

4. Tomashevskiy, V. M. (2005). *Modeliuvannia system [systems modeling]*. Kyiv: Vydavnycha hrupa VNU, 351 p. (in Ukrainian).
5. Ryzhova, I. I. (1987). *Matematicheskoe modelirovanie pochvennykh protsessov [Mathematical modeling of soil processes]*. Moscow: Moscow University Press, 82 p. (in Russian).
6. Zahrai, Ya. M., Kotovenko O. A. (2007). *Modeliuvannia i prohnozuvannia stanu dovkillia [Modelling and forecasting the state of the environment]*. Kyiv: KNUBA, 120 p. (in Ukrainian).
7. Poluektov, R. A. (2001). *Teoreticheskie i prikladnye modeli dinamiki agroekosistem [Theoretical and applied models of agroecosystem dynamics]*. Retrieved from: <http://www.sbras.ru/ws/Lyap2001/2071/#9>. (in Russian)

Нехай Валентин Валентинович – аспірант кафедри програмної інженерії, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Нехай Валентин Валентинович – аспірант кафедри програмної інженерії, Чернігівський національний технологічний університет (вул. Шевченка, 95, г. Чернігів, 14027, Україна).

Nekhai Valentyn – PhD student of Department of Software Engineering, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: valentin_nehai@meta.ua

UDC 621.3.05

Volodymyr Kazymyr, Andrii Mokrohuz

INFORMATION TECHNOLOGIES OF MOBILE APPLICATIONS DEVELOPMENT

Володимир Казимир, Андрій Мокрогуз

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ МОБІЛЬНИХ ДОДАТКІВ.

Владимир Казимир, Андрей Мокрогуз

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

The paper describes the problems that developers have to face during development of mobile applications. One of the most important issues, which developers need to overcome are GUI consideration for small screens of mobile devices, multiplatform development, performance issues, security issues and battery life. Some Web applications also require offline mode to satisfy users' requirements. Developers of mobile applications have to think through all these issues. Three types of mobile applications have been considered alongside with their advantages and disadvantages. Despite all disadvantages, all three types take their niche in the world of mobile applications. The overview of mobile application development technologies has been made in the article. As the result, client-server architecture with HTTP(S) application level protocol was identified as one of the most popular and reliable nowadays.

Key words: mobile application, web application, HTTP, client-server.

Fig.: 1. Bibl.: 21.

Описано проблеми, з якими розробники доводиться стикатися у процесі розроблення мобільних додатків. Одними з найбільш важливих питань, які розробники повинні вирішити, – це графічний інтерфейс користувача для невеликих екранів мобільних пристроїв, розробка для декількох платформ, проблем з продуктивністю, питання безпеки та час роботи батареї без підзарядки. Деякі мобільні веб-додатки також вимагають автономного режиму роботи для задоволення потреб користувачів. Розробники мобільних додатків повинні продумати всі ці питання. Три типи мобільних додатків були розглянуті разом з їх достоїнствами і недоліками. Незважаючи на всі недоліки, всі три типи займають свою нішу в світі мобільних додатків. Розглянуто технології розроблення мобільних додатків. У результаті було визначено, що клієнт-серверна архітектура з протоколом програмного рівня HTTP є однією з найпопулярніших і надійних на сьогодні.

Ключові слова: мобільний додаток, веб додаток, HTTP, клієнт-сервер.

Рис.: 1. Бібл.: 21.

Описаны проблемы, с которыми разработчики приходится сталкиваться в процессе разработки мобильных приложений. Одними из наиболее важных вопросов, которые разработчики должны преодолеть, – это графический интерфейс пользователя для небольших экранов мобильных устройств, разработка для нескольких платформ, проблем с производительностью, вопросы безопасности и время работы батареи без подзарядки. Некоторые мобильные веб-приложения также требуют автономного режима работы для удовлетворения потребностей пользователей. Разработчики мобильных приложений должны продумать все эти вопросы. Три типа мобильных приложений были рассмотрены наряду с их достоинствами и недостатками. Несмотря на все недостатки, все три типа занимают свою нишу в мире мобильных приложений. Рассмотрены технологии разработки мобильных приложений. В результате было определено, что клиент-серверная архитектура с протоколом уровня приложения HTTP является одной из самых популярных и надежных на сегодняшний день.

Ключевые слова: мобильное приложение, веб приложение, HTTP, клиент-сервер.

Рис.: 1. Библ.: 21.

Introduction. Mobile devices differ from the standard PCs and still require special tools for software development. It is possible to distinguish problems dealing with GUI (graphic user interface), 3D graphics, web security, battery life, access to the file system etc.

