

## **Social features of mobile application development**

A.I. Mostyaev<sup>1</sup>, Postgraduate Student, *reistlin12@gmail.com*

<sup>1</sup> *Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Moscow, 119991, Russian Federation*

**Abstract.** Modern mobile application development technologies evolved at an unprecedented rate seeking for newer and newer user demands. Developers are working hard to not fall behind and try to maintain the popularity of their applications in all kinds of ways, introducing new amazing features and options. The paper describes the most common features of mobile applications and their support comparing with desktop analogs. Considering these features during mobile application development and maintenance should eliminate a misunderstanding between mobile users and mobile developers. It will be useful for both sides.

The paper starts with a quick overview of the mobile application history, giving a picture of the evolution speed in the industry. Further, it describes the most common mobile application features. A special attention is given to both technical details and usability of applications. The following features are marked: close integration with an operating system, short sessions, internet service integration and variety of mobile devices. The development features are a special life cycle and integration of third-party internet services. A special attention is paid to the quality of application localization, localization features for some countries and working with text and visual data in application stores.

In conclusion, the paper gives a list of requirements to a modern successful application. The author also mentions an interesting fact that development features of these applications are related to the current social trends.

**Keywords:** *mobile applications, cross-platform applications, application architecture, application development, application support.*

The role of a personal computer as the only tool to browse and communicate via the Internet has significantly decreased. Nowadays, there are a lot of new devices: smartphones, tablets, smart clocks and TVs, IoT, etc. The majority of people use several devices and want to access familiar applications from any of them.

Many companies pay more attention to mobile access to their applications (either with an optimized web-version or a special mobile app). In this case, the demand on mobile development grows at an unprecedented rate. For example, last year 88 % of Facebook users accessed the network via smartphones and tablets [1]. Even in 2017 the amount of Android internet users exceeded Windows users [2]. According to the SlashData analysis company, the amount of mobile developers exceeded desktop developers in 2018 (11.5 millions versus 10.2 millions) [3].

But a lot of software vendors (and even developers) think that mobile application development and usage doesn't differ from the corresponding processes of the desktop software. This delusion causes bad users' feedback and reviews, nonfulfillment of financial plans in companies and unnecessary problems to the developers as they need to perform unusual and sometimes even unnecessary tasks. This leads to users' and vendors' disappoint-

ment and may cause distrust in mobile applications, as well as the feeling of the impossibility to use them full-scale comparing to desktop analogues.

The reason is a difference between mobile and desktop application use cases. It can be turned into an advantage for users if we take into account the difference in design and development stages of the a mobile application. The success of a mobile application may cover the success of its desktop version.

The paper describes basic features of mobile applications. If we recognize them, it should eliminate a misunderstanding between mobile users and mobile developers. A special attention is given to the interesting fact that mobile application development features depend on the current social trends and feelings directly.

### **About mobile development**

Mobile application development is a new topic for scientific research. Despite this fact, the amount and complexity of mobile app functions consistently grow. This issue can involve the research on common computer science tasks, such as image and sound recognition, artificial intelligence and parallel programming, as well as specific ones,

such as adaptive interface research and user interaction types.

The first mobile phones appeared in 1980s–1990s in the USA. They were intended only for calling and didn't include other functions. But in 2000s a volcanic mobile device evolution has started. The Ericsson company has released the first smartphone combined various functions and additional tools. Later, in 2007–2008 the first versions of the currently most popular mobile systems has appeared: iOS and Android. An incredible mobile revolution has begun with the development of these systems [4].

Nowadays, we cannot imagine a modern application without a mobile version. All latest technologies has become available on mobile devices almost instantly. For example, Virtual Reality (VR), Augmented Reality (AR), Internet of Things (IoT), neural networks, voice input, virtual helper (Siri on iOS, Google Assistant on Android), object and text recognition (in Google Translate), etc. If the first smartphones were considered as tiny computers with limited options and a casual user interface, modern smartphones can surely compete with desktop functions [5].

### Features of mobile applications usage

**Close integration with an operating system.** Mobile applications are closely integrated with an operating system. They can send and receive data from other apps using the system, handle system requests and receive service signals, such as changing device orientation, wireless network connection or disconnection, etc. *Application Programming Interface* (API) is a special feature to perform system requests. Mobile systems provide bigger and more flexible APIs than desktop ones. Their interface includes methods for a complex user interface and advanced device management, such as gesture support and an accelerometer. API is made at a high professional level, so developers rarely need to change it for their specific needs [6].

