

MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF  
KAZAKHSTAN

Non-commercial joint-stock company  
«ALMATY UNIVERSITY OF POWER ENGINEERING AND  
TELECOMMUNICATIONS»  
Department of IT-engineering

**APPROVED by**  
Head of the department  
PhD, associate professor  
\_\_\_\_\_ Kartbayev T.S.  
« \_\_\_\_ » \_\_\_\_\_ 2018 y.

**DIPLOMA PROJECT**

Theme: Development of “The Qazaq Circus” a mobile application on the Android platform

Specialty 5B070400 – «Computing systems and software»

Done by: Utegenov E.U.                      Group: BT-14-3  
Scientific supervisor: c.p-m.s, associate professor Kalizhanova A.U.

Consultants:  
on an economic part: c.e.s., prof. \_\_\_\_\_ Arenbaeva Zh.G.  
« 07 » 06 2018 y.

on a health and safety: PhD., associate prof. \_\_\_\_\_ Abikenova A.A.  
« 31 » 05 2018 y.

on application of  
computer facilities: senior lecturer \_\_\_\_\_ Ramazanova A.M.  
« 23 » 05 2018 y.

Compliance supervisor: PhD., senior lecturer \_\_\_\_\_ Bidakhmet Zh.  
« 4 » 06 2018 y.

Reviewer: PhD, Deputy Director of IICT. \_\_\_\_\_ Magzom M.M.  
« \_\_\_\_ » \_\_\_\_\_ 2018 y.

Almaty 2018

MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF  
KAZAKHSTAN

Non-commercial joint-stock company  
«ALMATY UNIVERSITY OF POWER ENGINEERING AND  
COMMUNICATIONS»

Institute of Control Systems and Information Technologies

Department of IT-engineering

Specialty 5B070400 – «Computing systems and software»

**ASSIGNMENT**  
for the diploma project

Student Utegenov Erkebulan Umirbekovich

Theme of diploma project: Development of “The Qazaq Circus” a mobile application on the Android platform

Approved by the order on the university № 155 from «23» October 2017

Completion date of the finished work «01» June 2018 y.

Initial data to the work (required parameters of research results (design) and initial data of the object): Technical documentation for Java, operating system Windows 10, Android Studio development environment.

List of questions to be developed in the diploma project, or the summary of the project:

- a) analytical part;
- b) an overview of the software tools used;
- c) design and development of the application;
- d) economical part;
- e) health and safety

List of graphic material (with an exact indication of mandatory drawings): 8 tables, 25 illustrations are presented.

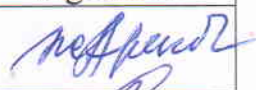
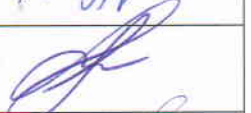

Main recommended references:

- 1 Herbert Schildt Java8. Complete guide, 9<sup>th</sup> pub.: «T.C. Williams», 2015. – 1376 p.
- 2 Prokhorenok N.A. Java. – St. Petersburg, 2018. – 320 p.

3 Reto M., Android 2. Programming applications for tablet computers and smartphones. – M. Eksmo, 2011.

4 Blokh D. Java. Effective programming, M. Lori, 2014. – 310 p

Consulting of related sections

| Section               | Consultant      | Timing                    | Signature   |
|-----------------------|-----------------|---------------------------|---|
| Economical part       | Arenbaeva Zh.G. | 01.03.2018-<br>31.05.2018 |  |
| Health and safety     | Abikenova A.A.  | 15.05.2018-<br>31.05.2018 |  |
| Computer facilities   | Ramazanova A.M. | 05.05.2018-<br>23.05.2018 |  |
| Compliance supervisor | Bidakhmet Zh.   | 05.05.2018-<br>04.06.2018 | <i>m. Duganov</i>   |

Diploma project preparation schedule

| Section name and list of developed issues                 | Deadlines for presentation to the supervisor | Annotation  |
|---|--|-------------|
| Theoretical (analytical) part of the work                 | 01.11.2017-<br>02.12.2017                    | <i>done</i> |
| An overview of the software tools used in the project     | 02.12.2017-<br>02.01.2018                    | <i>done</i> |
| Practical part, design and development of the application | 02.01.2018-<br>20.03.2018                    | <i>done</i> |

Date of issue of the assignment «25» October 2017.

Head of the department  Kartbayev T.S.

Scientific supervisor  Kalizhanova A.U.

Accepted by the student  Utegenov E.U.

## **Аңдатпа**

Бұл дипломдық жобада Қазақ Мемлекеттік Циркінің тапсырысы бойынша Android операциялық жүйесі үшін қолданбалы қосымшасы әзірнді. Бұл қосымша Қазақ Мемлекеттік Циркінің афишасын, жаңалықтарын және де басқа да өтіп жатқан оқиғаларды көруге мүмкіндік береді. Google Play мобильдік қосымшалар дүкенінде жариялануы мүмкін.

Сонымен қатар, қарастырылатын қосымшаны әзірлеу үшін еңбек жағдайларына талдау жасалды.

Сондай-ақ, жобаның экономикалық тиімділігін бекітетін экономикалық негіздемесі құрастырылды.

### **Аннотация**

В данном дипломном проекте рассмотрена разработка прикладного приложения для операционной системы Android, созданная по заказу Казахского Государственного Цирка. Данное приложение позволяет смотреть афишу, новости, и другие события, проходящие в Казахском Государственном Цирке и может быть опубликовано в магазине мобильных приложений Google Play.

Кроме того, сделан анализ условий труда для разработки рассматриваемого приложения.

Также составлено экономическое обоснование проекта, подтверждающее его экономическую целесообразность.

### **Annotation**

In this diploma project an application for Android OS, which was developed by the order of The Kazakh State Circus, has been examined. This application allows to see affiche, news and other events ongoing in The Kazakh State Circus and also may be published in the mobile application store Google Play.

In addition, analysis of working conditions for development of the application has been made.

Also economic justification of the project confirming its economic viability has been composed.

## Contents

|   |  |
|---|--|
| Introduction  | 8                                      |
| 1 Principles of development   | 10                                     |
| 1.1.1 Built-in Android applications   | 11                                     |
| 1.1.2 Main characteristics of the development environment for Android platform  | 12                                     |
| 1.1.3 Developer's framework   | 12                                     |
| 1.1.4 Android software stack  | 13                                     |
| 1.1.5 The Dalvik virtual machine  | 14                                     |
| 1.2 What Android apps consist of  | 15                                     |
| 1.2.1 AndroidManifest.xml   | 16                                     |
| 1.2.2 Creating Simple Values  | 21                                     |
| 1.2.3 Resources   | 22                                     |
| 1.2.4 Android user interfaces   | 24                                     |
| 2 Development tools   | 28                                     |
| 2.1 Graphic Shell Development Tools   | 28                                     |
| 2.2 Application code development tools  | 33                                     |
| 3 Stages of application development   | 42                                     |
| 3.2 Application development   | 49                                     |
| 3.2.1 Creating and implementing Activities  | 49                                     |
| 3.2.2 Creating and implementing Classes   | 50                                     |
| 4 Feasibility study   | 53                                     |
| 4.1 Calculation of costs for development of information technology  | 53                                     |
| 4.2 Calculating the price of a software product   | 62                                     |
| 4.3 Conclusion  | 62                                     |
| 5 Safety of life  | 64                                     |
| 5.1 Analysis of potentially hazardous and harmful factors affecting service personnel in the operation of technical equipment | 64                                     |
| 5.2 The room where the project is developed   | 65                                     |
| 5.3 Characteristics of the used equipment   | 65                                     |
| 5.4 Ventilation of the room   | 66                                     |
| 5.5 Conclusion  | 68                                     |
| Conclusion  | 70                                     |
| List of literature  | 71                                     |
| Appendix A  | <b>Ошибка! Закладка не определена.</b> |
| Appendix B  | <b>Ошибка! Закладка не определена.</b> |

## Introduction

Nowadays, we, as developers of applications for mobile platforms, have wide opportunities to go. A few years ago, it seemed not to be true that a typical developer could sell their applications to millions of users around the world using application stores, and users would buy these applications. According to various forecasts, the estimated global mobile application revenue was \$35 billion USD in 2014, \$45 billion USD in 2015, \$88 billion USD in 2016, and a predicted \$189 billion USD in 2020. The statistics indicate that people who fall in the age group of 18 to 24 years of age generally use more mobile apps than any other group. One of the reasons is constantly increasing competition between the major manufacturers of operating systems for smartphones, such as Microsoft with the Windows Phone platform, Apple with iOS and Google with Android. This suggests that in the near future these platforms will continue to develop at a rapid pace, which means that the need for mobile application developers will only grow up.

What is the uniqueness of the Android platform? The main idea of Google is that the company offers open access to the source codes of its operating system, provides a set of convenient development tools and a well-documented SDK, which should eventually lead to the emergence of a large number of software for this platform. In several years, Android has become the most successful project for mobile phones. Android occupies the mobile phone market, gradually ousting the recognized leaders from it. The Android system is now installed not only on smartphones, this platform has been adapted for tablets and netbooks.

A big step in the development of Google Android was the opening of an online application store – Android Market in October 2008, where you can buy applications and other software for devices, which are based on the new platform. In addition, software developers have the opportunity to charge money for their applications in Android Market, which makes the development of applications for this platform even more attractive.

Previously, when the android system was not perfect there were system crash problems, poor optimization and so on. It was too hard to programm, but with the release of android V.2, we can say that the world has changed and almost every release of a new OS on Android has proved that Google tries to do a better overall organization of the OS.

At the moment, we see the biggest advantrages of the system as Eclipse and Android Studio, which are gathering strength and gaining more supporters of the android system specifically programmers, because the system is free and there is Google Corporation behind it. There is also another big corporation – Microsoft, which has a lot of money. We must admit that principle languages are very similar to each other, both are cross-platform, but our preference was given to Java than C #, as there is a device for android to test the application. People choose Android phones more often rather than Windows Phone, so iOS is one of the rivals of Android. According to recent data, even iPhone is not so popular as phones on Android. This system is now everywhere in communicators, cell phones, tablets, TVs

and you can endlessly list these types of equipment, but the fact is future is behind Google.

The purpose of this diploma project was to create a mobile application for the Android operating system, which is aimed to help people who are in love with circus and art in general to see upcoming events and always be aware of the latest news.

# 1 Principles of development

## 1.1 Android operating system

Android - one of the new generation of operating systems, designed to work with the hardware of modern mobile devices. Now Windows Mobile, Apple iPhone and Palm Pre offer quite powerful and easy to use mobile application development environments. However, unlike Android, these are proprietary operating systems, in which, in certain cases, priority is given to embedded software, not to third-party applications. In addition, these operating systems limit the ability of applications to interact with data on the phone, and also limit or control the distribution of third-party applications created for these platforms.

Android gives new opportunities for mobile applications, offering an open development environment built on the open Linux core called as kernel. All applications have access to the hardware of the device, which uses a special series of API-libraries. In addition, full and controlled application interaction support is included.

On Android platform, all programs have the same status. Third-party applications are written on the same API as the embedded software, and also all programs have the same execution time. Users can remove or replace embedded software on alternative third-party development, whether it's a dialer or a desktop.

Simpler Android can be represented as a combination of three components:

- a) free open source operating system;
- b) open source development environments for creating mobile applications;
- c) devices, mostly mobile phones, on which Android operating system is installed along with the applications developed for it.

Android includes several necessary and interdependent elements:

- a) reference design of hardware with a list of requirements for mobile devices to ensure compatibility with the software;
- b) the core of the Linux operating system, which provides a low-level interface for managing hardware, memory and processes optimized for mobile devices;
- c) open source libraries designed for SQLite application development, WebKit, OpenGL and media manager;
- d) the execution environment for applications, including the Dalvik virtual machine and kernel libraries that are responsible for the functionality of Android; The runtime environment is small in size, which makes it possible to effectively use it on mobile devices;
- e) a set of software components that provide access to system services at the application level; among them a window manager and a location manager, content providers, telephony and touch screen capabilities;
- f) a set of user interface components for placing and launching applications;
- g) pre-installed applications supplied in the common software suite;
- h) A set of programs for development of applications, including tools, plug-ins and help documentation.



Particularly worth emphasizing is that the open architecture of Android allows you to fix any errors in the user interface or the design of embedded applications by writing extensions or replacing errors. Android provides the ability to create custom interfaces for mobile phones, as well as applications with functionality and design that best meet your needs.

### **1.1.1 Built-in Android applications**

Phones with the Android system are provided with a set of preinstalled programs developed within the framework of the Android Open Source Project (AOSP) (Open Source Project for Android).

Let's list the main of them:

- a) e-mail-client;
- b) application for working with SMS;
- c) a complete set of tools for personal data management, including calendar and address book;
- d) WebKit-based browser;
- e) music player and photo gallery;
- f) calculator;
- g) "Desktop";
- h) alarm clock.

In many cases, Android also includes the following licensed software from Google:

- a) Android Market application for downloading third-party programs developed for the Android platform;
- b) a full-fledged Google Maps application that includes Street View, Driving Directions, routed navigation, a satellite map and traffic information;
- c) A program for working with Gmail;
- d) a program for instant messaging Google Talk;
- e) A video player for working with the YouTube service.

Data that many of these applications access, such as the address book, are open to third-party programs. In addition, applications can handle events such as an incoming call or receiving SMS.

The appearance of the programs that are installed on new Android-powered phones can vary greatly depending on the hardware manufacturer and / or the operator, the distributor.

The open nature of the Android platform means that operators or OEMs can change the user interface and set of programs on any device running Android. Some manufacturers have developed their own Android-based interfaces, for example, Sense from HTC, MotoBlur from Motorola and a user interface from Sony Ericsson.

It is important to note that for all compatible devices, the platform and development environment remain unchanged regardless of the manufacturer or operator. The user interface can change, but the programs will work exactly the same on all compatible Android devices.

### **1.1.2 Main characteristics of the development environment for Android platform**

The main treasure of Android as the development environment was its API.

Android as an application-neutral platform provides the ability to create programs that will become as much an integral part of the phone as the components that come with it.

The following list illustrates the main characteristics of Android:

- a) Absence of expenses for use of the license, distribution and development, and also any mechanisms of certification of ready software products;
- b) access to the Wi-Fi-device;
- c) in GSM, EDGE and 3G networks intended for telephony and data transmission, you can call or receive calls and SMS, send and receive data;
- d) Comprehensive API for working with navigation services, such as GPS;
- e) Full control over multimedia devices, including playing or recording information from the camera and microphone;
- f) API for working with sensor devices, such as an accelerometer and a compass;
- g) libraries for working with Bluetooth with the ability to transfer data via the p2p protocol;
- h) transmission of IPC messages;
- i) storage for shared data;
- j) background applications and processes;
- k) widgets for the Desktop, Live Folders and Live Wallpaper;
- l) the ability to integrate the results of an application search into a system search;
- m) Built-in browser based on WebKit with open source and HTML5 support;
- n) Full support for applications that use the functionality of working with cards in their user interface;
- o) optimized for mobile devices graphics system with hardware acceleration, including a library for working with vector 2D graphics and support for 3D graphics using OpenGL ES 2.0;
- p) multimedia libraries for playing and recording audio, video files or images;
- q) localization with tools for working with dynamic resources;
- r) A set of software components for reusing components and replacing embedded applications.

### **1.1.3 Developer's framework**

The programming language for applications for the Android platform is Java. However, they are not run on the classic Java VM, but on a special Dalvik virtual machine.

Each Android application functions in a separate process inside its own copy of the Dalvik machine. All responsibility for memory and process management is assigned to Android, which stops or kills processes if you need to free resources.

Dalvik and Android are at the top of the Linux kernel, which deals with low-level interaction with hardware, including driver and memory management. In this case, a set of built-in API allows you to access all services, functionality and hardware stuffing.

### 1.1.4 Android software stack

The Android software stack consists of the elements shown in Figure 1.1. Their detailed description is given below. Simpler, they can be represented as a combination of the Linux kernel and the set of C / C ++ libraries that are available in the application framework. The latter provides management and operation of the working environment and applications.

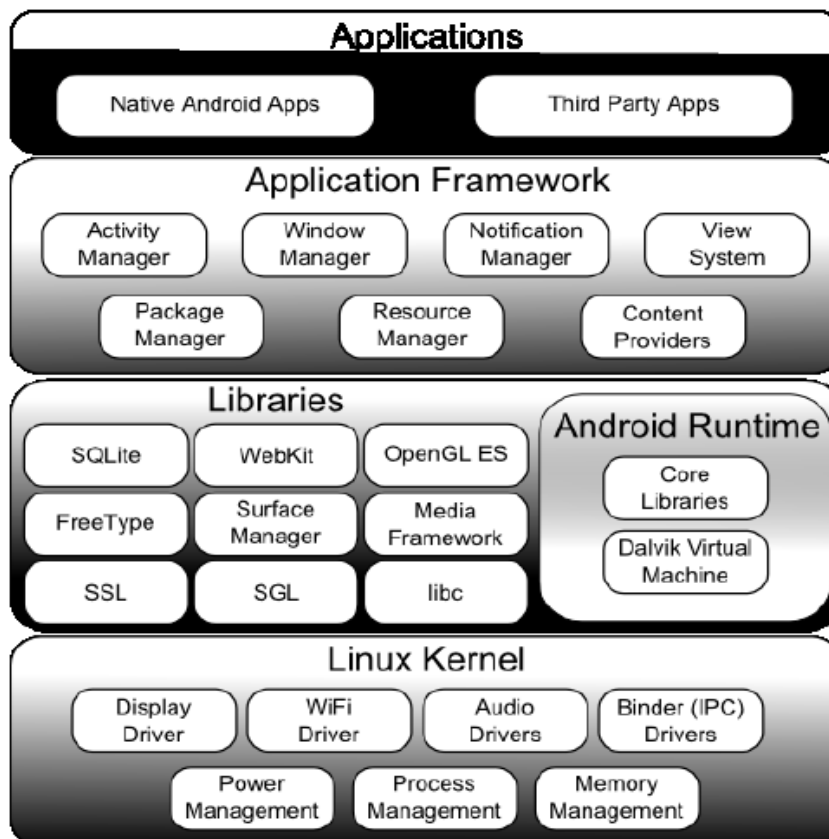


Figure 1.1 – Android software stack

*The Linux kernel.* The operation of system services (device drivers, process and memory management, power, security, network services) is provided by the Linux kernel version 2.6. It is also responsible for the level of abstraction between the hardware and the rest of the software stack.

