

Magnolia TILCA  
"Vasile Goldiș" Western University of Arad,  
Faculty of Economy Science, Engineering and Computer Science

# MOBILE COMPUTING USING ANDROID WITH AN EMPHASIZES ON ECONOMIC APPLICATION

Theoretical  
Article

---

## Keywords

*Android Studio*

*Economic apps*

*Java language*

*Linear regression*

---

## JEL Classification

L86

---

## Abstract

*As result of the convergence of computers and mobile phones, Android comes with the great opportunity of implementing personal mobile applications in a Java language environment. The paper presents two main aspects concerning the new age of programming. The first aspect refers to the existing library of applications in economic domain. The paper extracts the most useful financial and business apps that run on Android smart-phones. The second aspect presents an example of building an economic application using one of Android's various IDEs (Integrated Development Environments): Android Studio. The developed app calculates the prognosis of a business size starting from known statistic data sets and evaluates the effort of the business implementation, using the linear regression method. The app differs from other Android applications by the explicit answers that result from the regression calculus.*

## INTRODUCTION

The great impact of Android is well known in the mobile world: with “more than 11,868 distinct Android devices on the market” (Wagner, 2014) and “over one million Android applications (apps) published” (Wikipedia, 2015).

The generic term *Andorid devices* refers to the Andorid operating system, smartphones (produced by Samsung, Asus, LG, Sony etc.), tablets, multimedia devices as GPS, all of them running under the Andorid operating system (OS).

*Andorid applications* (apps) are mobile software programs necessary for the Andorid OS to run properly on the desired device, in order to satisfy particular demands of the user. Beside those ready-to-use applications – available on the Google Play store – users can build easy Android apps for their business, budget, blog, product or service using “do-it-yourself” creation platforms with no coding required, such as AppMaker, Appy Pie, Appery, BiznessApps and so on. For higher app performance, it is necessary to use intelligent code editor environments. Android Studio is one of the most used integrated development environments for developing on the Android platform.

The present paper investigates the mobile applications developed with the Android Operating System (OS). First some useful calculator-apps from finance and business area are presented. These calculator-apps may be downloaded from the Google Play Store Internet platform and used as calculators. Second, the paper describes the necessary steps in implementing a calculator-app using the Andorid Studio programming environment: the creation of the project, the designing of the layouts, the programming of the app. A calculator-app is also creating. The app provides a prognosis value and the degree of correlation between economic-social phenomena, via linear regression theory. In this regard, the paper presents the useful notions about linear regression method. The app addresses to any kind of user who wants to investigate the business domain.

## LITERATURE REVIEW

### ANDROID AS AN OPERATING SYSTEM

Android is the world's most popular operating system for mobile devices and tablets. It is an open source operating system, created by Google, and available to all kinds of developers with various expertise levels. From the first version (2007), up to the latest one (2015 - 5.1. Lollipop), the Android operating system comes with a lot of improvements. Some of them are: application framework enabling reuse and replacement of

components, Dalvik virtual machine optimized for mobile devices, integrated browser based on the open source WebKit engine, optimized 2D, 3D graphics, SQLite for structured data storage, media support for common audio and video, support for GSM Telephony, support for Bluetooth, EDGE, 3G, 4G and Wi-Fi, support for Camera, GPS, rich development environment including a device emulator, tools for debugging, memory and performance profiling and so on (Chandnami, et al., 2012).

The major feature of Android is the opportunity to improve a smart-phone's capabilities by downloading some great – free or paid-for – applications. There are more than one and a half million Android apps on Google Play (AppBrain/Stats, 2015), covering 45 domains (Google, 2015). A great interest is shown in the economic categories, Business and Finance, with more than 130 thousand existing apps (AppBrain/Stats, 2015).

The smartphone of a business person needs applications for managing documents similar to a PC. The business category comes with solutions such as: office suits (WPS Office+PDF, Polaris Office+PDF, OfficeSuite 8+PDF Converter etc.), file and folder management (File Commander-File Manager etc.), print or scan files (PrinterShare, PrintHand Mobile Print, Fast Scanner: Free PDF Scan etc.), messenger, chat and meetings options, communication with clients (Slack, BBM Meetings, Line@App etc.), business tracking and analytic views (Square Register) and so on.

A useful class of apps are the *calculator-apps* for quick economic computations, such as a percentage calculator (useful in shopping, discounts), investment return rate, loan interest rate, stock exchange quotes, markup, profit margin, company profit changes, taxes, etc.

Google's Android operating system has a large collection of personal finance calculator-apps. When you need to make a decision on important personal business matters you could simply use **Business Viability Calculator** or **Business Loan Calculator**; or if you want to find the payments of a mortgage loan and amortization values you can benefit from the quick calculations of **Mortgage Calculator Free** and so on. The list from Tabel 1 describes some of the most common business and finance calculators developed for smart-phones using the Android OS.