API makes it possible to achieve several goals at once. Developers create applications with a typical interface and system styled management. The system code implementing API is loaded only once, so it makes applications smaller and reduces the amount of consumed resources. The necessity to learn API first makes mobile development more advanced, so training mobile developers is more expensive for the companies [7].

**Short sessions.** The interaction process with a mobile application has a lot of switches between active applications. For example, between a

browser and a social network application or between a game and a mail application. It results in a short sessions for using applications, as users open them for a quick information view – new messages, quick Internet search, browsing current a location for route calculation, quick game in a public transport and so on. Short sessions create the following advantages for applications.

First of all, mobile applications should have a convenient user interface with not very deep navigation structure that allows reaching a necessary screen as quick as possible after an application launch [8].

A user should be able to reach common applications functions, like message sending, note making or music track switch. The feature affects an internal application structure, as well as external system integration. Mobile systems provide quick access to applications using the following ways:

- widgets displaying on a home screen or a quick menu with a small information;
- text or graphic notifications with optional quick actions, always available in the notification bar or on a lock screen.

Second, mobile applications should correctly handle possible minimizing and maximizing, preserving a user data. An application should be able to save data at any time for further restoring. It can be either web page loading, movie view or a bank operation.

**Internet service integration.** Internet service integration has become a usual option for mobile applications. Services can be used even in the applications that do not have a server side. There is no need in a stable and active Internet connection, usually short periods to send a bunch of the most important data are enough [9].

The most popular type of an Internet service is a user analytic tool. An application integrates a single third-party library to collect some common data about its users. For example, geography, language, device model, session time, screens visited, etc. All the data is sent to the tool's own server and a developer can view and analyze collected data using the provided web service or a special API.

Another popular type of internet service usage is social networks integration. Social networks are closely connected with mobile devices, their applications are provided with phones, some customers buy new devices just to have an access to the latest social application features. Considering this fact, mobile developers actively use social integration features, such as sign in via a social network or message sending. Messages sent can include user achievements description in a game (a new level

reached, a special artifact gained), recommendations for installation and many more limited only by fantasy of developers.

Also there is a way to use an Internet service for data storage. As many users have several mobile devices, they need an access to their information from several devices as easy as possible. Such information might be application settings, user statistics (game progress, notes list, etc.), some files and folders. The information can be stored in a special cloud storage services. The most important of them are Dropbox, Google Drive, iCloud and OneDrive. Users become more and more familiar with such services and expect they will be supported in new applications. Even mobile operating system vendors provide more and more options to store user data in their cloud storage spaces. The latest versions of Android and iOS support automated storage of application data in a cloud provided by the manufacturers (iCloud from Apple and Google Drive from Google).

**A variety of mobile devices.** Another feature of mobile devices supported by a mobile operating system is their variety. There are mobile phones of different sizes, tablets with large screens and small smart watches. Developers have to keep in mind this variety and design a user interface accordingly. Even if an interface can be scaled depending on a screen size, the best choice is to change a screen layout. At the same time, large screens show more information and small ones divide it in several screens. A special programming system is necessary to support such development style.

We introduced a special term “fragment” to describe an interface for several device screens. Each fragment is a logically completed element (containing usual visual elements) with its own life cycle. It provides some functionality for a user. An application screen can include only one fragment or combine several ones. For example, if an application has a fragment with a list of elements and a fragment with an element description, the following separation can be proposed. A list fragment should be displayed on a full screen on phones and a description fragment should be also shown on a full screen when a user clicks a list item. On a tablet, both fragments can be shown on the same screen, a list fragment can be placed on the left of the screen, while a description fragment is on the right. When a user clicks a list item, description fragment is updated according to the item; a screen layout is not changed in that way [10].

It should be also noted, that a small screen size significantly increases the work for designers and ergonomic specialists. Each visual element and

each user gesture should be chosen with care. Not only elements size should be considered, but also accessibility of the elements on the screen [11].

### Features of mobile application development

**Life cycle.** Besides the differences in application usage, there are differences in their development. We consider the main ones.

Mobile applications have a special life cycle compared to desktop applications. The main difference is in frequent updates, which are easily handled and delivered by application stores. The update mechanism is familiar to users and significantly affects an application life cycle.

1. Due to frequent updates, a life cycle spiral tightens up as they require all development stages for each release. Usually, updates provide new features alongside with fixes. These new features get feedback from users instantly, so the features may be corrected in the next versions and improved in the way users demand. That may reduce expenses of changing completed features after one big release.