Mobile devices have different screen resolutions and different hardware. Software also depends on manufacturer, so devices usually have different operation systems and applications. Therefore, every operation system requires different tools for application’s development, so developers have to choose platform and tools for development.

Applications for mobile devices require flexible solutions from developers of GUI, because web application for mobile device should not concede to web applications for PCs. Moreover, web applications for mobile devices should provide additional functionality and usability for users. This happens due to the fact, that mobile devices as usual have smaller screen size than PC, so developers have to think about design for web applications [1; 2].

We have distinguished some problems, which developers of mobile applications can face with.

There are many platforms available for mobile applications’ developers. Sometimes it is very hard to choose appropriate solution to implement mobile apps, especially if developers target couple platforms at once [3]. Fig. 1 shows distributions of popular operations system (OS) for mobile devices [4].

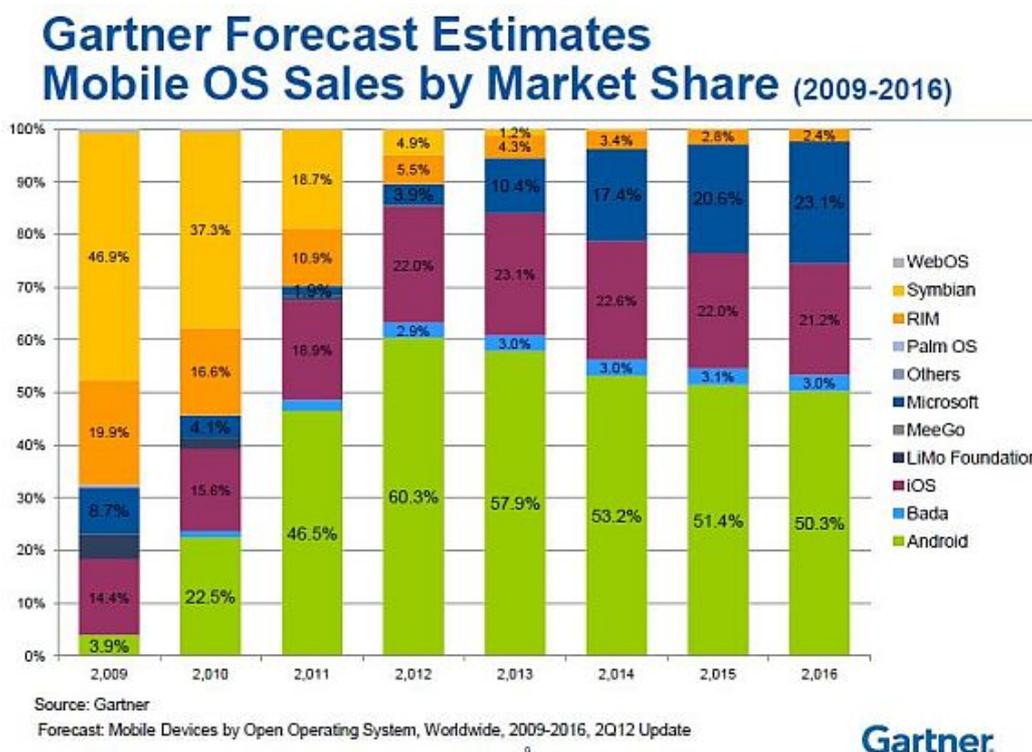


Fig. 1. Mobile devices OS distribution [4]

Choice of a technology to use becomes important issue on the way of mobile application developers.

Another issue that may arise is appropriate architecture and technologies for mobile applications. There are many types of mobile applications and architectures [5]. Obviously, each type requires different skills of developers and has restrictions in terms of OS usage on mobile devices. There is also different architecture can be hired behind each type of the applications. In the paper we show the difference between the mobile applications’ types and propose the appropriate solutions which can be applied in different situations. In addition, we consider some popular technologies to develop mobile applications [3].

Types of mobile applications. Nowadays mobile devices use for different purposes, for instance, users often use them for surfing internet, communication, gaming, multimedia etc. Hardware of mobile devices is becoming more powerful and users can use applications and games with high requirements for hardware. Developers need tools for 3D games and application development and these tools should provide opportunity to create high quality online applications as well. There are many game engines for different mobile platforms and one of them is game engine called unity 3D [6]. This game engine provides features to create high quality 3D game applications [7].

Another constraint dealing with 3D and multimedia application is battery life of mobile device. Developers should take care of economic using of battery and develop an application, which uses resources of mobile device efficiently [8].