*Libraries.* Android includes a variety of C / C ++ system libraries (for example, SSL and libc) that run on top of the kernel. Among them we can distinguish:

- a) library for working with multimedia, which provides playback of audio and video files;
- b) interface manager responsible for managing the display;
- c) graphics libraries, such as SQL and OpenGL, for 2D and 3D-graphics;
- d) library SQLite, which provides the work of built-in databases;
- e) SSL and WebKit for the built-in web browser and Internet security.

*Android working environment.* A special phone on the Android platform is made not so much by the mobile version of the Linux OS as by the Android operating environment. It includes kernel libraries and Dalvik virtual machine and provides the functioning of programs, and together with libraries forms the basis of the application framework.

*Kernel libraries.* Although Android applications are developed in Java, Dalvik is not a virtual Java machine. The Android kernel libraries provide the core functionality of the Java kernel libraries, as well as the unique functionality of Android.

*The Dalvik virtual machine.* Dalvik is a register-based virtual machine that is optimized so that several applications can be launched on the device at the same time. It is based on the Linux kernel, which provides workflows and low-level memory management.

*The application framework.* The framework includes a set of classes that are used to develop applications. It also provides generic abstract classes for accessing hardware and provides management of the user interface and application resources.

*Application layer.* All programs, both built-in and third-party, are developed at the application level using the same API libraries. The application layer functions within the Android work environment, using classes and services that are open to access at this level.

### **1.1.5 The Dalvik virtual machine**

One of the key components of Android is the Dalvik virtual machine (VM). Instead of a classic virtual Java machine, such as Java ME (Java Mobile Edition), Android uses its own VM, designed to provide efficient operation of multiple applications on one device.

The Dalvik VM is based on the Linux kernel, which provides operation of such low-level functions as security, threads, process and memory management. You can also write C / C ++ applications that will run directly on the base level of the Linux OS. Although such a possibility exists, there is no need for this.

If the application is concerned with the inherent speed and performance of C / C ++, Android provides access to the native development environment (NDK). It allows you to develop C ++ libraries using libc and libm libraries, and provides native access to OpenGL.

Access to devices and system services Android is carried out through the Dalvik virtual machine, which is considered an intermediate layer. Using the VM to execute the code of the control system program, developers have at their disposal an abstraction level that allows them not to worry about the design features of a particular device.

The Dalvik VM launches executable files, the format of which is optimized for minimal memory usage. You create an executable file with a .dex extension by transforming the compiled classes written in the Java language using the tools included in the development environment. In the next chapter you will learn about creating executable files of the Dalvik format.

## 1.2 What Android apps consist of

Applications in Android consist of loosely coupled components that are assembled together using a program manifest. A manifest is a file describing all the components of the application and the ways of their interaction, as well as metadata, including the requirements for the platform and hardware configuration.

The components listed below are the bricks that make up the applications.

*Activities.* Presentation level. Each application screen is the heir to the Activity class. Activities use Views to form a graphical user interface that displays information and interacts with the user. In terms of development for desktop platforms Activity - the equivalent of the Form (Form).

*Services.* Invisible engines of your application. Service components run in the background, launching notifications, updating Data Sources and visible Activities. Used for regular operations, which should continue even when the activity of your application is not in the foreground.

*Data sources.* Information stores. These components are needed to manage databases within the same application and provide them with access from the outside. Data sources are used to exchange information between different programs. This means that you can customize your own ContentProvider objects, opening them to other applications, and use other people's sources to work with the data that external programs have opened for you. Devices running Android contain several standard Sources that provide access to useful databases, including storage of multimedia files and contact information.

*Intentions.* A system for transferring messages between applications. Using Intention, you can broadcast messages at the system level or for specific Activities or Services. This dictates the need to perform the specified actions. After that, Android itself will determine the components that need to process the incoming request.

*Broadcast receivers.* Components that accept broadcast Intentions. If you create and register a BroadcastReceiver object, your application will be able to track the broadcast of Intentions that match the specified criteria. Broadcast receivers will automatically launch the program so that it can respond to the received Intention. Due to this, this mechanism is ideal for creating applications that use the event model.

*Widgets.* Visual software components that can be added to the home screen. This special type of Broadcast Receiver allows you to create dynamic, interactive components that users can embed into their home screens. In Chapter 10, you'll learn how to create your own widgets.

*Notifications.* System of user notifications. Allows you to signal about something, not paying attention to yourself or not interrupting the work of the current Activity. The notification mechanism is best suited for Services and Broadcast receivers, when it is necessary to attract the user's attention. For example, if you receive a text message or an incoming call, the device notifies you by flashing LEDs, playing sounds, displaying icons or displaying a message.

### 1.2.1 AndroidManifest.xml

Any application created in Android contains a manifest file, `AndroidManifest.xml`, which is stored in the project's root directory. The manifest allows you to describe the structure and metadata of your application, its components and requirements.

The manifest includes the nodes (tags) for each component (Activities, Services, Data Sources and Broadcast Receivers) that your application consists of, and using Intent Filters and permissions, determines how they interact with each other and with third-party programs.

The manifest contains attributes for specifying metadata (icons and visual styles). It should be noted that additional nodes of the upper level can be used to describe security settings, unit tests (unit tests), hardware and system requirements.

The manifest contains the `<manifest>` root tag with a package attribute that references the project package. Typically, this tag also includes the `xmlns:android` attribute, supported by system nodes within the file.

The `versionCode` attribute is used to set the current version of the application as an integer. This internal value is used to compare the versions of the program. Apply the `versionName` attribute to specify the public version that is displayed for users.

A typical `<manifest>` tag is shown in a code snippet.

```
<manifest xmlns:android=http://schemas.android.com/apk/res/android
package="com.my_domain.my_app" android:versionCode="1"
android:versionName="0.9 Beta"> [ ... вложенные узлы манифеста ... ]
</manifest>
```

The `<manifest>` tag includes nodes that describe the software components, security settings, classes for testing, and the requirements that make up your application. Specify the tags available within the `<manifest>` node, as well as code snippets in XML format, showing how to use these tags.

### *Uses-sdk*

Allows you to set the minimum, maximum, and target versions of the SDK that must be available on the device so that your application can function correctly. Based on the SDK version that is supported by the installed platform, and using the combination of the `minSDKVersion`, `maxSDKVersion`, and `targetSDKVersion` attributes, you can limit the range of devices that can run the application.

The `minSDKVersion` attribute points to the minimum version of the SDK that contains the API that is used in your program. If you do not specify a minimum version, the default value will apply, and your application will not be able to work correctly if you try to access APIs that are not available on the current device.

The `maxSDKVersion` attribute allows you to determine the latest version that you are ready to support. Your application will be invisible on the Android Market for devices managed by a system with a more recent version. To set the value for this attribute is recommended only if you are absolutely sure that the application does not work on a platform with a version higher than the specified one.

`targetSDKVersion` allows you to specify the platform for which you developed and tested the application. By setting the value for this attribute, you tell the system that to support this particular version, no changes are required that are related to direct or backward compatibility.

```
<uses-sdk android:minSdkVersion="4"          android:targetSdkVersion="5">
</uses-sdk>
```

### *Uses-configuration*

Use the usage-configuration tags to specify all the input mechanisms supported by your application. You can specify any combination that contains the following devices:

a) `reqFiveWayNav` - set this attribute to true if you need an input device that supports navigation up, down, left, right, and also pressing the selected item; this category includes both trackballs, and manipulators D-pad;

b) `reqHardKeyboard` - if your application needs hardware, keyboard, set to true;

c) `reqKeyboardType` - allows you to specify the type of keyboard - nokeys, qwerty, twelvekey or undefined;

d) `reqNavigation` - if you need a device for navigation, specify one of the following values - nonav, dpad, trackball, wheel or undefined;

e) `reqTouchScreen` - If your application needs a touch screen, select one of the following values: notouch, stylus, finger, or undefined.

f) You can specify several supported configurations, such as a device with a capacitive touch screen, trackball and hardware keyboard (either qwerty or twelvekey), as shown below.

```
<uses-configuration          android:reqTouchScreen=["finger"]
android:reqNavigation=["trackball"]>
```

```

android:reqHardKeyboard=["true"]
android:reqKeyboardType=["qwerty"/>           <uses-configuration
android:reqTouchScreen=["finger"]
android:reqNavigation=["trackball"]
android:reqHardKeyboard=["true"]
android:reqKeyboardType=["twelvekey"/>

```

### *Uses-feature*

One of the advantages of Android is the wide range of hardware platforms on which it can work. Use simple tags `usesfeature` to specify all the hardware capabilities required by the application. This will prevent your program from being installed on devices that do not meet the hardware requirements. You can request support for any equipment that is not required for compatible devices. To date, the hardware options offer the following options:

a) `android.hardware.camera` (if you need the application to work hardware camera);

b) `android.hardware.camera.autofocus` (if you want a camera with auto focus).

You can also use the `use-feature` tag to specify the minimum version of OpenGL that is required to run your application. Use the `glEsVersion` attribute to specify the version of OpenGL ES as an integer. The first 16 bits correspond to the major version, and the last - to the minor version.

```

<uses-feature           android:glEsVersion="           0x00010001"
android:name="android.hardware.camera" />

```

### *Supports-screens*

After the first wave of devices with HVGA screens in 2009, the list of devices running Android added models with support for WVGA and QVGA. Since future devices are likely to be equipped with large displays, with the `help-screen` tag you can specify the screen sizes that are supported (and not supported) by your application.

The exact figures will vary depending on the hardware, but overall the size and resolution of the screens is determined as follows:

a) `smallScreens` - screens with a resolution lower than the usual HVGA, usually it's about QVGA;

b) `normalScreens` - used to describe screens of standard mobile phones, at least HVGA, including HVGA and WQVGA;

c) `largeScreens` - large screens, much larger than a mobile phone;

d) `anyDensity` - set to true if your application is scalable to be displayed on the screen at any resolution.

In SDK 1.6 (API level 4), the default values for each attribute are true. Use this tag to specify the sizes of screens that you do not support.



```

<supports-screens
android:normalScreens=["true"]
android:anyDensity=["false"] />

```

```

android:smallScreens=["false"]
android:largeScreens=["true"]

```

### *Application*

Only one instance of this tag can be present in the manifest. It uses attributes that contain metadata for your application (including its name, icon, and visual style). At design time, you must set the debuggable attribute to true to enable debugging mode, although for the final versions, it's likely that you need to disable it.

The <application> tag also acts as a container, which includes nodes for Activities, Services, Data Sources, and Broadcast receivers that describe the components of the application. In addition, you can define your own implementation of the Application class. Later in this chapter, you will learn how to inherit this class and use it to control the state of the application.

```

<application
android:theme="@style/my_theme"
android:debuggable="true">
</application>

```

```

android:icon="@drawable/icon"
android:name="MyApplication"
[ ... вложенные теги ... ]

```

### *Activity*

The <activity> tag is required for each Activity that the application displays. Use the android: name attribute to specify the name of the Activity class.

Using these tags, add the main Activity that will be launched first, as well as the other screens and dialog boxes that can be displayed. An attempt to launch Activity without the corresponding description in the manifest will result in an exception being thrown. Each <activity> tag supports nested <intent-filter> nodes, indicating which Intention can start the Activity.

```

<activity
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>

```

```

android:name=".MyActivity"

```

### *Service*

As in the previous case, each Service class must have a service tag. The service tags support nested <intent-filter> nodes, through which latent binding occurs.

```

<service
android:name=".MyService"></service>

```

```

android:enabled="true"

```

### *Provider*

With this tag, you specify all the Data sources in the application. The data sources described in Chapter 7 are used to control access to databases and to exchange information within one or more programs.

```
<provider android:permission="com.paad.MY_PERMISSION"
android:name=".MyContentProvider" android:enabled="true"
android:authorities="com.paad.myapp.MyContentProvider">
</provider>
```

### *Receiver*

By adding a receiver tag to the manifest, you can register the Broadcast receiver without starting the application. Broadcast receivers track events at the global level: after registering, they will trigger when the system broadcasts or the application of the corresponding Intent. By registering them in the manifest, you can make this process completely anonymous. When broadcasting the corresponding Intent, your application starts automatically, launching the registered Receiver.

```
<receiver android:enabled="true" android:label="My Intent Receiver"
android:name=".MyIntentReceiver">
</receiver>
```

### *Uses-permission*

The uses-permission tags as part of the security system describe the credentials that, in your opinion, are needed by the application for full-fledged work. Added permissions are granted to the user before installation. To use many standard services in Android, you need authorization (in particular, for activities related to paid services and security, such as phone calls, receiving SMS or using geolocation services).

```
<uses-permission
android:name="android.permission.ACCESS_LOCATION"/>
```

### *Permission*

Third-party applications can also specify credentials before providing access to common software components. To restrict access to the application component, you must describe the appropriate credentials in the manifest. To do this, you must use the permission tag. The components of the current application can require credentials using the android: permission attributes. Other programs must use use-permission tags in their manifest to use these protected components.

Inside the permissions tag, you can specify the access level that is provided by this authority (normal, dangerous, signature, signatureOrSystem), a label and an external resource that contains a description and an explanation of the risks that accompany this authorization.

```

    <permission                android:name="com.paad.DETONATE_DEVICE"
android:protectionLevel="dangerous"                android:label="Self Destruct"
                android:description="@string/detonate_description">
    </permission>

```

### *Instrumentation*

Classes derived from Instrumentation provide a framework for testing software components during their execution. They contain methods-interceptors, through which the work of the program and its interaction with system resources are monitored.

```

    <instrumentation                android:label="My                Test"
android:name=".MyTestClass"
                android:targetPackage="com.paad.aPackage">
    </instrumentation>

```

A detailed description of the manifest and all these tags can be found at <http://developer.android.com/guide/topics/manifest/manifest-intro.html>

The New Project Wizard, which comes with ADT (New Project Wizard), automatically adds a manifest file for each new project.

## **1.2.2 Creating Simple Values**

Simple values - strings, colors, sizes and arrays (string and integer) are supported, this data is stored in XML format inside the res / values directory.

Using tags, the types of stored values are specified.

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">To Do List</string>
    <color name="app_background">#FF0000FF</color>
    <dimen name="default_border">5px</dimen>
    <array name="string_array">
        <item>Item 1</item>
        <item>Item 2</item>
        <item>Item 3</item>
    </array>
    <array name="integer_array">
        <item>3</item>
        <item>2</item>
        <item>1</item>
    </array>
</resources>

```

This example contains all available types of simple values. Each type of resource is stored in a separate file, for example, the file `res / values / strings.xml` includes only string constants.

### 1.2.3 Resources

Resources - external files (not code) that are used by your code are compiled into your application and built into it during operation. Android supports many different types of resource files, including XML, PNG and JPEG. XML files have very different formats depending on what they describe. Resources are described in the source code, and XML files are compiled into binary code for fast and efficient download. The lines are compressed into a more form, more memory-saving.

#### *List of resources*

Types of resources and their location:

- a) Layout files - `"/ res / layout /"`.
- b) Images - `"/ res / drawable /"`.
- c) Animation - `"/ res / anim /"`.
- d) Styles, strings, and arrays - `"/ res / values /"`.

The names can not be different:

- a) 'arrays.xml' for defining arrays.
- b) 'colors.xml' for defining colors.
- c) # RGB, # ARGB, #RRGGBB, #AARRGGBB.
- d) 'dimens.xml' for dimensioning.
- e) 'strings.xml' for defining strings.
- f) 'styles.xml' for defining object styles.
- g) Raw files like mp3 or video - `"/ res / raw /"`.

#### *Using Resources in the Code*

To use a resource in the code, you only need to know the complete resource ID and to what type of object your resource has been compiled. Here is the syntax for accessing the resource.

```
R.resource_type.resource_name
```

or

```
android.R.resource_type.resource_name
```

`Resource_type` is a subclass of `R` that contains a specific type of resource. `resource_name` is the resource attribute defined in XML files, or the file name (without extension) for the resource defined by other file types. Each type of resource will be added to a subclass of `R`, depending on its type.

The resources compiled by your application can be used without a package name (just like `R.resource_type.resource_name`). Android contains many standard resources, such as screen styles and button backgrounds. You can access them in the code via `android.R.resource_type.resource_name`, for example.

```
android.R.drawable.button_background
```

### *Link to Resources*

A value in an attribute or resource can also be a reference to another resource. This is often used in layout files to store strings (so you can localize the application) and images (located in another file), although the link can be to any type of resource, including colors and numbers.

For example, if we have resources with colors, we can write a layout file that sets the color of the text to the value contained in one of the resources.

```
<EditText
    android:layout_width="fill_parent"    android:layout_height="fill_parent"
    android:textColor="@color/opaque_red" android:text="Hello, World!" />
```

Note the prefix "@", indicating that this is a link to the resource, the text after it is the resource name in the form @ [package:] type / name. In the example, we do not define the package, because we refer to the resource in our own package. To refer to the system resource, you would have to write.

```
<EditText
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:textColor="@android:color/opaque_red"    android:text="Hello,
World!" />
```

In the following example, we use a resource reference, storing the rows in the layout file so that they can be localized.

```
<EditText
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:textColor="@android:color/opaque_red"
    android:text="@string/hello_world" />
```

### *Alternative resources and localization*

Alternative resources and localization is a serious problem, quite well solved in Android. Usually, more often than not, you would have to design a UI that is well suited to each of the possible screen resolutions at the same time, which is almost impossible.

You can add different UI, languages, or device support to your application with different component configurations.

Note that even if you add many different languages, UI and all other resources, the SDK itself will determine the set of resources that will be used. For example, Android itself will guess where what language you need and choose it. Or UI. To include additional resources, create parallel folders with resources and add a parameter (specifier) to each name through the dash where this folder belongs (language, screen orientation, dots per inch, resolution, etc.). For example, this project has English and German localization.

```
MyApp/ res/ values-en/ strings.xml values-de/  
strings.xml
```

Android supports several types of specifiers, with different values for each. Add them to the end of the resource folder name, separated from the name by a dash. You can add many specifiers, separating them from each other with dashes. For example, a folder that contains drawable resources for a specific configuration only.