2. Mobile application development is one of the most fast-moving industries. New devices, new tools and operating system updates appear regularly. This requires constant attention, testing in new conditions and adding new technologies.

3. Mobile device resources are limited, so before the release and especially after developers have to improve the performance of an application, optimize memory and battery consumption.

**Internet service usage.** Partial information storage on a server is a good solution to increase application functionality without using device resources. This way is widely used in mobile application development. A notable feature is that a big amount of functionality is performed by standard or free services. Network connection is used to collect user statistics, crash reports, in-app billing and advertising.

Special internet services are used to collect data about the users – Analytics and Crashlytics. An application sends usage data to the service servers in run time. Information can be standard or advanced. Standard information includes application usage data (launch times, frequency and duration of each screen session, transitions between screens), device data (manufacturer, device model, screen resolution, operating system version) and anonymous user data (location and language). Alongside with the standard information, an application can send its own data to the service. For example, applica-

tion settings, amount of clicks a certain button, list scroll intensity, browsed data identifiers. Service servers handle data and can generate various reports (amount of application users, most-viewed screens, users geography, etc.). Reports can be visualized by a lot of charts and filters.

A very important function is In-App Billing mechanism. There are a lot of free-to-play applications; a user can download the application for free and try a demo version. If a user wants to access all application options and content, he has to purchase an in-app items. The purchase is performed within the application and a user does not have to download another installation package. So purchase and unlock procedures are performed within an application. Application stores provide tools for In-App Billing and usually their policy denies using other billing services.

Another function is leader boards and achievements tracking. Leader boards can be worldwide, regional or available only to a group of users (for example, to some club). Achievements are granted for some important game actions defined by the developers. There are special services for achievements and leader boards handling all the operations with the data. Applications just have to send user data to a service using special request API.

Besides, we should note advertising services. Many applications use such services to generate revenue via ad impressions and clicks. The services are also easy to integrate. They provide several advertisement types, such as banners, interstitial and video ads [12].

**Application stores.** Every mobile platform has its own application store where users can find and download applications. Google Play is an Android application store, App Store is used on iOS devices. There are third-party stores and special forums besides the official application stores, but they are used by minority of users according to the statistics. Thus official stores are almost monopoly providers for mobile platforms, therefore it is important to promote an application in a store.

In addition to an application package, the following information needs to be provided in an application store: application category, title, description, screenshots of main screens, demo video, privacy policy, age restrictions and county availability. Application life is highly dependent on that information, as internal store search uses the data to perform search requests.

Each application has its own rating in an application store, which is based on user scores and reviews. Rating is a very important factor influencing users' download decision. Application stores

provide options to reply to user reviews, so a developer can help users to solve problems or collect feedback. A lot of users are familiar with Internet conversations and expect speed and good answers from developers. The developers can have special blogs or forums to provide users with the latest news and information about various application features. Users can also give feedback on such forums. Conversations with users has become very important in application success.

Another application store feature is their worldwide accessibility. Once an application is uploaded to an application store, it becomes available for downloading by a user from any country where a developer allows distribution. That requires more work from developers on application translations, but gives a chance to significantly increase potential application audience.

**Handling language features in applications.** Application translation is not always a simple translation of strings; sometimes an application needs to be optimized taking into account language features and even people culture.

For example, in Turkish it is customary to use common at first and then particular, not the other way around. If in English we commonly use "4 from 10", the Turkish version is "10'den 4". To implement this feature, there is a special format parameter in string patterns in an application source code. Each format parameter can have an ordinal value. So the original string will be "%1\$d from %2\$d" in English and "%2\$d'den %1\$d" in Turkish. The features are necessary to consider at the development stage to provide translation compatibility in the feature.

A special example is an Arabic language, which uses right-to-left writing system. All application visual elements shown on the left, should be shown on the right in Arabic. Even icons are needed to be mirrored from left-to-right view to the right-to-left view. Development tools support common sources for left-to-right and right-to-left layouts, but the feature should to be considered as early as possible [13].

**Application store culture features.** Apart from an application, application store material are also a subject to be translated. They have their own features. Text materials such as a title and description have to be translated according to language structure and grammar. Graphic materials like an application icon and screenshots can be adapted too. All screenshots should show the application in own language. All possible descriptions on screenshots also have to be translated.

Graphic information should also consider region culture. For example, users from Asia like

screenshots with big amount of information shown. There is a different situation in European countries. Another example is an application icon. Korean users prefer games with a female as a main character. Thus displaying a female person on an application icon makes an application more attractive for them.