Security of mobile applications and device is a very important problem. Web-based applications have a very high risk of attacks. These attacks target the most common vulnerabilities such as cross-site scripting call (cross-site scripting), the introduction of SQL (SQL injection), modification of the parameters (parameter tampering), fake cookie-files (cookie poisoning) and leaked. The traditional perimeter security systems such as firewalls and intrusion detection systems do not protect against attacks. Access control, firewalls, intrusion detection systems and intrusion prevention systems are an integral part of the security applications that protect the perimeter. However, these mechanisms are not fully protecting Web-based applications from attack on mobile devices, because security system are not very popular on mobile devices and many users of mobile devices do not even use firewalls or antivirus software.

These applications use Web-based technology and the principles of Web-user interaction with the application and can directly attack and bypass established security perimeter. Attackers know this, so most of the attacks are the direct attacks on Web-based applications. Application developers need to learn strategies to protect against attacks. They also have to take into account several factors that are the cause of a number of attacks, for instance: the majorities of Web-application developers are not security experts and may not know about the vulnerabilities. Many developers are not familiar with the best practices of Web-based applications protection. Often, the main priority is functionality, and security issues are resolved later by the modernization of the finished application.

The development environment is changing at high speed, including updates of the code itself, as well as infrastructure. Some of these changes are not checked and tested by professionals in the field of applications security.

The above factors dictate the rules that should be followed for any developer to write better code such as education, studying new solutions, integration of the testing in the development plan and identification of vulnerabilities in the early stages [9].

Creating applications that can run on different operating systems and different mobile devices can be a daunting task. As it was mentioned above, high demand for «smart» mobile applications requires high-performance hardware. One solution to this situation is to create Web-based applications, as they can run on mobile devices based on different platforms. In fact, you can do just one version of the application. However, there are two participants in communication, they are client and server. Web-applications developed for mobile devices are becoming increasingly popular. However, the instability of access to networks is a major obstacle to the use of Web-based technologies as part of the cloud infrastructure. Traditional Web-application simply does not work when there is no network access, but it is possible to provide opportunity for using application offline. Off-line Web-based applications are storing data in a database on the client side and mobile users can work with the functions of the cloud infrastructure or work in offline mode with a local host and a local database, synchronizing

TECHNICAL SCIENCES AND TECHNOLOGIES

data to the cloud when you resume access to the network. For instance, local database is implemented in the current version of Safari from Apple. It has a built-in database SQLite, which can be accessed from JavaScript via SQL-queries. This database stores business model of an application [10].

Normally mobile devices use Wi-Fi internet access points to access to the Internet. Hence, there is a significant problem dealing with battery life of mobile devices, because Wi-Fi consumes capacity of the battery. Manufacturers increase capacity of the battery for mobile devices to provide longer operation of the device. Another solution for this problem is changing Wi-Fi specification to incorporate power saving mode in Wi-Fi Technology to optimize power consumption of clients [11].

Many Mobile devices have the same set of functionality as PCs. No doubt, mobile devices can run all types of application such as native applications, web applications and hybrid applications. All these types of applications have advantages and disadvantages in comparison with each other, so developers should choose different types of applications for different purposes to achieve the best result.

Native applications

All resources give the same definition to the term “native application”. A native application is an application that has been developed for use on a particular operation system or platform or device [12]. Therefore, a native application could not be run on different platforms and sometimes a native application is developed for one version of operation system is not always able to run on another version of the same operation system (OS). This type of application is installed directly on mobile device by user and all the native application data is stored on mobile device as well [13; 14].

Native applications play a very important role and could not be replaced by other type of application in the nearest future. As it was mentioned above, a native application is developed only for one platform and OS. It is an advantage and a disadvantage at the same time, because native application can use all device-specific (platform specific) hardware and software. For instance, it is possible to use camera, GPS system, microphone, file system etc. [9]. Another advantage is that native apps are very fast and can produce the best users’ experience. Games for mobile devices are a very good example. Native apps have high level of security, because usually native app does not have access to the Internet, so it decreases potential risk of attacks.

However, orientation on one platform has a lot of limitations. One of them is number of native apps for platforms. Normally it is required to develop a native app for each platform and it is not easy task, because developing requires knowledge in objective C, Java and C#. Therefore, developing requires more resources and time. Native app should be installed and updated by user and it is not very convenient, because user can ignore update and continue using old version of application. Before users can install application it should be uploaded onto marketplace or app store and it requires additional time and costs from developers.