```
MyApp/  
  res/  
    drawable-en-rUS-port-92dpi-finger-keyshidden-12key-  
    dpad-480x320/
```

Moreover, you can define only a few specific configuration options for which a resource is defined.

```
MyApp/ res/  
  drawable-en-rUS-finger/ drawable-port/ drawable-port-160dpi/  
  drawable-qwerty/
```

Android will choose which of the different main resource files fits best at runtime, depending on the current device configuration.

*R.java*

The project's *R.java* is an automatically generated file indexing all the resources of your project. You use this class in your source text as a kind of way to access the resources that you have included in your project. This is especially important given the features of integrated development environments, because it allows you to quickly and interactively locate the specific information that you are looking for. Additionally, at compile time, you get the confidence that the resource you want to use really exists

### **1.2.4 Android user interfaces**

User interfaces (UI) in Android can be created in two ways, through XML-code or in java-code. Creating a graphical user interface structure in XML is very

preferable, because by the principle of the Viewer's Viewer Control, the UI should always be separated from the program logic. In addition, adapting the program from one screen resolution to another is much simpler. The definition of UI in XML is very similar to creating a generic HTML document where you have such a simple file.

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
The content of the body element.
</body>
</html>
```

It's like in Android XML-Layouts. Everything is well structured and can be expressed by tree structures.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"                android:layout_width="fill_parent"
android:layout_height="fill_parent">
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"      android:text="Hello      World"/>
</LinearLayout>
```

### *Hierarchy of Screen Elements*

The main functional module of the Android application is Activity - an object of the class `android.app.activity`. Activity can do many things, but separately it does not have a presence on the screen. To give your Activity a presence on the screen and design its UI, you work with Views and Viewgroups - the basic units of user interface expression on the Android platform.

### *Views*

View is an object that extends the base class `android.view.view`. This is a data structure whose properties store Layouts and content for a particular rectangular area of the screen. The View object processes the dimension, its layout, the drawing, the center changes, the scrolling, and the keys / signs for the area of the screen it represents. The View class serves as the base class for all graphic pieces - a number of fully implemented subclasses that draw interactive elements of the screen. The graphic fragments process their own dimension and pattern, so you can use them to

create your UI more quickly. The list of available graphical fragments includes TextView, EditText, Button, RadioButton, Checkbox, ScrollView, etc.

*Viewgroups*

Viewgroup is the object of the android.view.viewgroup class. Viewgroup is a special type of the View object whose function is to contain and manage the View and Viewgroup. Viewgroups allow you to add a structure to your UI and create complex screen elements that can be accessed as a single object. The Viewgroup class serves as the base class for Layouts, a series of fully implemented subclasses that provide common Layouts of the screen. Layouts give you a way to embed a structure for the View series.

*UI with tree structure*

On the Android platform, you define UI Activity using the View tree and the Viewgroup nodes, as shown in the diagram below. A tree can be as simple or complex as you do it, and you can build it using sets of predefined graphics and Layouts Android, or custom View types that you create yourself. Figure 1.2 shows a tree-like system of the user interface.

To attach a tree to the screen and calculate it, your Activity calls its method setContentView () and passes the information to the root object of the node. Once the Android system receives information on the root object of the node, it starts working directly with the node to measure, and calculate the tree. When your Activity becomes active and receives priority, the system registers your Activity and asks the root node to measure and calculate the tree. Then the root node asks that its child nodes count themselves - in turn, each Viewgroup node in the tree is responsible for miscalculating its direct child nodes. As mentioned earlier, each View group has the responsibility of measuring its available space, the location of its child nodes, and calling draw () on each child node to allow all of them to calculate themselves. Child nodes can request the size and location in the parent, but the parent object has the final solution, where and how large each child can be.

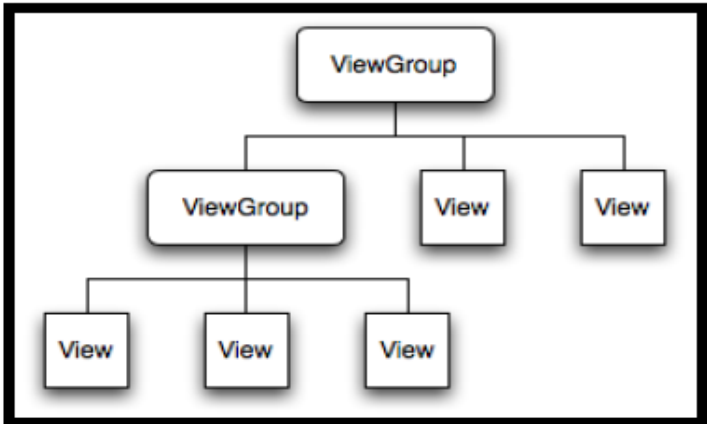


Figure 1.2 – UI Android OS - tree structure



### *Comparison of Android UI Elements with Swing UI Elements*

Since some developers who read this may have found that UIs are similar to Swing, there will now be some common features between Android and Swing:

- a) Activity в Android – почти (J) Frame в Swing.
- b) View в Android – (J) Component в Swing.
- c) TextViews в Android – (J) TextField в Swing.
- d) EditTexts в Android – (J) TextField в Swing.
- e) Button в Android – (J) Button в Swing.

Setting the listeners to the View in Android is almost the same as in Swing.