Application stores also provide different prices for countries. Some of them are even in local currency, the rest are shown in American dollars. Local prices may vary depending on a country live level and culture. For example, “.99” ending is widely used in Europe and America, while the Asians typically use rounded up prices. Some numbers like 24, 42 and 49 are considered unlucky in Asian countries, so developers should avoid them in Asian prices.

### Conclusion

The raise of mobile applications industry is closely connected with the society culture. A lot of popular apps try to follow modern social trends, resulting in great success and evolution. Let's list the main requirements for a modern and successful application again.

An application should consider a dynamic life style of people. It should provide a completed logically full operation in a few minutes (or even seconds).

New functions should be included into an application once it becomes popular.

An application should be universal and convenient, provide comfortable user experience on all their devices.

An application should use as much system abilities as possible to achieve an application goal making user experience more familiar and convenient.

An application should effectively use various internet services. However, it should also support an offline mode.

Globalization demands language and cultural adaptation for various countries. If an application supports a small language list, its possible audience is significantly smaller.

User feedback might significantly affect application popularity. The amount of downloads, ratings, developer answers to reviews crucially affect users' download decision. Low-quality applications quickly fall to the bottom of search results and have no chances to become popular.

Following the requirements make mobile application development more expensive. However,

it gives a unique chance to become famous worldwide helping people to live in their style.

### References

1. *Number of Monthly Active Facebook Users Worldwide as of 4th Quarter 2018 (in millions)*. Available at: <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/> (accessed February 04, 2019).
2. Android's popularity eclipses Windows among internet users. *The Verge*. Available at: <https://www.theverge.com/2017/4/3/15159320/android-windows-internet-usage-statistics-competition> (accessed February 04, 2019).
3. *Developer Insights. SlashData – The Leading Analyst of the Developer Economy*. Available at: <https://www.slashdata.co/reports> (accessed February 04, 2019).
4. Cho Y.C., Jeon J.W. Current software platforms on mobile phone. *ICCAS Conf.* 2007, pp. 1862–1867. DOI: 10.1109/ICCAS.2007.4406649.
5. Lin F., Ye W. Operating system battle in the ecosystem of smartphone industry. *Symp. on Information Engineering and Electronic Commerce*. 2009, pp. 617–621. DOI: 10.1109/IEEC.2009.136.
6. Hammershoj A., Sapuppo A., Tadayoni E. Challenges for mobile application development. *Proc. 2010 14th Intern. Conf. on Intelligence in Next Generation Networks (ICIN)*. 2010, pp. 1–8.
7. Kamthan P. Towards high-quality mobile applications by a systematic integration of patterns. *J. of Mobile Multimedia*. 2008, vol. 4, no. 3, pp. 165–184.
8. Franke D., Kowalewski S., Weise C. A mobile software quality model. *Proc. 2012 12th Intern. Conf. on Quality Software*. 2012, pp. 154–157. DOI: 10.1109/QSIC.2012.49.
9. Wasserman A.I. Software engineering issues for mobile application development. *Proc. FSE/SDP Workshop on Future of Software Engineering Research*. 2010, Santa Fe, New Mexico, USA, pp. 397–400. DOI: 10.1145/1882362.1882443.
10. Ali N., Solís C., Ramos I. Comparing architecture description languages for mobile software systems. *Proc. 1st Intern. Workshop on Software Architectures and Mobility*. 2008, Leipzig, Germany, pp. 33–38. DOI: 10.1145/1370888.137089.
11. Charland A., Leroux B. Mobile application development: web vs. native. *ACM Conf.* 2011, pp. 49–53. DOI: 10.1145/1941487.1941504.
12. Zakharov V., Malkovsky M., Mostyaev A. Java or alternatives? The experience of transferring applications to Android platform. *SWorld Scientific Papers*. 2015, vol. 5, no. 1, pp. 15–27 (in Russ.).
13. Zakharov V., Malkovsky M., Mostyaev A. Programming language choice problems during cross-platform application development. *INJOIT*. 2017, vol. 5, no. 7, pp. 29–37 (in Russ.).