Web applications

A web application is an application that is stored on remote server and users have access to the application over the Internet via browser interface. Web apps are very good solution for different types of devices and platforms [15]. “Build once, run anywhere” concept is a very promising and can reduce resources and time for developers. Web application does not require separate version for each type of the device platform they are crossplatform and can be created only once [16]. There are a lot of tools and technologies for developing web applications, so, development is becoming more convenient, rapid and robust. As it was mentioned above, a web app access is provided via browser, so users do not need to install application on their device. Users need only access to the internet and appropriate version of

browser on the mobile device. Developers can update web application and all users get new version of it.

Web apps could not fully replace native application for mobile device, because it is impossible to get access to the native platform features and some securities issues arise. Users for accessing web apps require the internet connection and using of web apps remains impossible without it. Web app developers found solution to this problem. It was decided to provide offline access to users of web apps and use local data storage with synchronization of the data.

Despite these entire disadvantages web applications' development is improving very fast because of HTML5, friendlier version of web language and the hard work of W3C [17; 18].

Hybrid applications

A hybrid application is an application that combines elements of native and web applications. Therefore, hybrid app development combines advantages and disadvantages related to web and native apps [19]. Developers have to use toolkits for hybrid application development. For instance, Sencha's Complete Team, Titanium, Mobile frame, PhoneGap are toolkits which allows development of hybrid applications. A hybrid application is a web application which is built using HTML5 and java script, but it is wrapped into native container which gives access to native features of the mobile device.

A hybrid application can function without internet connection and store data locally. As it was mentioned above, it has access to platform features and web services. Developers of hybrid applications just need knowledge in HTML and Java Script to implement hybrid application. Basic knowledge in platform specific language is also required, but it is not so important. Another benefit is support of hybrid application by all platforms, so, it is reduces resources required for development.

However, there are many drawbacks dealing with hybrid applications. Despite opportunity of using native features of the platform, hybrid applications have a very poor performance while accessing camera, file system etc. Native apps have better performance and provide more secured access. Another problem is graphic user interface (GUI), because hybrid applications do not have all prebuild effects like touch etc. It is very hard to develop 3D hybrid applications, because performance of these apps does not allow using 3D acceleration on 100%. However, hybrid applications have their place among other types of applications for mobile device [20].

Technologies to develop mobile applications. Nowadays more and more applications have client server architecture and it is very hard to find piece of software, which does not have server part. Client and server software parts should have opportunity to communicate with each other. There are many technologies to provide communication between client and server, for instance, RMI (remote method invocation), XML-RPC, JSON-RPC, CORBA etc. It is also possible to use low level API based on sockets to provide interaction. However, client and server usually use different API, technologies, for instance, server can be implemented in Java language, and clients are using .Net, Objective C technologies or Android API. It is required to provide communication between these pieces of software using unified technology to avoid implementation of the same business logic in different languages. Of course, developers could implement different types of server software to provide functionality for each type of client, but it is not efficient and takes a lot of time.

Web service technology allows providing communication between different types of client and server software and server software could be implemented only once. W3C definition of web service is - "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web

TECHNICAL SCIENCES AND TECHNOLOGIES

service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards” [21]. Web service is an application component, which uses open protocols for communication and can be used by different applications and bases on XML or JSON. A web service based system with client server architecture is a very useful for developing application and has several advantages compared to other technologies. Web services can work outside the local network and allows using different languages to access web services. Client application is free to choose service and make a request anytime, because web services are always available. As it was mentioned above, web services provide data for clients, which were developed for different platforms. Web services use standard web technologies and this makes them very easy and powerful technology for developing. Security of web services is also one of the main advantages and security is already build-in, for instance SSL.

Web services use plain text protocols, so amount of data transferred through the network increases significantly. Another disadvantage is an impossibility to send data from server to client without request from client’s side. Asynchronous requests and long polling allows doing that, but it is not always applicable. Web services use HTTP and HTTPS, which are stateless, so client and server know nothing about each other when no data is transferred. Server could delete all information about client if client does not send any information. Nevertheless, HTTP(S) is the most popular protocol to provide communication between client and server applications.

Summary. Mobile device applications have some traits, which other types of applications does not have. Mobile application developers solve big amount of problems to create a mobile application. However, they should pay attention on GUI, which has limitations because of small screens, security for standalone and web based applications, tools they use to create mobile applications, performance issues which mobile device can face with and other limitations, which are typical for mobile devices.

Appropriate type of a mobile application is an important thing to consider. Three types of mobile application have been defined in the article. These types are a native application, a web-based application and a hybrid application. Developers can use any of the types, however wrong decision can delay development.