```
// Android myView.setOnClickListener(new OnClickListener(){ ...  
// Swing myButton.addActionListener(new ActionListener() {...
```

## 2 Development tools

### 2.1 Graphic Shell Development Tools

To create graphic elements of the project, software developed by Adobe for working with vector graphics is used - Illustrator CC. To purchase it, go to <http://www.adobe.com/en/products/illustrator.html> and apply for an annual subscription to the application, then download and install the program. In addition, there is an opportunity to purchase a DVD-ROM in specialized stores. The beginning of the installation process is shown in Figure 2.1.

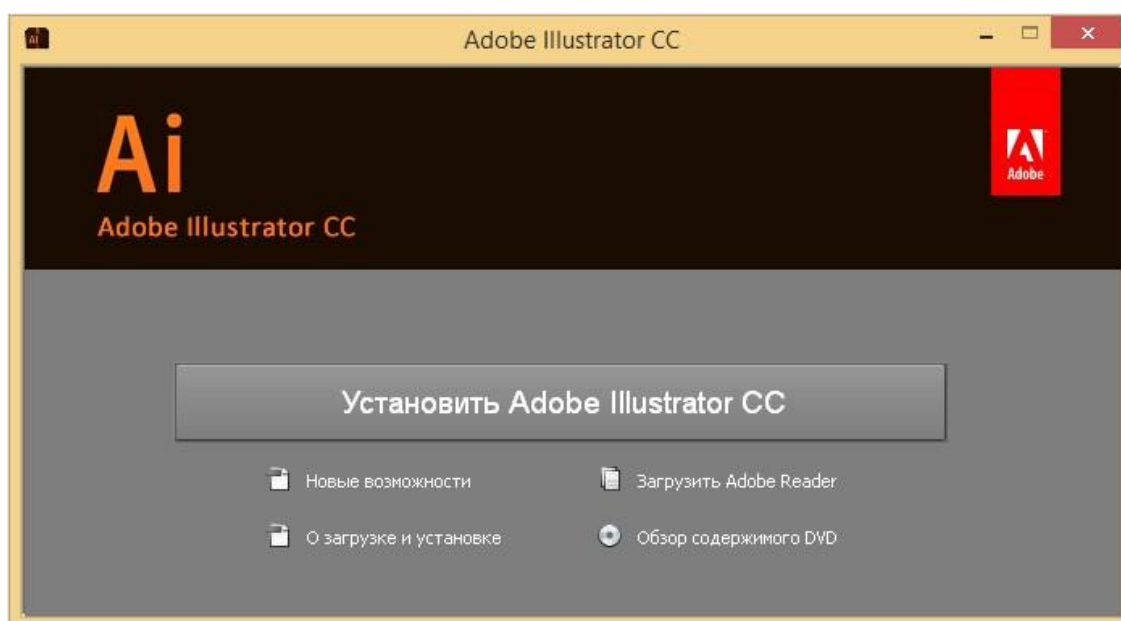


Figure 2.1 – Beginning of the installation of Adobe Illustrator CC

There are 2 versions of Adobe Illustrator CC, for 32 and 64 bit OS, respectively. If you are using a 64x system, you do not need to install both versions, only a version for your OS is enough. Installing one version of the program will save the disk space. After selecting the required version, click the "install" button. Figure 2.2 shows the choice of the installed version, followed by Figure 2.3 with the interface of the already installed application.

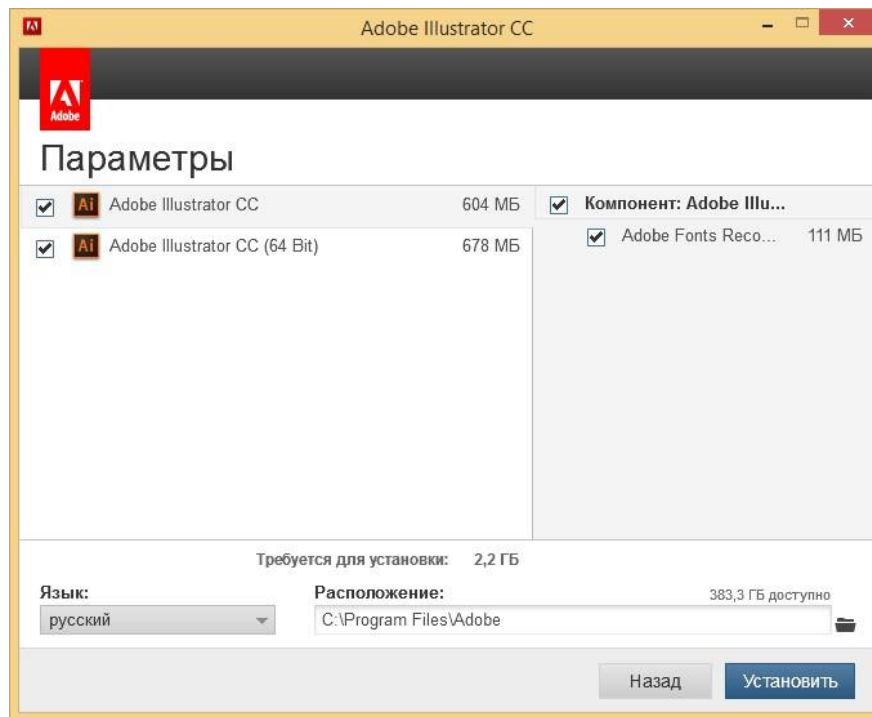


Figure 2.2 – Selecting which versions of the application are installed

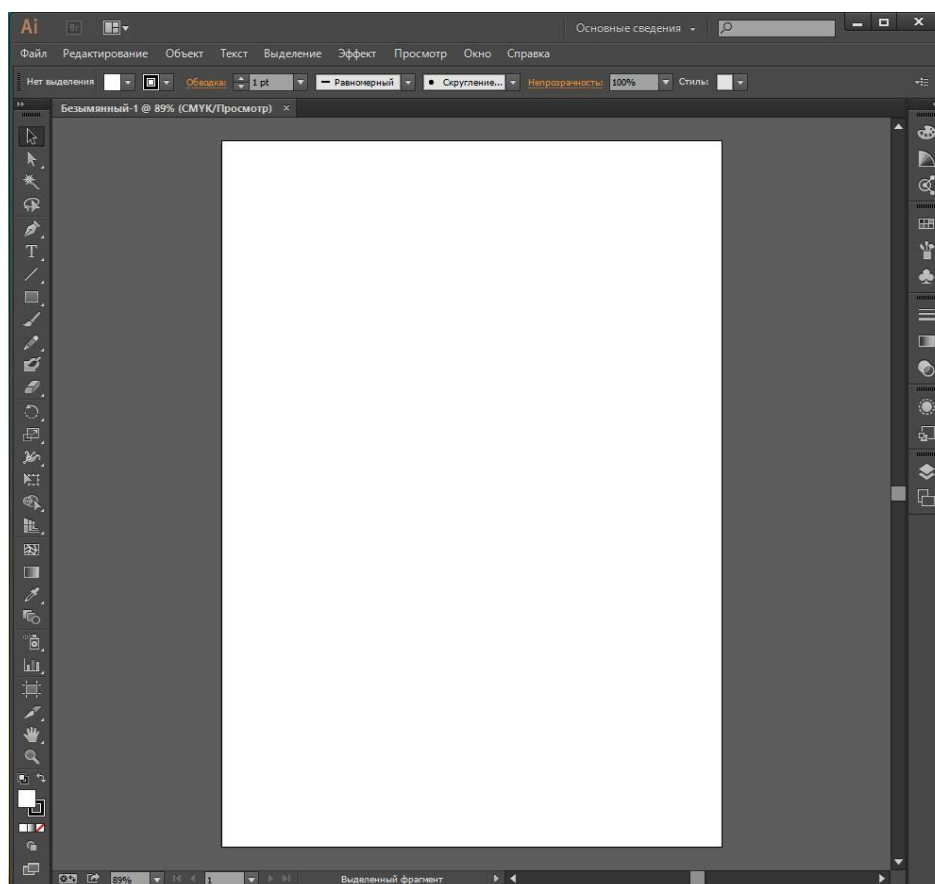


Figure 2.3 – The window of the running Illustrator CC application

Adobe Illustrator CC is the latest version of the vector image editor, which provides new opportunities for designers, and also improves existing tools.

#### *Dynamic angles*

Make your work elegant thanks to a more accurate and intuitive user interface. Experiment with rounding corners of shapes and outlines using markers or by entering parameter values in the new "Angles" dialog box or directly on the control panel. One or more corners can be rounded, rolled, or chamfered at the same time.

#### *Completely modernized tool "Pencil"*

Now with this tool you can more accurately draw curves with the ability to add straight lines, as well as the extension and closure of contours. Use predefined settings to create smooth contours with fewer dots or more accurate brush strokes with a natural brush. This new technology extends the capabilities of the Brush, Blob Brush and Smoothing tools.

#### *Changing the shape of the contour segment*

Drag the contour segments freely to create the desired shape. The new contour shape change technology, which is available for the "Reference Point" and "Direct Selection" tools in the pen tool menu, provides a more accurate and intuitive way to edit the contour segments.

#### *Integration with Typekit*

Select the desired Adobe Typekit font by opening the Typekit site directly from the Font or Headset menu. Select any of the more than 700 Typekit fonts, and then sync it with your computer. In the "Font" menu, you can quickly apply the desired filter to Typekit fonts.

#### *Support for Windows 8*

Use for drawing pen with the function of determining the degree of depression that comes with the newest tablets Windows 8, as well as improved support for the function of direct touch input. Enjoy the support of HiDPI displays on computers running Windows 7 and 8.

#### *Customizable toolbar*

Personalize your workspace and tools for specific tasks. Create specialized toolkits by dragging only the right tools onto the user panel, for example, tools for drawing, editing, or selection. After that you can hide the entire "Tools" panel, getting more space for creativity.

#### *Import and export settings*

Synchronize your Illustrator settings on multiple computers. Just export them to a folder from which other users can then import them. The synchronization function allows you to not only standardize the settings on your devices, but also share them with colleagues.

#### *New features that help save time*

Use such new time-saving features as the ability to draw artboards and resize them from the center, access the fill and stroke controls right in the Swatches panel, and the sliders to adjust the opacity. Improved drawing capabilities with perspective

Experiment with changing the perspective grid mesh attributes, such as correcting the perspective and the horizon, and viewing the dynamic changes to the graphic objects, taking into account these edits.

#### *Updating the export function to SVG format*

Export your work to a scalable and optimized SVG file format that can be adapted to screens of various sizes and resolutions, and then perform end-to-end editing of SVG files while maintaining accurate pixel alignment.

#### *Change Text tool*

Create projects and add text to them using an effective "Change Text" tool. Now you can work with symbols as with separate objects. Experiment with the fonts, move, scale and rotate the text. Now you can create works not only with the mouse or stylus, but also simply touching the touch screen of the mobile device.

#### *Images in the hands*

Draw with the brush in which the photo was placed. The object, patterned and discrete brushes can contain raster images, which allows you to create a complex design in a few minutes, drawing strokes that mimic brush strokes with a natural brush. Whichever Illustrator brush you use, the shape and appearance of the stroke can be changed to your liking.

#### *Font search*

Quickly find the perfect font. In the Symbol panel, enter a font style, for example, "bold" or "italic," the font family name, or part of the font name. Only those search results that match the parameters are displayed.

#### *Using multiple files*

Import in Illustrator several files at a time and manage the process with new functions. Now you can determine the location of files (images, graphics and text), apply scaling to them, and use a new kind of thumbnails to specify the location of the file in the project.

#### *Extracting Cascading Style Sheets (CSS)*

Writing code for such web elements as icons and patterns can be tedious. However, now it is even easier to create websites thanks to the program Illustrator, which itself creates CSS code, even for logos that include gradients. Copy and paste the code directly into your web editor.

#### *Color Sync*

Fix found and liked you color themes using the application Adobe Kuler for iPhone. Publish your color themes and appreciate thousands of other users on the Kuler website. Synchronize your favorite color themes and get instant access to them from Illustrator.

#### *Convert text from point to text in the field and vice versa*

Instantly switch between the text in the field and the text from the point. Converting a text object can now be done in just a second, which greatly simplifies the process of creating a design in text layouts. Working with the imported text became even easier thanks to the function of changing the format.

### *Autographs for diaper brushes*

Create corners in just a few steps. Create patterned brushes in just a few seconds thanks to the automatic corner creation function, which is ideal for the rest of the stroke design. It is no longer necessary to deal with tedious creation of special sharp corners.

### *Using the Free Transform tool*

Improve your skills in creating a design using the Free Transform tool. Move, scale, and rotate objects directly on the touch screen of your mobile device. Or use a mouse or other manipulator to intuitively and quickly transform objects directly on the editing area.

### *Integration with Behance*

Keep your work right from Illustrator CC to Behance to showcase your finished projects or publish the ones you work on. As the project completes, download new versions and instantly get feedback about your work from other designers from around the world.

### *Syncing settings*

Create projects on any computer: Mac or PC.

Synchronize your workspace settings in the cloud-based Creative Cloud service, including settings, styles, brushes, and Illustrator libraries, and use them wherever you are.

### *Packing files*

The file packing feature allows you to automatically collect all necessary fonts, associated graphics and a packing report in one folder. You can use the packaging of files for more convenient delivery of projects to the customer or their systematization on the computer.

### *Extracting Images*

Easily extract images that have been placed and built into the Illustrator file. Extract the files in a few seconds and start editing them. You can also extract files embedded in an image that was received from another user. Links to image files will be created automatically.

### *Using multiple artboards*

Arrange and view up to 100 artboards of different sizes, cascaded or grided. Easily add, delete and rename areas, and change the order of their location. Save, export and print the artboards individually or together.

### *Variable width strokes*

Draw strokes of variable width, easily performing an adjustment at any stage of the work. Create and save width profiles and apply them to strokes - or use variable-width styles.

### *Images*

Convert raster images to editable vectors using an efficient trace mechanism that provides an exceptional level of control over working with colors and shapes. Simple, intuitive functions ensure high line accuracy, sharp fitting and reliable results.

### *Adobe Mercury Performance System*

Handle large, complex files with high accuracy, speed and reliability. With built-in support for 64-bit Mac OS and Windows computing, the Adobe Mercury Performance system allows an application to access all of the RAM in order to easily open, save and export large files, and preview complex designs.

## **2.2 Application code development tools**

To write the PP code in this project, there will be a used environment for developing Android Studio, and Java will serve as the language. To work in Android Studio, you must have the Java SE Development Kit (JDK) installed on the system you are using. The recommended version of JDK is at least 7. You can install it by following the link.

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads1880260.html>

Java SE Development Kit is a free development toolkit and is distributed free of charge. So, after going through the above link, you need to choose the version that corresponds to your OS, Windows, Linux or Mac OS X. After downloading the executable file, you must run it and follow the instructions to install this tool.

After that, download the Android Studio installation file from the official Android Developer site. Installation files are available at.

<http://developer.android.com/intl/ru/sdk/installing/studio.html>

Here you should also select the version that suits your OS and after the download of the executable file is complete, install the program following the instructions. Here it is worthwhile to pay attention to the fact that Android NDK loading for work in Android Studio is not required, since all necessary utilities are already included in the package of the application.

After launching the application, you will be greeted with the Welcome screen shown in Figure 2.4.

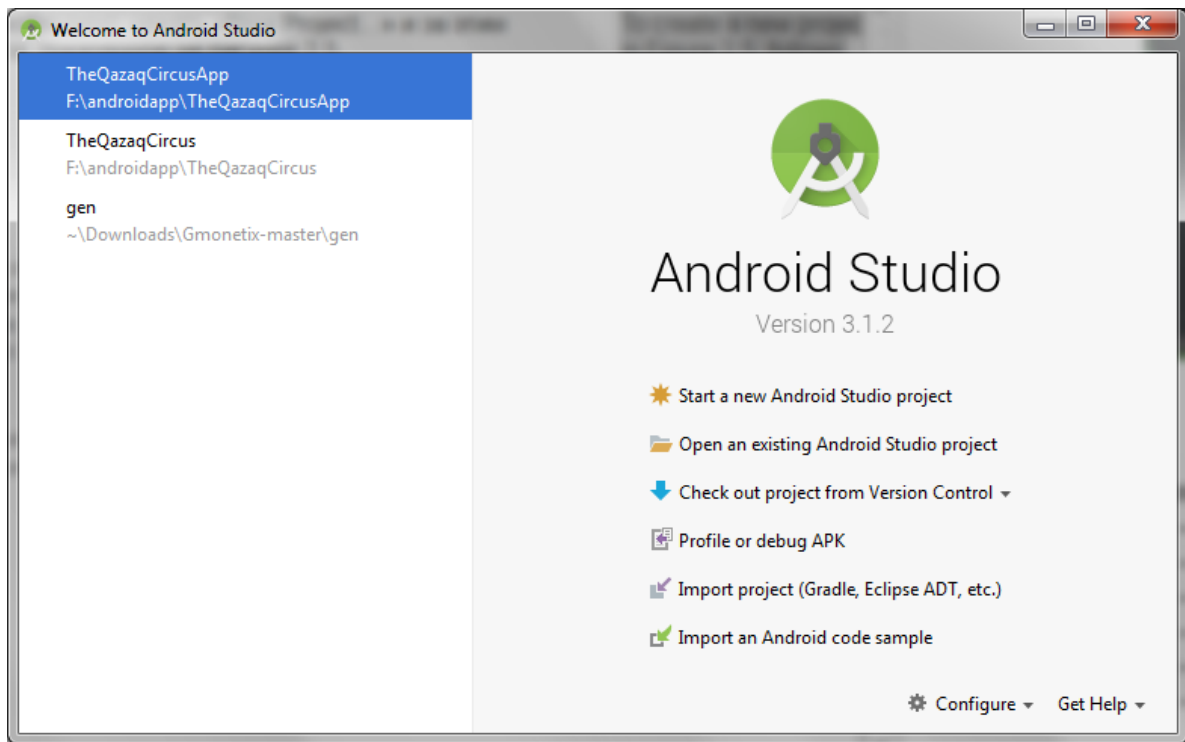


Figure 2.4 – Welcome screen of Android Studio

Here you can:

- a) Create a new project.
- b) Import an existing project.
- c) Open an existing project.
- d) Open any recent project, and perform a search among them.