УДК 004.424.2  
DOI: 10.15827/0236-235X.126.238-243

Дата подачи статьи: 07.02.19  
2019. Т. 32. № 2. С. 238–243

## Социальные особенности разработки мобильных приложений

*А.И. Мостяев*<sup>1</sup>, аспирант, *reistlin12@gmail.com*

<sup>1</sup> *Московский государственный университет им. М.В. Ломоносова, факультет вычислительной математики и кибернетики, г. Москва, 119991, Россия*

Современные технологии разработки мобильных приложений развиваются бурными темпами, стремясь удовлетворить все новые и новые запросы пользователей. Разработчики стараются не отставать и поддерживать популярность своих приложений всеми возможными способами, внедряя самые современные возможности и функции.

В данной работе описаны основные особенности мобильных приложений и методов их поддержки по сравнению с аналогами для персональных компьютеров. Учет этих особенностей при разработке и поддержке мобильных приложений должен устранить разрыв и недопонимание между пользователями мобильных приложений и их разработчиками. Это, несомненно, будет полезно для обеих сторон.

Краткий обзор истории мобильных приложений, сделанный в начале статьи, сразу дает представление о скорости развития данной области. Затем подробно описываются основные особенности мобильных приложений. Подчеркивается специфика как технической реализации, так и удобства использования приложений. Среди особенностей выделены тесная интеграция с операционной системой, короткие сеансы работы, повсеместное использование интернет-сервисов и разнообразие поддерживаемых устройств. Уделено внимание особенностям жизненного цикла мобильных приложений и интеграции сторонних библиотек для работы с интернет-сервисами. Кроме того, рассматриваются качество локализации приложений, специфика локализации для отдельных стран и работа с текстовыми и визуальными материалами в магазинах приложений.

В работе приводится список необходимых требований к современному успешному приложению. Также отмечается интересный факт: особенности разработки подобных приложений прямо отражают тенденции развития современного общества.

**Ключевые слова:** мобильные приложения, кроссплатформенные приложения, архитектура приложений, разработка приложений, поддержка приложений.

### *Литература*

1. Number of Monthly Active Facebook Users Worldwide as of 4th Quarter 2018 (in millions). URL: <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/> (дата обращения: 04.02.2019).
2. Android's popularity eclipses Windows among internet users. The Verge. URL: <https://www.theverge.com/2017/4/3/15159320/android-windows-internet-usage-statistics-competition> (дата обращения: 04.02.2019).
3. Developer Insights. SlashData – The Leading Analyst of the Developer Economy. URL: <https://www.slashdata.co/reports> (дата обращения: 04.02.2019).
4. Cho Y.C., Jeon J.W. Current software platforms on mobile phoneю ICCAS Conf. 2007, pp. 1862–1867. DOI: 10.1109/ICCAS.2007.4406649.
5. Lin F., Ye W. Operating system battle in the ecosystem of smartphone industry. Symp. on Information Engineering and Electronic Commerce. 2009, pp. 617–621. DOI: 10.1109/IEEC.2009.136.
6. Hammershoj A., Sapuppo A., Tadayoni E. Challenges for mobile application development. Proc. 2010 14th Intern. Conf. on Intelligence in Next Generation Networks (ICIN). 2010, pp. 1–8.
7. Kamthan P. Towards high-quality mobile applications by a systematic integration of patterns. J. of Mobile Multimedia. 2008, vol. 4, no. 3, pp. 165–184.
8. Franke D., Kowalewski S., Weise C. A mobile software quality model. Proc. 2012 12th Intern. Conf. on Quality Software. 2012, pp. 154–157. DOI: 10.1109/QSIC.2012.49.
9. Wasserman A.I. Software engineering issues for mobile application development. Proc. FSE/SDP Workshop on Future of Software Engineering Research. 2010, Santa Fe, New Mexico, USA, pp. 397–400. DOI: 10.1145/1882362.1882443.
10. Ali N., Solís C., Ramos I. Comparing architecture description languages for mobile software systems. Proc. 1st Intern. Workshop on Software Architectures and Mobility. 2008, Leipzig, Germany, pp. 33–38. DOI: 10.1145/1370888.137089.
11. Charland A., Leroux B. Mobile application development: web vs. native. ACM Conf. 2011, pp. 49–53. DOI: 10.1145/1941487.1941504.
12. Захаров В.Б., Мальковский М.Г., Мостяев А.И. Java или альтернативы? Опыт переноса приложений на платформу Android // Сб. науч. тр. SWorld. 2015. Т. 5. № 1. С. 15–27.
13. Захаров В.Б., Мальковский М.Г., Мостяев А.И. Проблемы выбора языков программирования при разработке кроссплатформенных приложений // INJOIT. 2017. Т. 5. № 7. С. 29–37.