Year of Responsive Web Design. Mashable. Dec. 2012. url:

[9] Thomas Claburn. Google's Secret Patent Portfolio Predicts gPhone. InformationWeek. Sept. 2007.

[10] Phonegap, 2019 (online), available: https://phonegap.com/ (accessed May 15, 2019).

[11] Titanium, 2019 (online), available: https://www.appcelerator.org/ (accessed May 10, 2019).

[12] T. Majchrzak, T.M. Gronli, Comprehensive analysis of innovative cross-platform app development frameworks, Proc 50thHawaii International Conference on System Sciences, 2017.

[13] M. Latif, Y. Lakhrissi, E. H. Nfaoui, N. Es-Sbai, Cross platform approach for mobile application development: a survey, in: Proc. International Conference on Information Technology for Organizations Development (IT4OD), Fez, 2016, pp. 1–5.

[14] O.L. Goaer, S. Waltham, Yet another DSL for cross-platforms mobile development, in: Proc. 1st Workshop on the Globalization of Domain Specific Languages, 2013, pp. 28–33

[15] D. Kramer, T. Clark, S. Oussena, MobDSL: a domain specific language for multiple mobile platform deployment, in: Proc. IEEE International Conference on Networked Embedded Systems for Enterprise Applications, Suzhou, 2010, pp. 1–7. D

[16] M. Usman, M.Z. Iqbal, M.U. Khan, A product-line model-driven engineering approach for generating feature-based mobile applications, J. Syst. Softw. 123 (2017) 1–32,

[17] J. Perchat, M. Desertot, S. Lecomte, Component based framework to create mobile cross-platform

applications, Procedia Comput. Sci. 19 (2013) 1004–1011.

[18] W.S. El-Kassas, B.A. Abdullah, A.H. Yousef, A.M. Wahba, Taxonomy of cross-platform mobile applications development approaches, Ain Shams Eng. J. 8 (2) (2015) 163–190,

[19] W.S. El-Kassas, B.A. Abdullah, A.H. Yousef, A.M. Wahba, Enhanced code conversion approach for the integrated cross-platform mobile development (ICPMD), IEEE Trans. Softw. Eng. 42 (11) (2016) 1036–1053,

[20] Node.js., Node.js, 2017 (online), available: https://nodejs.org/ (accessed Feb 10, 2017).

[21] Tobias Gebauer. Die 10 besten responsive Frameworks. German. TheWebdesign. Apr. 2013. url: http://www.the-webdesign.net/die-besten10-responsive-frameworks/ (cit. on pp. 20, 21.

[22] Ronen Halevy. The History of RIM and the BlackBerry Smartphone, Part 3: The Evolution of Color. berry review. Mar. 2009. url: http: //www.berryreview.com/2009/03/16/the-history-of-rim-the-blackberrysmartphone-part-3-the-evolution-of-color/ (cit. on p. 5).

[23] Eva Harb et al. Responsive Web Design. TU Graz. 2011. url: http: //courses.iicm.tugraz.at/iaweb/surveys/ws2011/g3-survey- resp-webdesign.pdf (cit. on pp. 10–12).

[24] Greg Hickman. What Small Businesses Need To Know About Mobile Marketing. url: http://mobilemixed.com/what-small-businesses-needto-know-about-mobile-marketing/ (cit. on p. 10).

[25] Daniel IoPCmag. Original Android Prototype Revealed During Google, Oracle Trial. PCWorld. Apr. 2012.