- e) Check the software version is up-to-date.
- f) Edit Android Studio settings.
- g) Get access to the right.

To create a new project, click "Start a new Android..." and the project creation window, shown in Figure 2.5, follows.



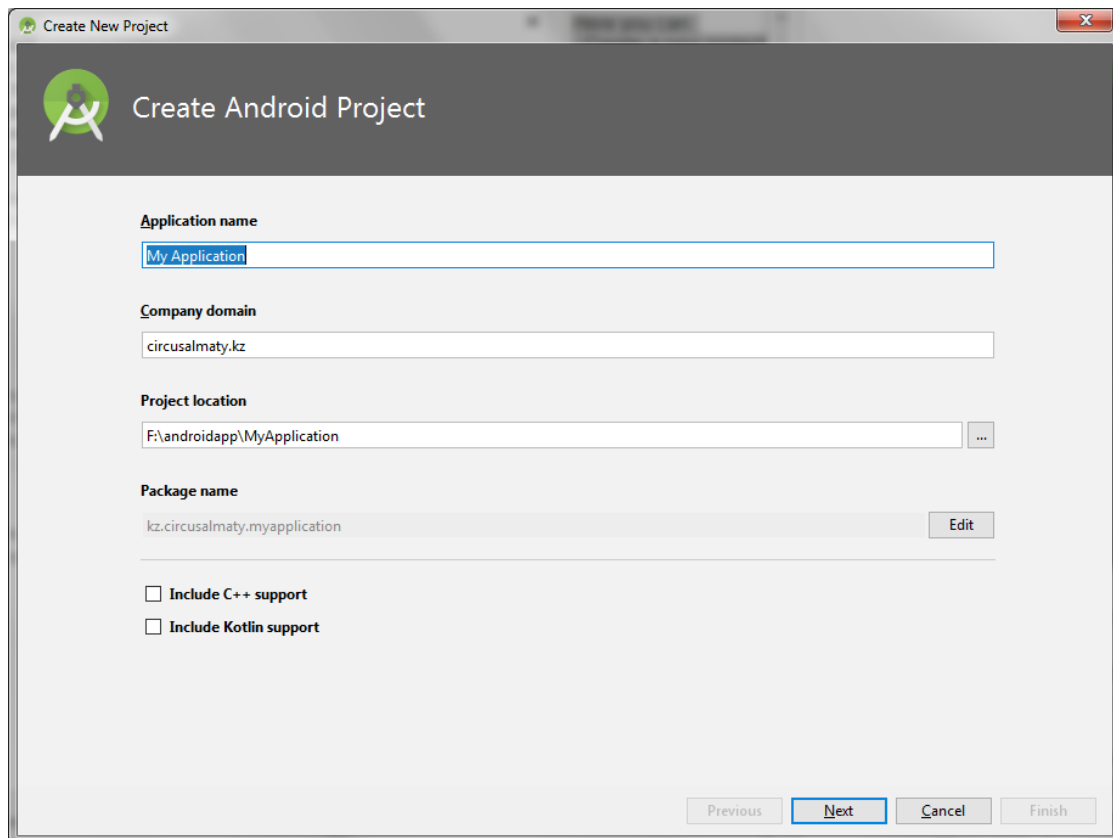


Figure 2.5 – Android Studio project creation window

On this screen you can specify the name of the application, the name of the module, the name of the files for the application, and the location of the directory with the project data. There is also a possibility to choose the minimum version of the OS supported by the application and the main one, on which it is oriented first. In addition, there is a choice of language level. Immediately available is the choice of the basic theme of the application, light and dark. If required, you can put a marker in the box of the "Create custom launcher icon" option, to create an application icon directly in Android Studio. And the conclusion is the choice of "Support Mode".

If the "Create custom launcher icon" was selected in the project window, after clicking the "next" button, you will see the window for creating and editing the application icon, which is also shown in Figure 2.6. This editor will satisfy the needs of almost all developers, because it allows you to create quickly and qualitatively the application icons for different screen densities and immediately see the result.

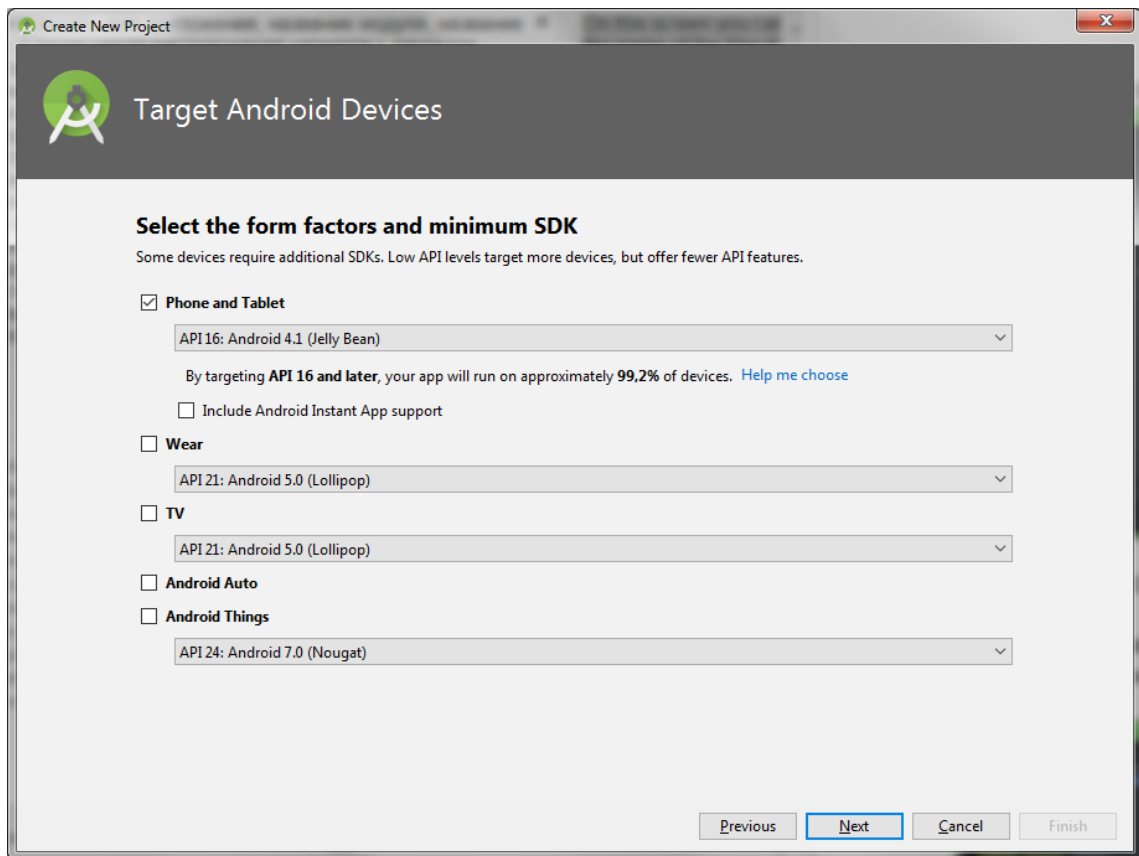


Figure 2.6 – The window for selecting the form factor and minimum SDK

For the Minimum SDK, select API 19: Android 4.4 (KitKat).

The minimum required SDK-this is the earliest version of Android that your application supports, is indicated using the API level. First we will have enough KitKat version.

In the future, when you become an experienced developer to cover the maximum number of supported devices, you should set the Minimum SDK parameter in the smallest available version that the system allows your application (the versions become available after downloading to the SDK Manager) to provide the basic set of functions. If an item from the application is available only on newer versions of Android, and this is not critical for the main functionality of the application, you can enable this feature only when running on the versions that support it (as discussed in the Supporting Different Platform Versions).

Leave all other parameters (TV, Wear, and Glass) unchecked and click Next.

After selecting the form factor and minimum SDK, you go to the basic activity selection screen shown in Figure 2.7.

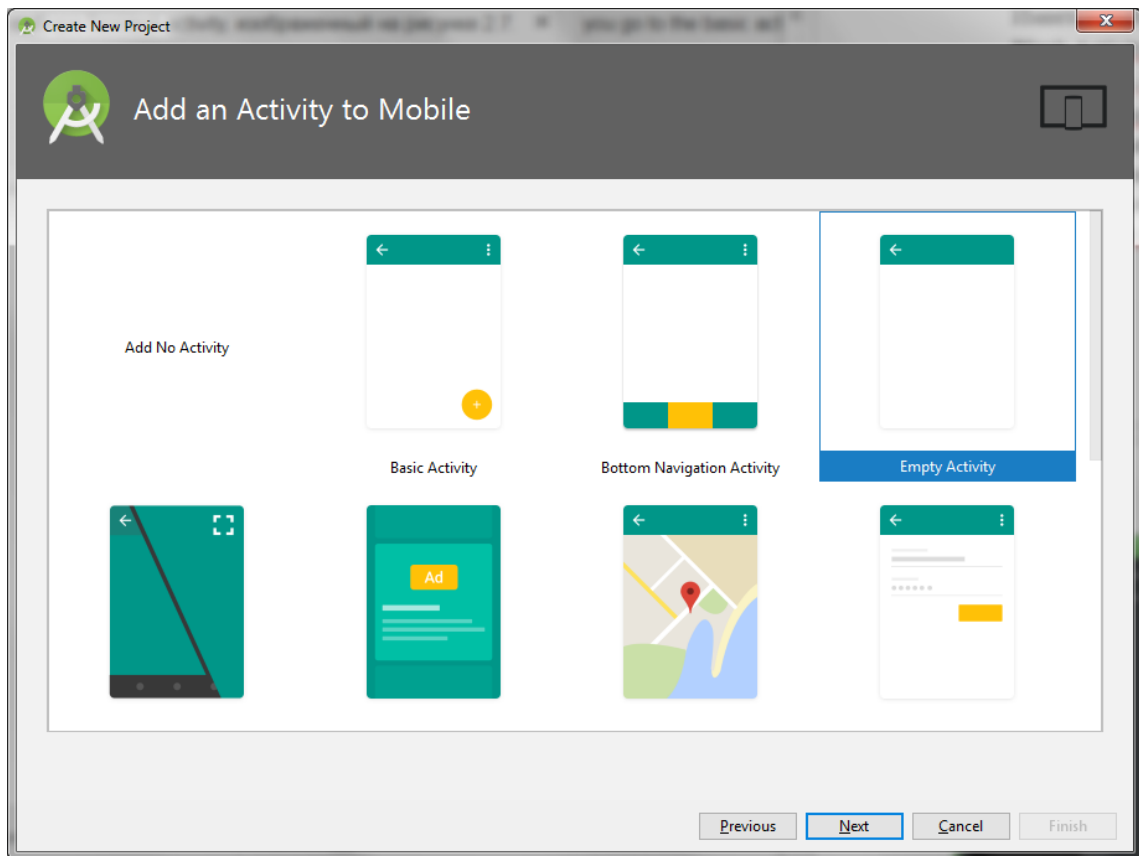


Figure 2.7 – The main activity selection window

There is a choice from a variety of options, for example:

- a) Blank Activity.
- b) Fullscreen Activity.
- c) Master/Detail Activity.

Activity (also called activity) is one of the distinguishing features of the Android framework. In fact, activities are screens of the android application. When the user starts the application, the main activity screen starts, for example, with a list of emails in the mail program, and when the user selects some content for viewing, for example, a letter in the list, it opens on another activity screen. See Activities for more information.

On the Add an activity to Mobile screen, select Empty Activity and click Next. In this case, the development environment will create an empty main activity screen.

Next, we specify the name of our start Activity and the name of its layout as shown in Figure 2.8, as well as the name of the first fragment, as we chose to support the fragments when creating the project. You can also select the Activity navigation type.

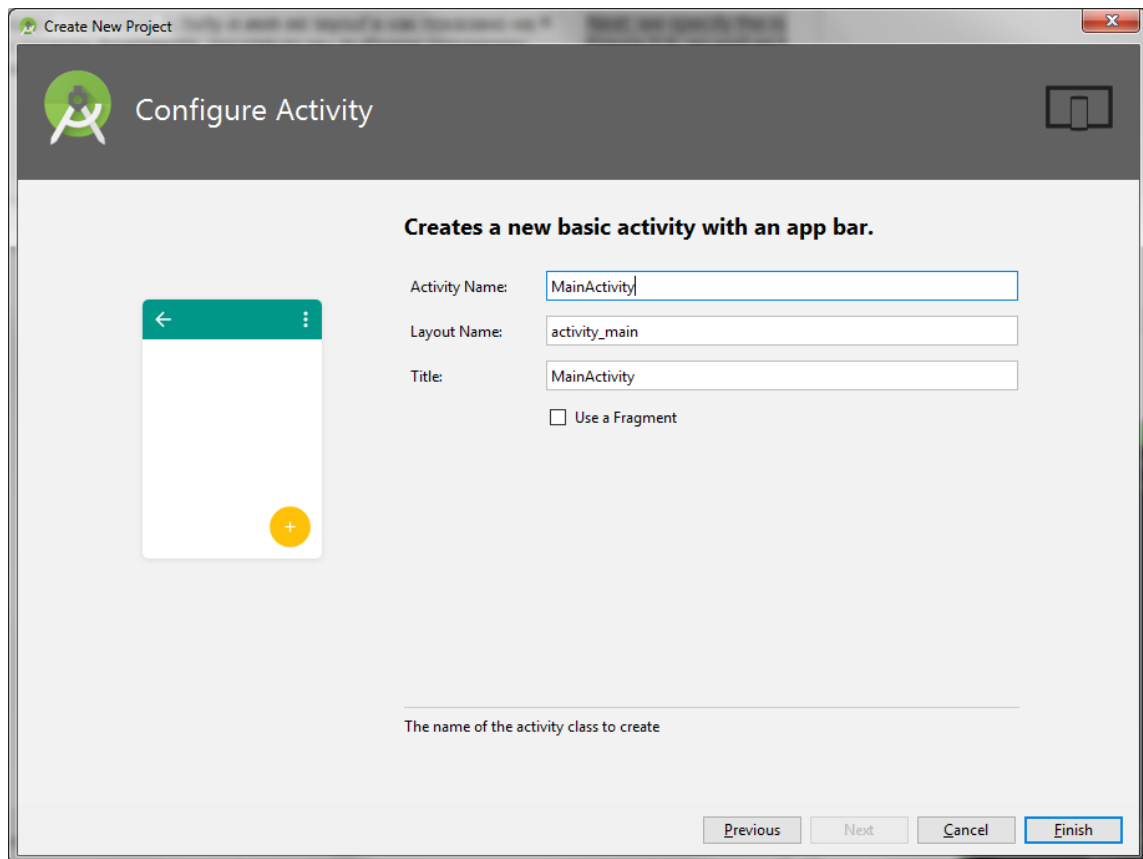


Figure 2.8 – Entering the name of our Start Activity and the name of its layout

After that, we are transferred to the window of the IDE itself. The first thing you should pay attention to is the structure of the project. And yes, it's not like Eclipse.

There are still familiar src and res folders, but res now lies inside the src folder, along with the new java folder into which our packages and classes migrated. This structure of the project Android Studio is due to the new system of project assembly - Gradle. It helps us manage dependencies in our project and connect external libraries, but it's not in this article that is more detailed about it.

It's worth mentioning that when you import projects with the usual structure, everything works just fine.

Figure 2.9 shows the main working window of the development environment.

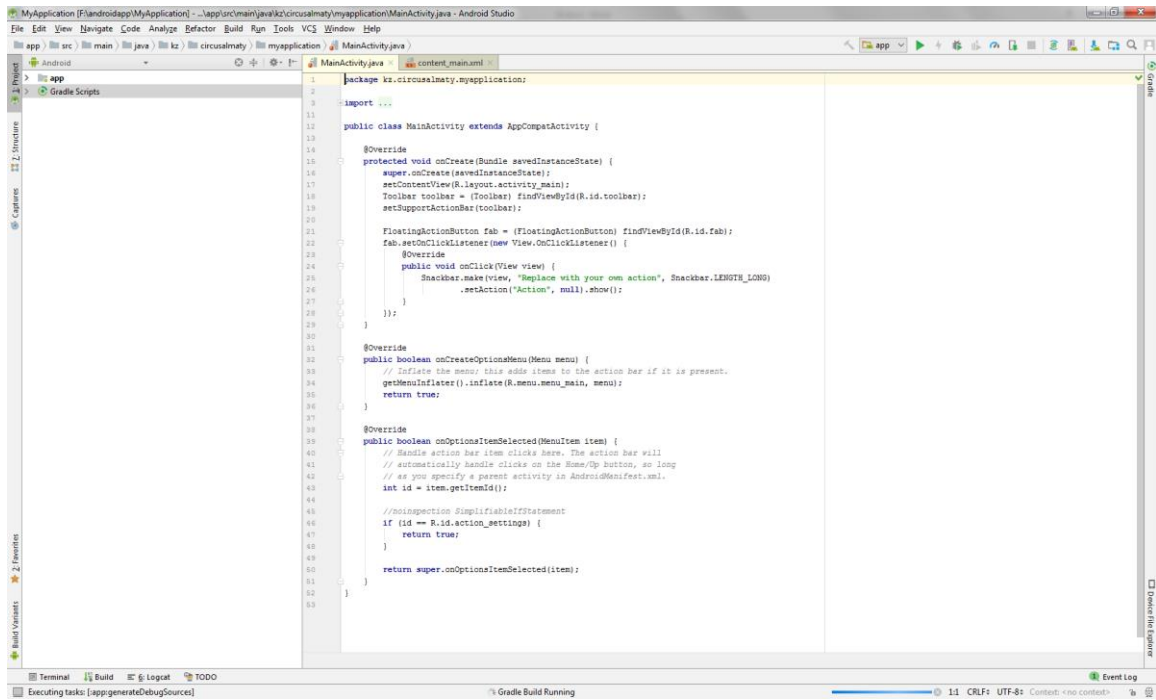


Figure 2.9 – The main IDE window

Figure 2.10 shows a working window of the xml development.

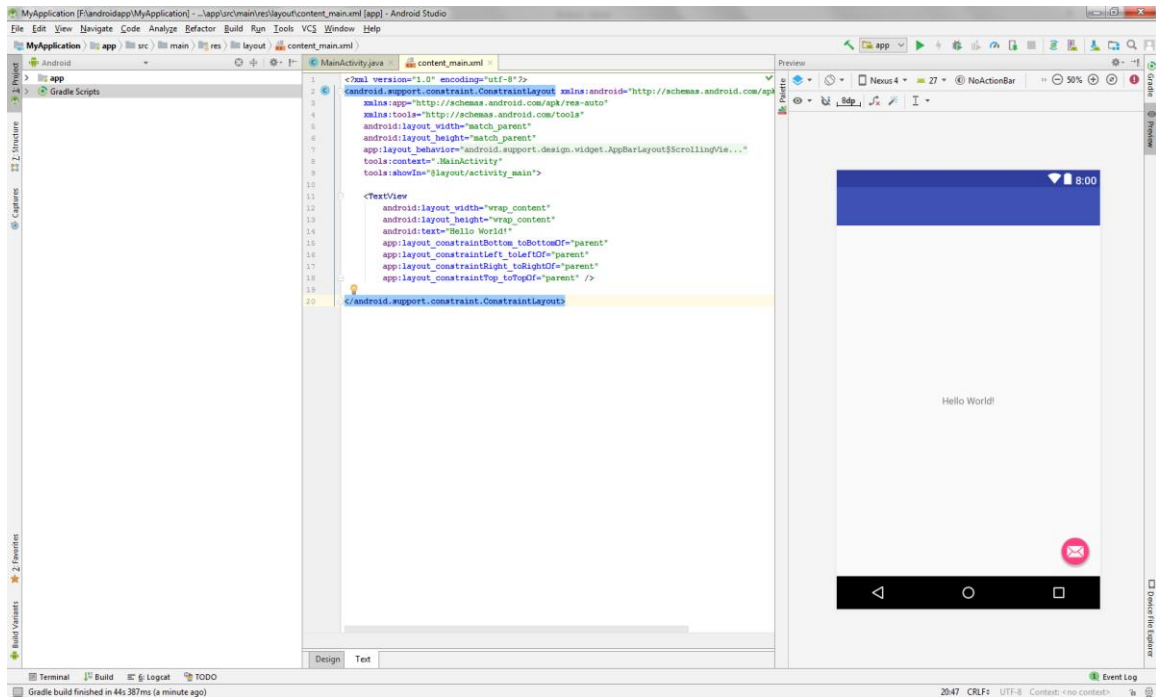


Figure 2.10 – Example of the xml window of the Rector

Having examined Figure 2.10, the following can be noted:

- a) When editing xml in text mode, now there is also a preview.

b) The specified colors and pictures used in the layout'e are displayed on the border in the form of small thumbnails, which easily help to understand what kind of resource you are using.

c) When selecting a resource, its contents are displayed in pop-up JavaDocs, like `@android: color / holo_green_dark`.

d) Resources from `dimens` are automatically displayed by values, and when hovering you can always find out what kind of resource you are using.

Improved integration with Android components.

Let's try to add a new class. We get into the package where we want to place the class, and we press the wonderful combination `alt-insert`. I want to note that hotkey's in Android studio is sometimes quite difficult to remember, in comparison with Eclipse, but carry a much more complex and flexible functionality.

The studio offers us a choice of several objects to create. Briefly about each:

a) Java Class is actually a Java Component. Allows you to create one of the main Java components: Class, Interface, Enum, Annotation and even Singleton.

b) Module - creating, in fact, the module. A module is usually an auxiliary project in Android Studio. Modules in the project will be all external library projects (for example, ActionBarSherlock or Facebook SDK).

c) File - a regular file of any actually with any resolution (txt, json, xml, etc.).

d) Package - the package of our application.

## Creating Android Components

Figure 2.11 shows a window for creating a new application component.

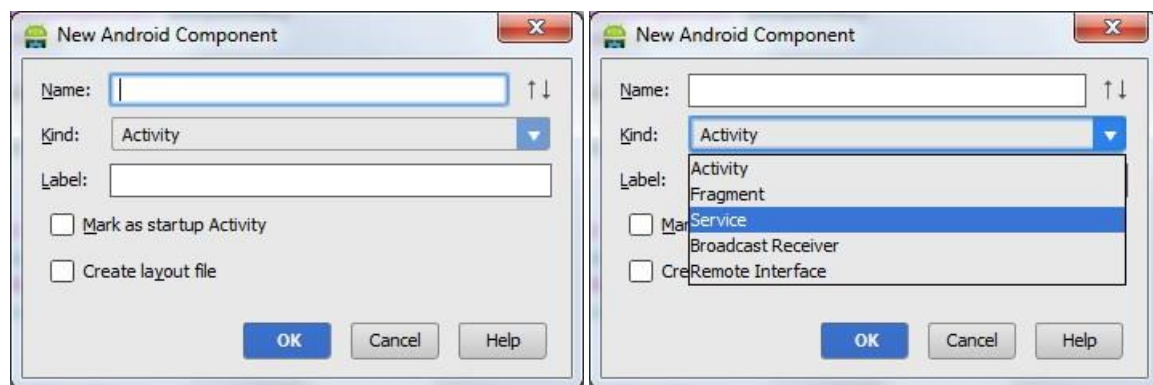


Figure 2.11 – New component creation window

Activity - creates an Activity on one of the predefined templates, immediately registering it in the Manifest.

Android Component is a versatile thing that allows you to create any of the key components of our Android, immediately create a layout for it if necessary, and also label Activity as the starting one. By default, it registers the component in the Manifest, if necessary.

Package-info.java - a file describing the information about the package.

HTML File - actually creates an html file.

Also in the Android Studio is an advanced text editor, which has the following advantages:

- a) Connecting lines between the beginning and end of if, while, switch constructs or methods.
- b) More intelligent code analysis.
- c) Built-in Android Sources connection.
- d) The ability to inherit from the class, or create tests for it in 2 clicks.

## 3 Stages of application development

### 3.1 Design

We are creating a website [www.circusalmaty.kz](http://www.circusalmaty.kz), which will be the basis of our program. The site is written in php language, for ease of use, download it to CMS Wordpress.

Wordpress - a system for managing the content of an open source website; written in PHP; database server - MySQL; Scope - from blogs to fairly complex news resources and online stores. The built-in system of "themes" and "plugins" together with a successful architecture allows you to design projects of broad functional complexity. The complete website screen can be seen on Figure 3.1

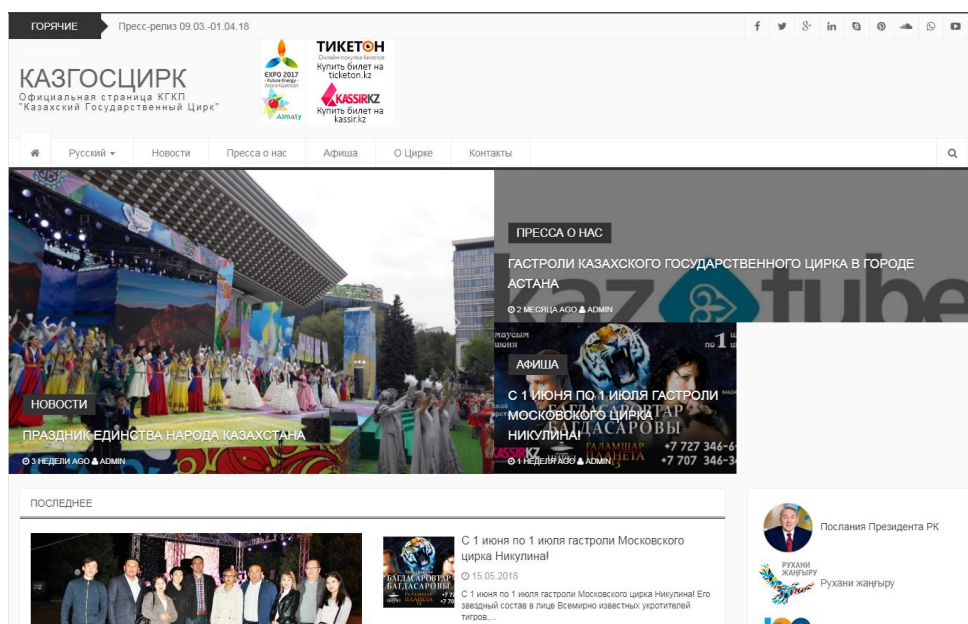


Figure 3.1 – Main page of the site

WordPress powers more than 30% of the web — a figure that tends to rise every day. Everything from simple websites, to blogs and complex portals as well as enterprise websites, and even applications, are built with the help of WordPress.

WordPress tries to combine simplicity for users and publishers with under-the-hood complexity for developers. This makes it flexible while still being easy-to-use.

The following is a list of some of the features that come with WordPress; however, there are literally millions of plugins that extend what WordPress does, so the actual functionality is nearly limitless. You are also free to do whatever you like with the WordPress code, extend it or modify in any way you want or use it for commercial projects without any licensing fees. That is the beauty of free software, free refers not only to price but also the freedom to have complete control over it.

Here are some of the features that everybody will definitely love.



### *Simplicity*

Simplicity makes it possible for you to get online and get publishing, quickly. Nothing should get in the way of you getting your website up and your content out there. WordPress is built to make that happen.

### *Flexibility*

With WordPress, you can create any type of website you want: a personal blog or website, a photoblog, a business website, a professional portfolio, a government website, a magazine or news website, an online community, even a network of websites. You can make your website beautiful with themes, and extend it with plugins. You can even build your very own application.

### *Publish with Ease*

If you've ever created a document, you're already a whizz at creating content with WordPress. You can create Posts and Pages, format them easily, insert media, and with the click of a button your content is live and on the web.

### *Publishing Tools*

WordPress makes it easy for you to manage your content. Create drafts, schedule publication, and look at your post revisions. Make your content public or private, and secure posts and pages with a password.

### *User Management*

Not everyone requires the same access to your website. Administrators manage the site, editors work with content, authors and contributors write that content, and subscribers have a profile that they can manage. This lets you have a variety of contributors to your website, and let others simply be part of your community.

### *Media Management*

They say a picture says a thousand words, which is why it's important for you to be able to quickly and easily upload images and media to WordPress. Drag and drop your media into the uploader to add it to your website. Add alt text, captions, and titles, and insert images and galleries into your content. They have even added a few image editing tools you can have fun with.

### *Full Standards Compliance*

Every piece of WordPress generated code is in full compliance with the standards set by the W3C. This means that your website will work in today's browser, while maintaining forward compatibility with the next generation of browser. Your website is a beautiful thing, now and in the future.