All these types of application can be a part of the client-server architecture. Communication between client and server can be provided by HTTP protocol.

References

1. Vithani, T. & Kumar, A. (2014). Presentation 5. A comprehensive mobile application development and testing lifecycle. *IT Professional Conference (IT Pro), 2014*, Gaithersburg, MD, pp. 1–3.
2. Mahmud, D.M. & Abdullah, N.A.S. (2014). Mobile application development feasibility studies: A case study in Universiti Teknologi MARA. *Open Systems (ICOS), 2014 IEEE Conference on*, Subang, pp. 30–35.
3. S. Vojvodić, M. Zović, V. Režić, H. Maračić and M. Kusek, "Competence transfer through enterprise mobile application development," *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, Opatija, 2014, pp. 448–452.
4. Gartner, Inc. is an American research and advisory firm providing information technology related insight. Retrieved March 23, 2016, from: <http://www.gartner.com/technology/home.jsp>
5. Barkan, S. (2016). *Native Vs Mobile Web Vs Hybrid Apps*. Retrieved March 14, 2016, from <http://mobilenext.net/native-vs-mobile-web-vs-hybrid-app/>.
6. *Mobile Game Engines for iPhone, iPad, & Android: Cocos2D, Corona, ShiVa, Torque, Unity3D, & UNREAL*. Retrieved April 3, 2015, from <http://www.burtonsmidiagroup.com/blog/2010/06/game-engines-for-iphone-ipad-android-cocos2d-corona-torque-unity-3d/>.
7. *Unity 3D official site*. Retrieved April 13, 2015 from <http://unity3d.com/>.
8. *Standards for Web Applications on Mobile: February 2015 current state and roadmap*. Retrieved April 10, 2013, from <http://www.w3.org/2012/02/mobile-web-app-state/>.
9. *Mobile Device Security*. Retrieved April 11, 2015, from <https://wiki.internet2.edu/confluence/display/itsg2/Mobile+Device+Security>.
10. *My Safari Browser SQLite Database Hello World Example*. Retrieved April 20, 2013, from <http://www.bennadel.com/blog/1940-My-Safari-Browser-SQLite-Database-Hello-World-Example.htm>.

TECHNICAL SCIENCES AND TECHNOLOGIES

11. Keshav, K., Indukuri, V.R., Venkataram, P. (2012). "Energy efficient scheduling in 4G smart phones for Mobile Hotspot application." Proceedings of the *Eighteenth National Conference on Communication (NCC)*, 2012/2/3, pp.1-5.
12. *Definition of: native application*. Retrieved April 20, 2015, from http://www.pcmag.com/encyclopedia_term/0,2542,t=native+application&i=47651,00.asp.
13. *Mobile applications: native v Web apps – what are the pros and cons?* Retrieved April 20, 2015, from <http://mobithinking.com/native-or-web-app>.
14. *Native app*. Retrieved April 20, 2015, from <http://searchsoftwarequality.techtarget.com/definition/native-application-native-app>.
15. *Web application (Web app)*. Retrieved April 20, 2015, from <http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>.
16. *Native Apps vs. Mobile Web: Breaking Down the Mobile Ecosystem*. Retrieved April 20, 2015, from <http://www.wired.com/insights/2012/11/native-apps-vs-mobile-web/>.
17. *Why Web Apps Will Crush Native Apps*. Retrieved April 20, 2015, from <http://mashable.com/2012/09/12/web-vs-native-apps/>.
18. *Native App vs. Mobile Web App: A Quick Comparison*. Retrieved April 20, 2013, from <http://sixrevisions.com/mobile/native-app-vs-mobile-web-app-comparison/>.
19. *Hybrid application (hybrid app)*. Retrieved April 20, 2015, from <http://searchsoftwarequality.techtarget.com/definition/hybrid-application-hybrid-app>.
20. *Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options*. Retrieved April 20, 2013 from http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options.
21. *Web Services*. Retrieved April 14, 2015 from <http://www.w3.org/DesignIssues/WebServices.html>.

Kazymyr Volodymyr – Doctor of Technical Sciences, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

Казимир Володимир Вікторович – доктор технічних наук, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Казимир Владимир Викторович – доктор технических наук, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

E-mail: vvkazymyr@gmail.com

Mokrohuz Andrii – PhD student, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

Мокрогуз Андрій Олександрович – аспірант, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Мокрогуз Андрей Александрович – аспирант, доктор технических наук, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

E-mail: myworkingadr@gmail.com