### *Easy Theme System*

WordPress comes bundled with two default themes, but if they aren't for you there's a theme directory with thousands of themes for you to create a beautiful website. None of those to your taste? Upload your own theme with the click of a button. It only takes a few seconds for you to give your website a complete makeover.

### *Extend with Plugins*

WordPress comes packed full of features for every user, for every other feature there's a plugin directory with thousands of plugins. Add complex galleries, social

networking, forums, social media widgets, spam protection, calendars, fine-tune controls for search engine optimization, and forms.

#### *Built-in Comments*

Your blog is your home, and comments provide a space for your friends and followers to engage with your content. WordPress's comment tools give you everything you need to be a forum for discussion and to moderate that discussion.

#### *Search Engine Optimized*

WordPress is optimized for search engines right out of the box. For more fine-grained SEO control, there are plenty of SEO plugins to take care of that for you.

#### *Multilingual*

WordPress is available in more than 70 languages. If you or the person you are building the website for would prefer to use WordPress in a language other than English, that's easy to do.

#### *Easy Installation and Upgrades*

WordPress has always been easy to install and upgrade. If you're happy using an FTP program, you can create a database, upload WordPress using FTP, and run the installer. Not familiar with FTP? Plenty of web hosts offer one-click WordPress installers that let you install WordPress with, well, just one click!

#### *Importers*

Using blog or website software that you aren't happy with? Running your blog on a hosted service that's about to shut down? WordPress comes with importers for blogger, LiveJournal, Movable Type, TypePad, Tumblr, and WordPress. If you're ready to make the move, it has been made easy for you.

#### *Own Your Data*

Hosted services come and go. If you have ever used a service that disappeared, you know how traumatic that can be. If you have ever seen adverts appear on your website, you've probably been pretty annoyed. Using WordPress means no one has access to your content. Own your data, all of it — your website, your content, your data.

#### *Freedom*

WordPress is licensed under the GPL which was created to protect your freedoms. You are free to use WordPress in any way you choose: install it, use it, modify it, distribute it. Software freedom is the foundation that WordPress is built on.

#### *Community*

As the most popular open source CMS on the web, WordPress has a vibrant and supportive community. Ask a question on the support forums and get help from a volunteer, attend a WordCamp or Meetup to learn more about WordPress, read blogs posts and tutorials about WordPress. Community is at the heart of WordPress, making it what it is today.

#### *Plugin System*

The WordPress APIs make it possible for you to create plugins to extend WordPress. WordPress's extensibility lies in the thousands of hooks at your disposal. Once you've created your plugin, they've even got a plugin repository for you to host it on.

### *Theme System*

Create WordPress themes for clients, other WordPress users, or yourself. The WordPress API provides the extensibility to create themes as simple or as complex as you wish. If you want to give your theme away for free you can give it to users in the Theme Repository

### *Application Framework*

If you want to build an application, WordPress can help with that too. Under the hood WordPress provides a lot of the features that your app will need, things like translations, user management, HTTP requests, databases, URL routing and much, much more.

### *Custom Content Types*

WordPress comes with default content types, but for more flexibility you can add a few lines of code to create your own custom post types, taxonomies, and metadata. Take WordPress in whatever direction you wish.

### **The Latest Libraries**

WordPress comes with the latest script libraries for you to make use of. These include jQuery, Plupload, Underscore.js and Backbone.js. They are always on the lookout for new tools that developers can use to make a better experience for our users.

The Qazaq Circus is an application that allows you to view entries from the official page of the Circumstation on the phone. Takes a very small place in the phone memory. This application has a simple and convenient interface. The appearance of the application should be combined with the rest of the user interface elements of the operating system, the platform's advanced development experience. The design of the PP must strictly follow the established standards and requirements of the carrier platforms (in this case Google). Using all the above rules, the concept of the Qazaq Circus interface is being developed.

The main interface of the program is presented in two main parts: the main page and the category page.

The interface of the main menu of the program is presented to the users of the on-screen display on which the list of the latest records located on the site of the Kazgotsirka is placed (Figure 3.2).

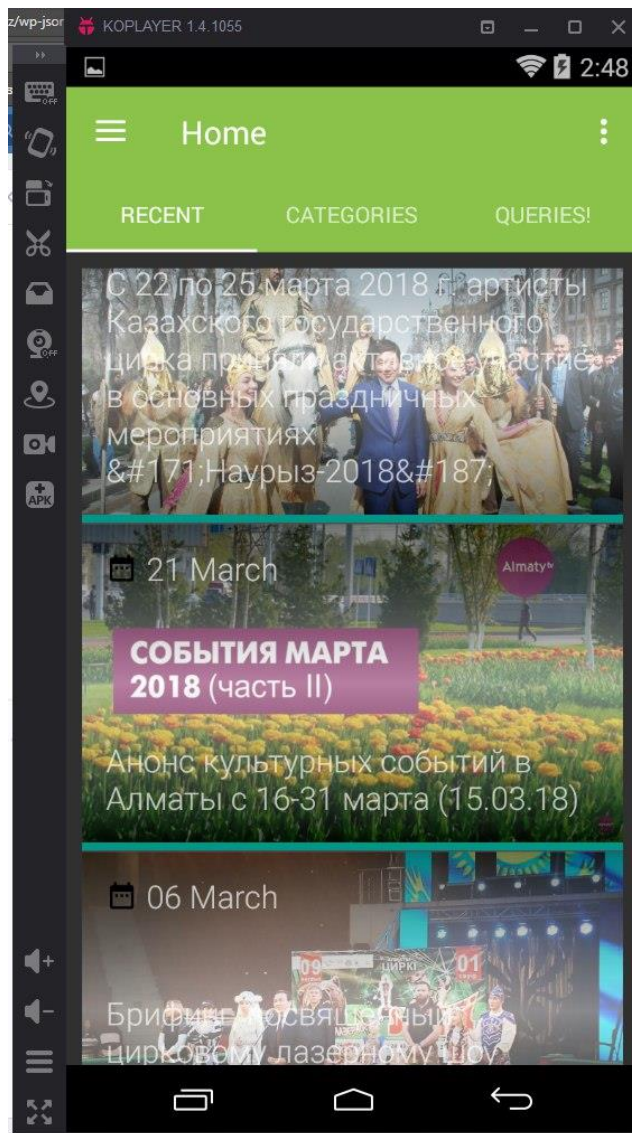


Figure 3.2 – Application home page

It is possible to view records by category, as shown in Figure 3.3. The categories have the following structure:

- a) News
- b) Poster
- c) Press about us
- d) State symbols
- e) Events

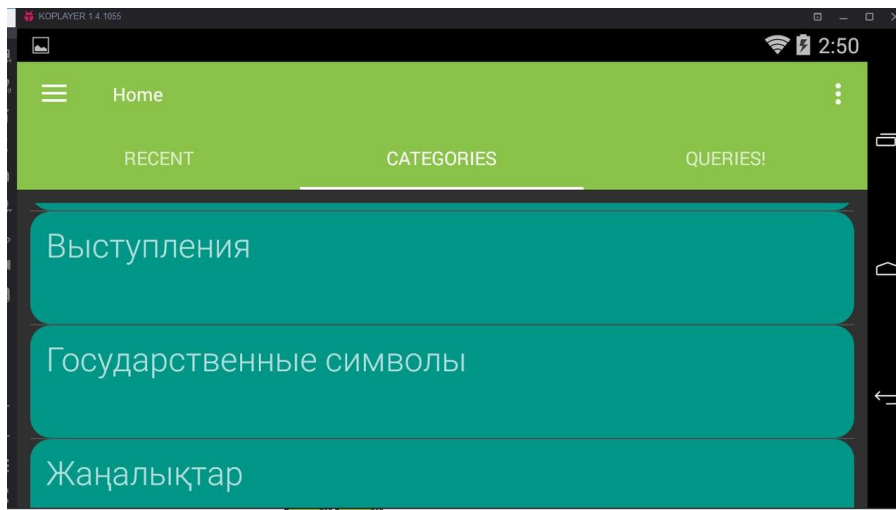


Figure 3.3 – Categories

This window shows us the possibility of our application to open and show you site categories that are available and can be highlighted on the official website of the Kazakh state circus.

The list of categories, as can be seen in Figure 3.2, consists of:

- a) news in Russian
- b) news in the Kazakh language
- c) news in English
- d) press about us in Russian
- e) press about us in Kazakh
- f) press about us in English
- g) a poster in Russian
- h) the poster in the Kazakh language
- i) the poster of us in English
- j) contacts in Russian
- k) contacts in the Kazakh language
- l) contacts us in English
- m) about the circus in Russian
- n) about the circus in the Kazakh language
- o) about the circus of us in English

Each record can be opened separately. It is possible to view images and videos from the site. In addition, you can change the orientation of the phone and the application will adjust to the width of the screen, which is a very convenient function, especially when reading a large amount of data.

This is demonstrated in Figures 3.4-3.5



Figure 3.4 – Post from the main page

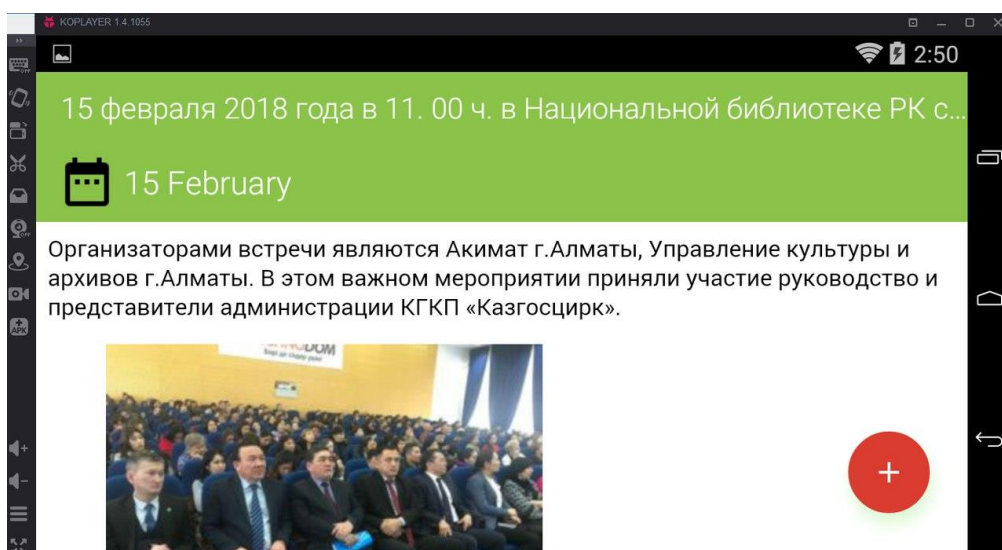


Figure 3.5 – Post from the main page

## 3.2 Application development

The project consists of several folders and files. The main ones are:

a) src is the source code for Java. Here is the main file for work. There will be new classes;

b) gen - files generated by Java itself;

c) res - resource files. Contains several subdirectories:

d) res / drawable- \* dpi - these five folders contain resources for different screen extensions. If you go into each folder, you can find there the file ic\_launcher.png, which is the icon of your application. There is nothing in the drawable-ldpi folder, since this is a folder for old phones that you no longer need to support.

a) res / layout - this folder contains xml-files that describe the appearance of forms and various form elements. After the project is created, there already exist activity\_main.xml and fragment\_main.xml.

b) res / menu - here are the resources for the menu.

c) res / values - here we have some string resources, resources of colors, themes, styles and dimensions that we can use in our project. Also there are similar folders valuesw820dp, values-v11 (for tablets Android 3.0), values-14 (for Android 4) intended for certain types of devices.

When developing the application, the individual components of the application were drawn in Adobe Illustrator CC:

a) application icon;

b) markers of the beginning and end of the cropped file;

c) icons for help buttons, reset, save, add a new file, write a file, etc.

### 3.2.1 Creating and implementing Activities

The Activity class is an essential component of the Android application, and the way to start and link actions is a fundamental part of the platform application model. Unlike software paradigms in which applications are launched using the main () method, the Android system initiates code in the Activity instance, calling special callback methods that correspond to specific stages of its life cycle.

A small number of activities were created in the / layout folder, which consist of the following files:

a) activity contact us

b) activity about the app

c) activity home

d) activity post

e) activity post of particular category

f) activity settings

g) activity site page loader

h) activity splash screen

i) app bar home

- j) content home
- k) fragment ask your problem
- l) fragment categories
- m) fragment recent
- n) nav header home
- o) row
- p) toolbar

The implementation of these forms is given in Appendix A

### 3.2.2 Creating and implementing Classes

Classes were created in the / java folder, which are the following files:

- a) AboutTheAppActivity
- b) AskYourProblemActivity
- c) Categories
- d) ContactUs
- e) Home
- f) PermissionTransferToHomeActivity
- g) Post
- h) PostsOfParticularCategory
- i) Recent
- j) SettingsActivity
- k) SitePageLoader
- l) SplashScreen

The implementation of these classes is given in Appendix B

After creating all forms and classes next thing we must do is to connect our application with data on the website. We are using json requests to get information from database through wordpress. The problem is wordpress doesn't give an access to its database, so we need to use some plugins to get Wordpress API shown on Figure 3.6.

|                          |  |   |
|--------------------------|--|---|
| <input type="checkbox"/> | <b>Better REST API Featured Images</b>                               | Adds a top-level field with featured image data including available sizes and URLs to the post object returned by the REST API.                                       |
| <input type="checkbox"/> | <b>WP REST API</b><br><a href="#">Деактивировать</a>                 | JSON-based REST API for WordPress, originally developed as part of GSoC 2013.<br>Версия 2.0-beta15   Автор: <a href="#">WP REST API Team</a>   <a href="#">Детали</a> |
| <input type="checkbox"/> | <b>WP REST API - filter fields</b><br><a href="#">Деактивировать</a> | Enables you to filter the fields returned by the api.<br>Версия 1.0.7   Автор: <a href="#">Stephan van Rooij</a>   <a href="#">Детали</a>                             |

Figure 3.6 – Wordpress API plugins

WP REST API – helps us to use Wordpress API with no security dangers. Whilst Better REST API Featured Images gets better quality photos and allows to



use them in the application. And WP REST API – filter fields – is a plugin which gives us a way to sort fields and get only that kind of information we need, for example, if one needs to get only “id” and “date” – they can just write “fields=id,date”.

Json is chosen because it can be read by human and is very practical. We are creating a class called Const, where we put json requests, as we mentioned before. Each request gets some sort of information from database and allows us to it in other parts of the project.

```

public static final String url = "https://circusalmaty.kz/wp-
json/wp/v2/posts?categories=2&per_page=100&fields=id,date,link,title,better_feat
ured_image";
public static final String get_categories_url = "https://circusalmaty.kz/wp-
json/wp/v2/categories?fields=id,description,name";
public static final String get_all_posts_of_category_url =
"https://circusalmaty.kz/wp-
json/wp/v2/posts?categories=CATEGORY_ID&fields=id,date,link,title,better_feat
ured_image";
public static final String get_content_by_id = "https://circusalmaty.kz/wp-
json/wp/v2/posts/POST_ID?fields=content";
public static final String pages = "https://circusalmaty.kz/wp-
json/wp/v2/pages/PAGE_ID?fields=content";

```

It is also important to notice that human can't read json requests' result, as it's just a flow of different information, which is show on Figure 3.7.

```

[{"id":1581,"date":"2018-06-
01T18:27:01","link":"https://circusalmaty.kz/news/den-zashhityi-detey-
v-kazgostsirke.html","title":{"rendered":"\u0414\u0435\u043d\u044c
\u0437\u0430\u0449\u0438\u0442\u044b \u0434\u0435\u0442\u0435\u0439
\u0414\u0435\u0437\u0437\u043e\u0441\u0442\u0438\u0442\u0435\u0439
\u0414\u0435\u0437\u0437\u043e\u0441\u0442\u043e\u0441\u0442\u0435\u0439"},"bette
r_featured_image":
{"id":1596,"alt_text":"","caption":"","description":"","media_type":"image"
,"media_details":
{"width":5184,"height":3456,"file":"2018/06/IMG_1527.jpg","sizes":
{"thumbnail":{"file":"IMG_1527-150x150.jpg","width":150,"height":150,"mime-
type":"image/jpeg","source_url":"https://circusalmaty.kz/wp-
content/uploads/2018/06/IMG_1527-150x150.jpg"},"medium":
{"file":"IMG_1527-300x200.jpg","width":300,"height":200,"mime-
type":"image/jpeg","source_url":"https://circusalmaty.kz/wp-
content/uploads/2018/06/IMG_1527-300x200.jpg"},"medium_large":
{"file":"IMG_1527-768x512.jpg","width":768,"height":512,"mime-

```

Figure 3.7

But there is an online viewer, which helps us to see the information we get in a very simple way as shown on Figures 3.8-3.9. This is needed to correctly wirte json requests and see all fields as a tree, so we can be sure of the information we are asking for.

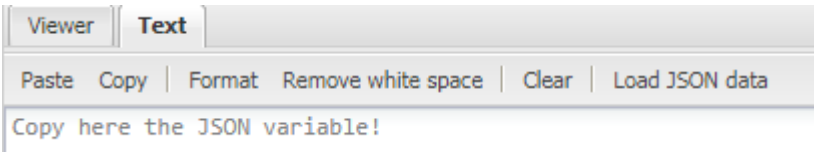


Figure 3.8 – Main page of <http://jsonviewer.stack.hu/>

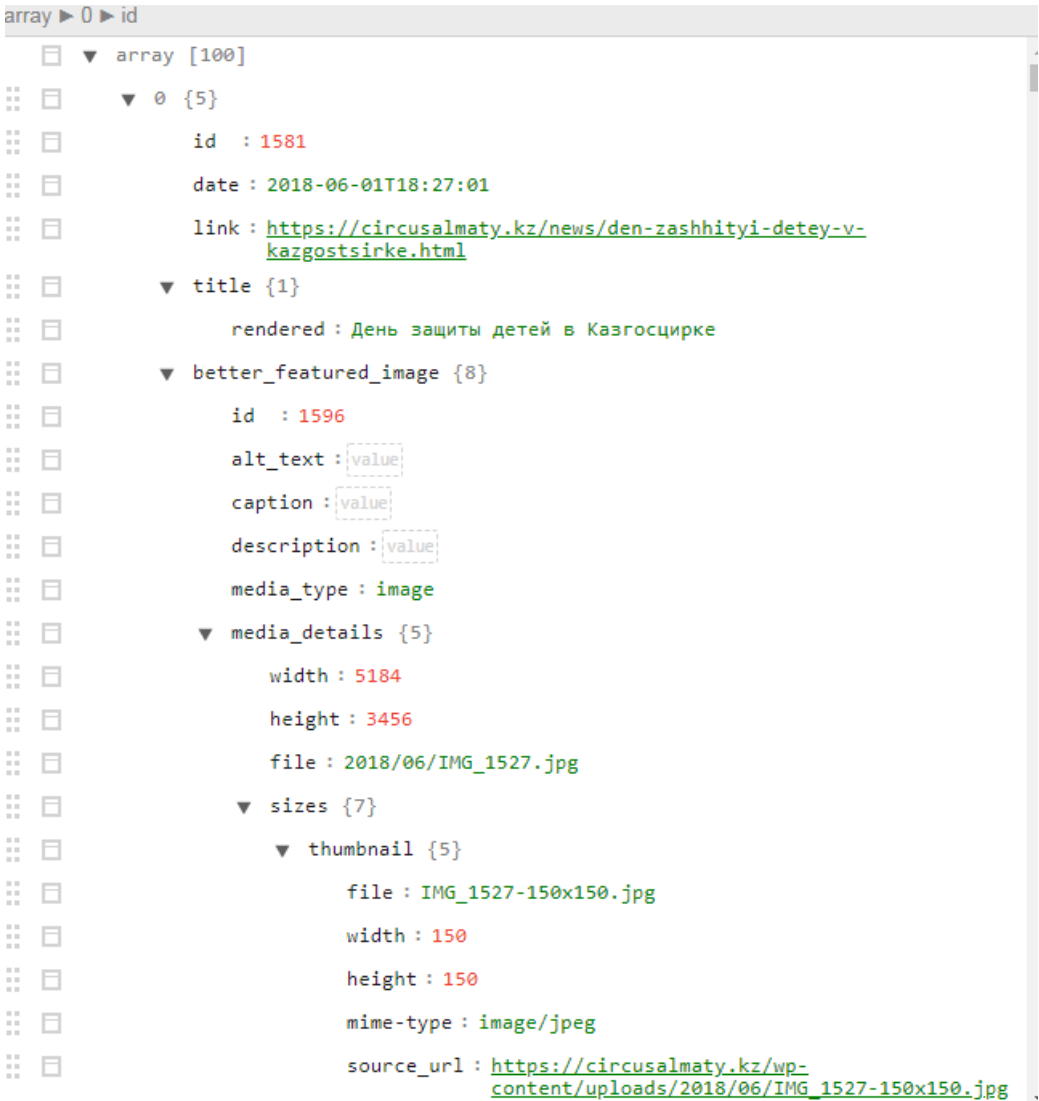


Figure 3.9 – Json viewer

## 4 Feasibility study

### Description of work and justification of the need

The theme of this diploma project is "Developing of "The Qazaq Circus" a mobile application on the Android platform»

The mobile application for the Kazakh State Circus is an innovation, because we have never had anything like this before. The application allows the user to find out all the news about the circus, and easily see what upcoming events are.

### 4.1 Calculation of costs for development of information technology

Calculation of the total costs for the development of a design solution in the form of information technology ( $C_{\text{ии}}$ ) is carried out by the formula

$$C_{\text{ии}} = Z_{\text{фот}} + Z_{\text{csi}} + M_i + P_{\text{ci}} + P_{\text{mi}} + P_{\text{hki}} + \Pi_{\text{zi}} + P_{\text{hi}} \quad (4.1)$$

where  $Z_{\text{фот}}$  – total payroll of developers, tenge;

$Z_{\text{csi}}$  – deductions for social tax, tenge тенге;

$M_i$  – cost of materials, tenge;

$P_{\text{ci}}$  – costs for special software needed to develop a project solution, tenge;

$P_{\text{mi}}$  – costs associated with the operation of machinery, tenge;

$P_{\text{hki}}$  – expenses for scientific business-trips, tenge;

$\Pi_{\text{zi}}$  – other expenses, tenge;

$P_{\text{hi}}$  – overheads, tenge.

The size of the payroll of developers ( $Z_{\text{фот}}$ ) is calculated by the formula

$$Z_{\text{фот}} = Z_{\text{oi}} + Z_{\text{di}} \quad (4.2)$$

where  $Z_{\text{oi}}$  – basic salary, tenge;

$Z_{\text{di}}$  – additional wage, tenge.

The cost of labor depends on the volume and complexity of software development. The total volume ( $V_0$ ) of the software product is determined based on the number and volume of the function implemented by the program

$$V_0 = \sum V_i, \quad (4.3)$$

where  $V_i$  – the volume of software's certain function;

$n$  – total number of functions.

From Appendix C, we set the volume of the software (the source code lines, LOC) are 17265. Thus,

$$V_0 = 17265 \text{ code lines}$$

The total complexity of the project is calculated by the formula (4.4):

$$T_o = T_H \cdot K_c \cdot K_T \cdot K_H \quad (4.4)$$

where  $K_c$  – coefficient that takes into account the complexity of the software;  
 $K_T$  – correction coefficient, taking into account the degree use in the development of standard modules;  
 $K_H$  – coefficient, taking into account the degree of novelty of software;  
 $T_H$  – normative labor intensity.

The complexity factor is determined on the basis of the data presented in (Table 4.1) and is  $K_c = 0.26$ , because in the project there are more than 3 characteristics – the functioning of the system, interactive access, storage, maintenance and retrieval of data in complex structures.

Table 4.1 – Additional software complexity factors

| Characteristics of the Software  | Values of $K_c$ |
|--|-----------------|
| 1. Operation of the software in the extended operating environment (communication with other software) | 0,08            |
| 2. Interactive access  | 0,06            |
| 3. Ensuring the storage, management and retrieval of data in complex structures                        | 0,07            |
| 4. The presence of the software simultaneously several characteristics of the table. D.4.1, annex D    |                 |
| a) 4.1 2 characteristics   | 0,12            |
| b) 4.2 3 characteristics   | 0,18            |
| c) 4.3 More than 3 characteristics   | 0,26            |

The correction factor, taking into account the degree of use for the development of the project of standard modules ( $K_T$ ), is determined on the basis of the data presented in Table 4.2 and is 0.8, because

The correction factor, taking into account the novelty of the project being developed ( $K_H$ ) is determined on the basis of the data presented in Table 4.3 and is 0.7; The old OS environment is used and a new type of PC (novelty category B) is not used.

The basis for determining the normative complexity are, the enlarged norms of time for the development of the project, depending on the specified volume of software and complexity group.

The normative complexity of the project ( $T_H$ ) is determined on the basis of the calculated volume of PP and the complexity category, which is specified taking into account the complexity and novelty of the project and the degree of use of standard modules in the development.

Table 4.2 – Values of a correction factor that takes into account the use of standard modules of typical programs and software ( $K_T$ )

| The degree of coverage of the implemented functions of the software being developed by standard modules, standard programs and software | Values $K_T$ |
|---|--------------|
| 1. From 60 % and above  | 0,6          |
| 2. From 40 % to 60  | 0,7          |
| 3. From 20 % to 40 %  | 0,8          |
| 4. Up to 20 %   | 0,9          |
| 5. Typical programs and software not used to implement the function of the software being developed                                     | 1,0          |

Table 4.3 – Correction factors that take into account the novelty of software ( $K_H$ )

| Category of novelty | Degree of novelty   | Usage                     |                         | Values $K_H$ |
|---------------------|---|---------------------------|-------------------------|--------------|
|                     |   | Based on a new type of PC | In a new OS environment |              |
| A                   | Fundamentally new software,   | +                         | +                       | 1,75         |
|                     | Not having available analogs  | – +                       | +                       | 1,6          |
|                     |   | –                         | –                       | 1,2          |
|                     |   | –                         | –                       | 1,0          |
| B                   | Software, which is the development of a certain parametric series of software   | +                         | +                       | 1,0          |
|                     |   | – +                       | –                       | 0,9          |
|                     |   | –                         | –                       | 0,8          |
| B                   | Software, which is the development of a certain parametric series of software developed for previously mastered types of PC and OS configurations | –                         | –                       | 0,7          |

For the 1st category of complexity of software  $T_H = 520$ .

We calculate the total volume of labor intensity starting from formula (4.4):

$$T_o = 520 \cdot 0,26 \cdot 0,8 \cdot 0,7 = 72,8 \text{ people/day}$$

Based on the complexity, the planned number of developers ( $\Psi_p$ ) and the planned timeframes required for the project as a whole ( $T_p$ ) are determined. The following tasks can be solved:

a) Calculation of the number of performers at the given terms of project development;

b) Determining the timing of project development for a given number of performers.

The number of project executors ( $\Psi_p$ ) is calculated by the formula (4.5)

$$\Psi = T_o / (T_p \cdot \Phi_{\text{эф}}) \quad (4.5)$$

where  $\Phi_{\text{эф}}$  – effective work time fund of one employee in the course of the year (days);

$T_o$  – total complexity of project development (person / day);

$T_p$  – the term of the project (years).

The term of the project development ( $T_p$ ) is determined by the formula (4.6)

$$T_p = T_o / (\Psi_p \cdot \Phi_{\text{эф}}) \quad (4.6)$$

where  $\Psi_p$  – planned number of developers.

An effective work time fund for one employee ( $\Phi_{\text{эф}}$ ) is calculated by the formula (4.7)

$$\Phi_{\text{эф}} = D_r - D_n - D_b - D_o \quad (4.7)$$

where  $D_r$  – number of days in a year;

$D_n$  – number of holidays in a year;

$D_b$  – number of days off per year;

$D_o$  – number of days on vacation.

Thus, in accordance with the production calendar for 2018

$D_r = 365$ ;  $D_n = 8$ ;  $D_b = 110$ ;  $D_o = 24$ , effective time fund of one worker will be

$$\Phi_{\text{эф}} = 365 - 8 - 110 - 24 = 223 \text{ days}$$

The planned number of developers  $\Psi_p = 1$ , therefore by formula (4.6)

$$T_p = 72,8 / (1 \cdot 223) = 0,32 \text{ years}$$

Thus, according to calculations made and in accordance with formula (4.4)

$$\Psi = 72,8 / (0,32 \cdot 223) = 1 \text{ person.}$$

The basic salary of performers for specific software is calculated by the formula (4.8)

$$Z_{oi} = \sum_{i=1}^n T_{\text{чi}} \cdot T_{\text{ч}} \cdot K \quad (4.8)$$

where  $n$  – number of executors engaged in the development of a specific software;

$T_{\text{чi}}$  – hourly rate of the  $i$ -th performer (thousand KZT);

$\Phi_n$  – planned fund of working time of  $i$ -th executor (days), number of working days per month;

$T_{\text{ч}}$  – number of hours of work per day (hour), 8 hours;  $K$  – bonus factor, is 1,38.

The average monthly number of working days for 2018, according to the production calendar for 2018, with a five-day working week

$$\Phi_{\pi} = 20,42 \text{ days}$$

According to the specifics and complexity of the functions performed, the staffing of the group of executing specialists participating in software development is compiled, with the definition of education, specialty, qualification and position (table 4.4).

Table 4.4 – Information on employees involved in the project

| Specialist Executor | Number | Wage per month, tenge |
|---------------------|--------|-----------------------|
| Engineer-programmer | 1      | 220 000,00            |
| Total               | 1      | 220 000,00            |

The work of the programmer-developer is assumed conditionally on a contractual basis, in the amount of 220,000.0 tenge.

The hourly wage rate is calculated by formula (4.9) by dividing the monthly wage rate established at a 40-hour weekly working time rate and the total time pool ( $\Phi_p$ )

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_p} \quad (4.9)$$

where  $T_{\text{ч}}$  – hourly tariff rate (thousand tenge);

$T_{\text{м}}$  – monthly tariff rate (thousand tenge).

Common fund of time by formula (4.10)

$$\Phi_p = T_{\text{ч}} \cdot \Phi_{\pi} \quad (4.10)$$

In this way using formula (4.10)

$$\Phi_p = 8 \cdot 20,42 = 163,36 \text{ hours}$$

Tariff rate of the project developer

$$T_{\text{чи}} = 220000 / 163,36 = 1346,72 \text{ tenge per hour}$$

In accordance with formula (4.8), the basic wage of the head is

$$3_{oi} = 1346,72 \cdot 8 \cdot 20,42 = 220\,000 \text{ tenge} \quad (4.18)$$

The results of calculating the basic salary are presented in Table 4.5.

Table 4.5 - Summary of the calculation of basic salary costs

| The name of the content of works | Performer           | Labor intensity normative-hour | Salary per hour work, tenge / hour | The amount of wage, tenge |
|----------------------------------|---------------------|--------------------------------|------------------------------------|---------------------------|
| TK                               | Engineer-Programmer | 40                             | 1346,72                            | 53 868,80                 |
| Modeling                         | Engineer-Programmer | 40                             | 1346,72                            | 53 868,80                 |
| Programming                      | Engineer-Programmer | 60                             | 1346,72                            | 80 803,20                 |
| Testing                          | Engineer-Programmer | 14                             | 1346,72                            | 18 854,08                 |
| Introduction                     | Engineer-Programmer | 10                             | 1346,72                            | 13 467,20                 |
| Total                            |                     | 164                            |                                    | 220 000,00                |

The additional wage is 10% of the basic wage and is calculated by the formula (4.11)

$$З_{\text{дi}} = З_{\text{oi}} \cdot H_{\text{д}} / 100 \quad (4.11)$$

where  $H_{\text{д}}$  – coefficient of additional wages of developers 23%.

$$З_{\text{дi}} = 220\,000 \cdot 0,23 = 50\,600,00 \text{ tenge}$$

The social tax makes 11% (item 358 item 1 TC RK) from the income of the worker, and is calculated by formula (4.12)

$$З_{\text{ci}} = (\Phi\text{OT} - \Pi\text{O}) 11\% \quad (4.12)$$

where  $\Pi\text{O}$  – pension contributions, which account for 10% of the WFP and are not subject to social tax

$$\Pi\text{O} = \Phi\text{OT} 10\% \quad (4.13)$$

In this way

$$\Pi\text{O} = 220\,000 \cdot 0,1 = 22\,000 \text{ tenge}$$

$$З_{\text{ci}} = (220\,000 - 22\,000) \cdot 0,11 = 21\,780,0 \text{ tenge}$$

Table 4.6 - Expenses for material resources and software



| Name of material resource             | Unit of measurement | Amount of material consumed | Unit price, tg | Sum, tg        |
|---------------------------------------|---------------------|-----------------------------|----------------|----------------|
| Lenovo Ideapad 100 laptop             | pcs                 | 1                           | 120 000        | 120 000        |
| OC Windows 10                         | pcs                 | 1                           | free of charge | free of charge |
| Android Studio v 2.2.3                | pcs                 | 1                           | free of charge | free of charge |
| Java v7                               | pcs                 | 1                           | free of charge | free of charge |
| TOTAL expenses for material resources |                     |                             |                | 120 000        |

Depreciation Rate ( $H_a$ ) is calculated by the formula (4.14):

$$H_a = \frac{C_{\text{обор}}}{T_{\text{НОРМ}} \cdot C_{\text{обор}}} \cdot 100\% \quad (4.14)$$

where  $T_{\text{НОРМ}}$  – normative service life (for laptop – 4 years).

$$H_a = \frac{120\,000}{4 \cdot 120\,000} \cdot 100\% = 25\%$$

Depreciation deductions are made according to the established depreciation rates, expressed as a percentage of the book value of equipment and are calculated by the formula (4.15):

$$A = \frac{C_{\text{обор}} \cdot H_a \cdot N}{100 \cdot 12 \cdot t} \quad (4.15)$$

where  $H_a$  – depreciation rate;  
 $C_{\text{обор}}$  – initial cost of equipment;  
 $N$  – time of use of a personal computer;  
 $t$  – number of working days per month.

Calculation of depreciation charges:

$$A = \frac{120\,000 \cdot 25 \cdot 164}{100 \cdot 12 \cdot 21} = 19\,523,81 \text{ tenge}$$

The total expense for material resources ( $3_M$ ) is determined by formula (4.16):

$$З_M = \sum_{i=1}^n P_i * Ц_i \quad (4.16)$$

where  $P_i$  – consumption of the  $i$ -th kind of material resource, natural units; $Ц_i$   
 - цена за единицу  $i$ -го вида материального ресурса, тг;  
 $i$  – type of material resource;  
 $n$  – number of types of material resources.

It is necessary to calculate the costs for electricity in the form given in the table 4.7.

Таблица 4.7 - Затраты на электроэнергию

| Name of equipment         | Passport capacity, kW | Coefficient of power use | The operating time of equipment for the development of PP, h | The price of electricity, tg / kW * h | Sum, tg |
|---------------------------|-----------------------|--------------------------|--|---------------------------------------|---------|
| Lenovo Ideapad 100 laptop | 0,45                  | 0,7                      | 164  | 27,84                                 | 1 438,2 |
| TOTAL expenses for energy |                       |                          |  |                                       | 1 438,2 |

The total amount of electricity expenses ( $З_э$ ) is calculated by formula (4.17):

$$З_э = \sum_{i=1}^n M_i * K_i * T_i * Ц \quad (4.17)$$

where  $M_i$  – passport capacity of the  $i$ -th electrical equipment, kW;  
 $K_i$  – power factor of the  $i$ -th electrical equipment (accepted as  $K_i=0.7,0.9$ );  
 $T_i$  – the operating time of the  $i$ -th equipment for the entire period of development of PP h;  
 $Ц$  – price of electricity, tg / kWh;  
 $i$  – type of electrical equipment;  
 $n$  – amount of electrical equipment.

$$З_э = 0,45 \cdot 0,7 \cdot 164 \cdot 27,84 = 1\,438,2 \text{ tenge}$$

Expenses under the item "Overhead" ( $P_{hi}$ ), calculated by the standard ( $H_{PH}$ ) as a percentage of the basic wage of performers. The standard is established as a whole for the organization is calculated by formula (4.18).

$$P_{hi} = З_{oi} \cdot H_{PH} / 100\% \quad (4.18)$$

where  $P_{hi}$  – overhead costs for a specific software (thousand KZT);  
 $H_{PH}$  – norm of overhead costs in general for the organization (20%).  
 Thus, the overhead will be:

$$P_{hi} = 220\,000 \cdot 0,2 = 44\,000 \text{ tenge}$$

The full cost of researching the technology of developing applications for mobile devices, formula (4.1):

$$C_{ni} = 220\,000 + 21\,780 + 44\,000 + 19\,524 + 1\,438 = 306\,742 \text{ tenge}$$

Summary results of calculation of software development costs and their structure are presented in (Table 4.8) and on (Figure 4.1).

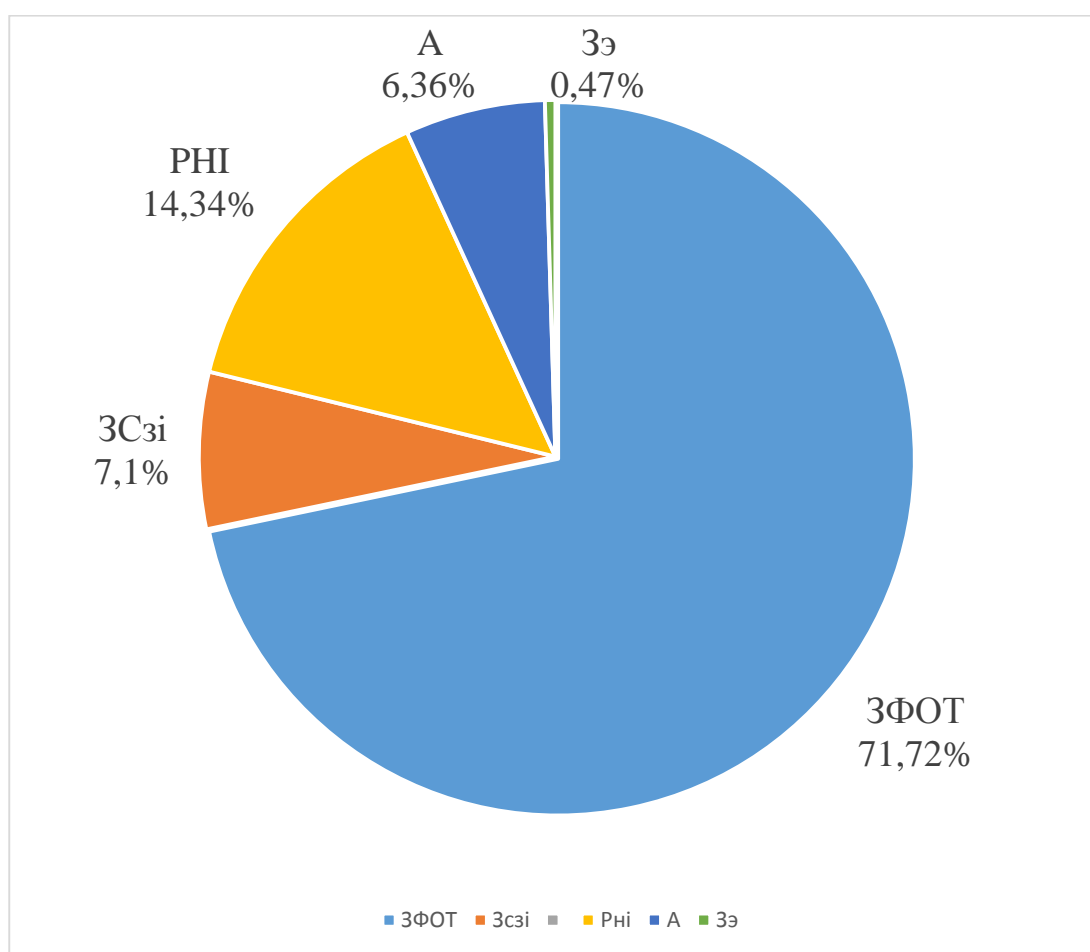


Figure 4.1 - Cost structure for the development of PP

Table 4.8 - Research costs of mobile application development technology

| Development costs | Conditional symbol | Value, tenge | percentage of the total amount |
|-------------------|--------------------|--------------|--------------------------------|
| Wage fund         | 3ΦOT               | 220 000      | 71,72                          |
| Social tax        | 3C3i               | 21 780       | 7,1                            |

Continuation of Table 4.8

|                         |          |         |       |
|-------------------------|----------|---------|-------|
| Overheads               | $P_{hi}$ | 44 000  | 14,34 |
| Depreciation deductions | A        | 19524   | 6,36  |
| Electricity expenses    | $З_э$    | 1 438   | 0,47  |
| Total:                  |          | 220 000 | 100   |

## 4.2 Calculating the price of a software product

Calculation of the price of ПП, which is developed by one organization on the order of another and will not be replicated, is carried out according to the formula (4.19)

$$Ц_{ПП} = З_{ПП} + \Pi_{\Pi} + НДС \quad (4.19)$$

where  $Ц_{ПП}$  – price of a software product, tenge;  
 $З_{ПП}$  – the cost of developing a project solution, in this case a software product, tenge;

$\Pi_{\Pi}$  – planned profit, tenge;

VAT – value added tax, tenge.

The planned profit is (20%) of the cost of development.

$$\Pi_{\Pi} = 306\,742 \cdot 0,2 = 61\,348,4 \text{ tenge}$$

VAT accrued on the PP is determined as follows by formula (4.20)

$$VAT = (З_{ПП} + \Pi_{\Pi}) \cdot k_{НДС} \quad (4.20)$$

where  $k_{НДС}$  – rate of value added tax. VAT is calculated using formula (4.20)

$$VAT = (306\,742 + 61\,348,4) \cdot 0,12 = 44\,170,85 \text{ tenge}$$

The price of the software product is calculated by the formula (4.19):

$$Ц_{ПП} = 306\,742 + 61\,348,4 + 44\,170,85 = 412\,261,25 \text{ tenge}$$

## 4.3 Conclusion

The total cost of development of the application for The Kazakh State Circus on OS Android amounted to 412 261.25 tenge, which includes possible costs for the development of the software product.

The largest share in the cost of the project is the cost of staff salaries, more than 50% of the costs involved in the process of this development.

## **5 Safety of life**

Development of a mobile application for The Kazakh State Circus on the Android platform. This project will help users to obtain the necessary information about the events in the Kazakh State Circus. The application is developed in the office, in the building of this organization. Two people will work in the office.

### **5.1 Analysis of potentially hazardous and harmful factors affecting service personnel in the operation of technical equipment**

In this graduation project, the development of a mobile application on Android OS is considered. The application will be developed in the office, in connection with which it is necessary to take into account the issues of creating optimal development conditions, well-being, safety and health preservation.

There are several harmful factors affecting employees working on personal computers (PCs):

- a) unsatisfactory microclimate of premises;
- b) psychoemotional stress.

Without strict consideration of safety rules and industrial sanitation, inaccurate compliance with safety requirements can lead to an accident, or to occupational diseases and industrial injuries. Labor protection is ensured by a system of legislative acts, socio-economic, organizational, technical, hygienic and therapeutic and preventive measures and means aimed at creating working conditions in which the working and harmful factors of production are excluded. Creating the most favorable, comfortable working conditions, improving occupational safety and health, no doubt leads to higher labor productivity, social development and welfare.

According to GOST 12.1.005-88. SSBT. Optimal and permissible standards of the microclimate, depending on the category of work, the work of people in the room refers to the work of light weight (1a), since the equipment is controlled by computers. The office is located in the administrative building of The Kazakh State Circus, directly close to the stables and pens for animals, so the issue of ventilation is extremely important.

With the purpose of creating normal conditions for workers of communication enterprises, the norms of the industrial microclimate are established. In the premises when working with a computer, the following climatic conditions must be observed:

#### 1) Cold season:

- a) optimum temperature 22-24 C°, permissible temperature 18-26 C°;
- b) relative humidity 40-60%, permissible humidity 75%;
- c) air speed relative and permissible 0,1 m / s.

#### 2) Warm season:

- a) Optimum temperature 23-25 C°, permissible temperature 20-30 C°;
- b) Relative humidity 40-60%, permissible humidity 55%;
- c) speed of air movement relative to 0.1 m / s and permissible 0.1-0.2 m / s.

## 5.2 The room where the project is developed

Consider the room in which the development is carried out:

- a) type of premises: office;
- b) dimensions of the working area: length 4 m, width 3 m, height 3 m;
- c) glazing of the room - double (one window measuring 2000x2000 mm);
- d) artificial lighting - lamps: 2 lamps, each with 2 fluorescent lamps (PVLМ-1x40);
- e) interior walls - light;
- f) The room for visual conditions of work refers to the V category, since the smallest object of discrimination
- g) n is from 1 to 5 mm;
- h) type of work - developing an online application for placing free ads;
- i) number of jobs - 2;
- j) the category of work is light.

The layout of the room is shown in Figure 5.1.

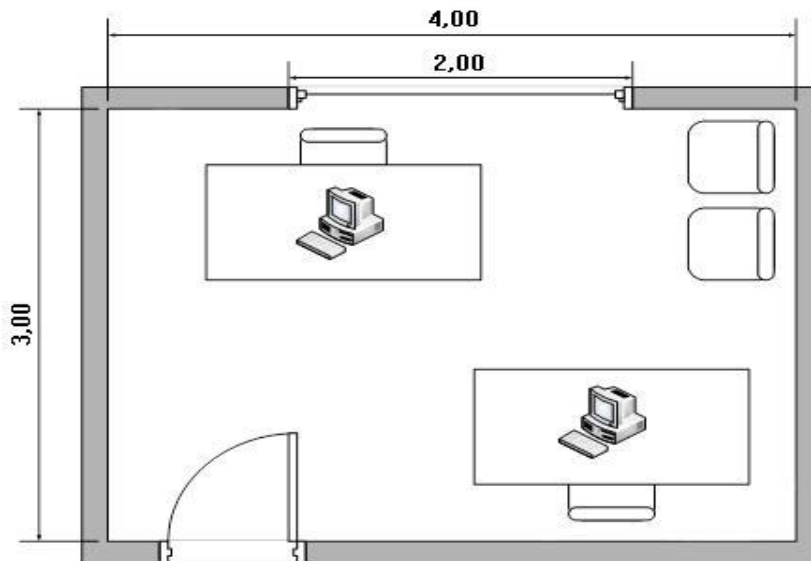


Figure 5.1 – Floor plan

## 5.3 Characteristics of the equipment used

Personal Computer. Technical characteristics of the device:

- a) Dimensions: 1600x700x1050mm (laptop + table + armchair);
  - b) power supply: alternating voltage 220-250 V, frequency 50 Hz, power 90 W;
  - c) Quantity - 1 piece.
- Router:

- a) 4-port commutator 10/100 Mbps;
  - b) Quantity - 1 piece;
- Installed 1 window conditioner Samsung SMG4590, features:
- a) cooling mode - 2400W;
  - b) heating mode - 2400W;
  - c) flow - 404 m<sup>3</sup> / h;
  - d) noise level of 31 dB;
  - e) dimensions - 754x272x176 mm;
  - f) weight - 39 kg;
  - g) It is designed for a room of up to 21 m<sup>2</sup>.

## 5.4 Ventilation of the room

Here is the calculation of the air conditioning system in the room.

Air conditioning provides the best microclimate in the room and the operating conditions of precise and sensitive equipment, and should be performed in accordance with the chapter [20].

The norm of the room space per person is 6 m<sup>2</sup>. The office assumes the availability of air ducts, which are located behind suspended ceilings. Air is smoothly fed and removed from the office either through ventilation grilles in the walls, or through special sockets installed on the ceiling. There is also an air conditioner that refreshes the air.

$V_{vent}$  - the volume of air required for the exchange;  $V_{pom}$  - the volume of the working room.

For calculation, we take the following dimensions of the working room (Figure 5.3)

Length A = 4 m;

Width B = 3 m;

Height H = 3 m.

Accordingly, the volume of the room is calculated by formula (5.1)

$$V_{pom} = A * B * H = 36 \text{ m}^3 \quad (5.1)$$

The volume of air required for the exchange is determined on the basis of the heat balance equation in formula (5.2):

$$V_{vent} * C (\text{tuphod} - \text{tapproach}) * Y = 3600 - Q_{release} \quad (5.2)$$

where Q is the excess heat (W);

C = 1000 – specific thermal conductivity of air (J / kg \* K);

Y = 1.2 – air density (mg / cm).



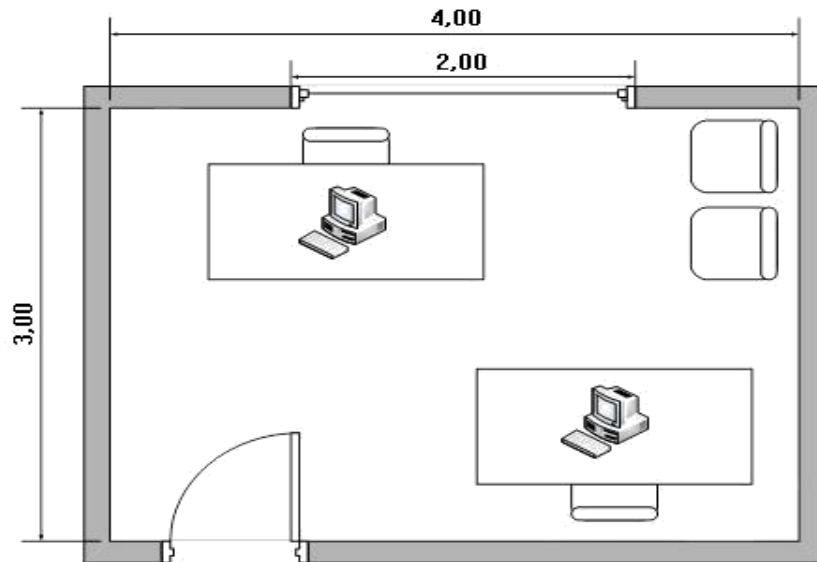


Figure 5.3 – Workplace plan

The temperature of the outgoing air is determined by formula (5.3)

$$\text{Departure} = t_{m.m.} + (H-2) t \quad (5.3)$$

where  $t = 1-5$  degrees - exceeding  $t$  by 1 m of the room height;  $t_p = 25$  degrees - the temperature in the workplace;

$H = 3$  m - height of the room;  $t_{\text{approach}} = 18$  degrees.

$$\text{Departure} = 25 + (3 - 2) 2 = 27$$

$$Q_{\text{sit}} = Q_{\text{us1}} + Q_{\text{izb2}} + Q_{\text{izb3}} \quad (5.4)$$

where  $Q$  is. - excess heat from electrical equipment and lighting.

$$Q_{\text{us1}} = E * p \quad (5.5)$$

where  $E$  is the energy loss factor for the heat sink ( $E = 0.55$  for lighting);

$P$  is the power.

$$P = 60 \text{ W} * 5 = 300 \text{ W}. \quad (5.6)$$

$$Q_{\text{us1}} = 0.55 * 300 = 165 \text{ W}$$

$Q_{\text{izb2}}$  - heat input from solar radiation,

$$Q_{us2} = m * S * k * Q_c \quad (5.7)$$

where m is the number of windows, we take m = 1;  
S - area of the window,

$$S = 2 * 2 = 4 \text{ m}^2; \quad (5.8)$$

K is the coefficient that takes into account the glazing. For double glazing k = 0.6;

$Q_c = 127 \text{ W / m}$  - heat input from the windows.

$$Q_{in2} = 4 * 1 * 0.6 * 127 = 304.8 \text{ W} \quad (5.9)$$

Q<sub>izb.3</sub> - heat dissipation of people

$Q_{iz3} = n * q$ , where

$q = 80 \text{ W / person}$ , n - number of people, for example, n = 3

$$Q_3 = 3 * 80 = 240 \text{ W} \quad (5.10)$$

$$Q_s = 165 + 304.8 + 240 = 709.8 \text{ W} \quad (5.11)$$

From the heat balance equation follows

$$V_{ent} = 3600 * 709.8 / (1000 * (27-18)) = 283.92 \text{ m}^3 \quad (5.12)$$

You need to think carefully about the location of the air conditioner in the office. It is possible to install the channel air conditioner behind the suspended ceiling and to divert air to different points of the room through air ducts. This will ensure an even distribution of air and temperature. If the height of the false ceilings does not allow the installation of a channel conditioner (as in this case), two or even three indoor units located at different points in the room can be provided. This option is especially justified in rooms of irregular or elongated shape. Semi-industrial air conditioners can connect up to three indoor units of different types to one outdoor unit. This will reduce the cost of the entire system and keep the building wall from a multitude of blocks.

## 5.5 Conclusion

In this section, an analysis was made of the working conditions in this office premises, located in the administrative block of the enterprise, in particular, the analysis of ventilation.

We have calculated all the necessary parameters for air conditioning in the room, i.e. automatic maintenance of its condition in the room in accordance with certain requirements, regardless of changes in the state of outdoor air and the conditions in the room itself, since the office is located in close proximity to animal pens and the need for a well-designed ventilation system is high.

## Conclusion

Software development is an interesting and complex task, the fulfillment of which can be more than justifying itself. It requires logical thinking and a structural approach. It includes many tasks that are not directly related to the study of code languages. It always has new knowledge that can be purchased, new requirements that require research and new technologies that are worth experimenting with. The award is the kind of working final version of the application and the awareness of its contribution to achieving this goal. The software defines the shape of the modern world, and those who work on its creation can contribute to the image of the future.

During the implementation of this diploma project, the basic principles of developing mobile applications, as well as the requirements for them, were studied. The principles of construction and operation of the PP for the Android operating system have been studied. In addition, it was also found that the topic under consideration is the most relevant in the realities of the modern mobile devices market.

Also, an application was developed for the corresponding platform, which, if desired, can be transferred to other mobile operating systems using the architecture of the ARM processor. This application meets all standards and requirements of the holder's platforms.

The total cost of the product was calculated, and based on the summation of the various types of costs that are possible in production. This calculation corresponds to any type of product realization, both for development for the customer, and for independent development, for subsequent publication in the online application store. In the process of researching similar products on the market, it is established that the cost of development fully corresponds to the quality of the developed PP and allows to obtain a large amount of profit.

In the section "Safety of life", the necessary conditions for the organization of a workplace corresponding to all established norms were calculated for the employees involved in the project implementation. Calculated the necessary ventilation, required for comfortable work without harming the health of employees.

## List of literature

- 1 Freshgamer. URL: <https://www.youtube.com/user/freshgamer10/videos>
- 2 How to start developing an android app. URL: <http://startandroid.ru/en/>
- 3 Build for Android. URL: <http://developer.android.com/intl/ru/index.html>
- 4 Android book. URL: <https://metanit.com/java/android/>
- 5 Badagiev B.T. Documentation on labor protection in the organization. - Moscow: Alfa-press, 2010.
- 6 Reto M., Android 2. Programming applications for tablet computers and smartphones. - Moscow: Eksmo, 2011.
- 7 Dusebaev M.K., Begimbetova A.S. Methodical instructions for diploma projects (for students of all forms of training specialties 050719 - Radio engineering electronics and telecommunications, 050704 – Computer engineering and software). - Almaty: AIPET, 2008.
- 8 BNaR of The Republic of Kazakhstan 2.04-05-2002 "Natural and artificial lighting. General requirements. - Astana, 2002.
- 9 Korobko, V.I. Labor protection: A manual for university students /I. V. Korobko. - Moscow: Unity-Dana, 2013. - 239 p.
- 10 Abdimuratov Zh.S., Mananbaeva S.E. Life safety. Methodical instructions for the execution of the section "Calculation of ventilation" in the final works for all specialties. Undergraduate. - Almaty: AIPET, 2009.
- 11 Abdimuratov Zh. S., Mamanbaeva S. Ye. Calculation of the production room. - Almaty: AUCP, 2009.
- 12 Aidarkhanova M., Fundamentals of economic theory. - Moscow: Tome, 2010.
- 13 Makhotina M.V., Simonenko V.I. Economics in the schemes: Textbook. - Moscow: Eksmo, 2011.
- 14 Nosova S.S., The economic theory. - Moscow: Knorus, 2010.
- 15 Artamonova V.S., Ivanova S.A. Economic theory. - St. Petersburg: Peter, 2010.
- 16 Kamaeva V.D., Lobacheva E.I. Economic theory: Textbook. -Moscow: Yurayt, 2010.
- 17 Graduate work of bachelors. Economic section: manuals / Ed. Babich AA, Kazyken B.B., Sagira A.A. - Almaty: AUPET, 2009.