These calculator-apps underline the main aspects of business/finance. Many other apps-calculator oscillate around the same themes.

But how are these applications constructed? Which are the mobile development environments and the programming languages that Android applications can be implemented?

The next section is an introduction to Android mobile programming: after the description of the main characteristics of the Android Studio – a software platform for developing Android apps – the paper presents the detailed steps of implementing an app-calculator using the Android Studio software.

## METHODOLOGY

### ANDROID STUDIO – A DEVELOPMENT ENVIRONMENT

The demand for app developers remains high through 2016. The developers have to acquire skills in mobile computing which means to choose the *mobile platform* they will develop for, then to learn the *programming languages* and software *development environment* of the platform.

Google's Android is one of the most popular *mobile platforms* for creating apps, having a powerful development framework, as well as an open marketplace for distributing them to users instantly.

Android's platform provides tools with advanced features for developing, debugging and packaging apps. This is about Android Studio, the official Android integrated *development environment* (IDE). Android Studio consists of

- *a source code editor*: “capable of advanced code completion, refactoring, and code analysis” (Android Developer);
- build automation tools* represents the toolkit used to build, test, run and package the apps. The Android build system contains a project wizards (Code Templates and GitHub Integration), an Android project viewer and module support (Multi-screen app Development), a virtual device manager for all shapes and sizes for common Android devices (Android Virtual Device Manager), an opportunity to create multiple apk for an Android app with different features using the same project (Android Builds Evolved, with Gradle) so on. Here, *apk* (Android application package) “is the package file format used to distribute and install application software onto Google's Android OS.” (Wikipedia, 2015);

- *a debugger* called Android Debug Monitor tests and debugs the target program.

The programs are written inside of a mobile development environment using the *Java programming language*.

Developing with Android Studio is the perfect method because it can directly invoke the needed tools while developing applications.

Android Studio and the Java language are not the only options for Android App development. There are plenty of alternatives to Google's tools for Android app development IDE and computer language, some of them mentioned below:

- for developing websites directly on an Android device, using HTML/CSS/JavaScript languages, it is suitable to use the Android web editor *AIDE Web* (<http://www.android-ide.com/>)
- for developing real Android apps directly on the Android device, using C/C++/Java languages, use *AIDE* (<http://www.android-ide.com/>)
- for easy development of mobile and desktop applications with the web based <http://www.applicationcraft.com> under the HTML5 language, use the drag-and-drop IDE *Application Craft*
- for quick development of any type of Android apps in modern version of Visual Basic, use *Basic4Android* (<http://www.b4x.com/b4a.html>)
- for a complete solution for developing connected apps for Android (or Windows, iOS) from a single codebase in ObjectPascal/C++ languages, use *RAD Studio XE* (<http://www.embarcadero.com>)

This list shows that different languages can be considered when developing apps. For a complete list of free and commercial IDEs for Android, please check Fowler (Fowler, 2014).

## MAIN FINDINGS

The development process for Android applications with Android Studio consists of four main steps (Android Developer): 1. environment setup, 2. project development, 3. debugging, 4. testing and publishing.

An economic application will be implemented following first three steps (the “publishing” step is not included in this paper). After researching through economic calculator-apps we have noticed an important aspect that is missing from almost all Android economic apps: the economic interpretation of generated results. The interest of this paper is concentrated on the econometric domain, where interpreting the results is very important. Taking the simplest econometric aspect, the linear regression, what is the interpretation of its linear equation? What does positivity of the beta coefficient mean? What is the significance of the beta parameter testing? What happens if we choose another statistical sample or the entire population? The answers to these questions are well known by specialists in econometric science, but what about other people involved in business field? How do they interpret the regression results starting at some firm data input, without knowing the entire econometric notions?

Starting from these questions, the Android application developed in this paper is a practical calculator for prognosis of some scale starting from some well known scales and using the linear regression model. The app also evaluates the

validation of the model and its results. All the computed values are explicitly described in phrases that help users make future decisions at business level. For example, the app tests the relation between the returns of the sales in five months and the number of clients that entered in the store in that period.

1. To *setup the development environment* means to install the Android SDK (Software Development Kit – a programming package that enables a programmer to develop apps), the Android Development Tools and the Android platforms. Android Studio for Windows runs under Windows 2003/Vista/7/8, needs 2 GB RAM minimum, 400 MB hard disk space and another 1 GB for the Android SDK. Before installing Android Studio, it is necessary to install the Java Development Kit (JDK) 7. The detailed steps of setup can be followed from the URL address <http://developer.android.com/sdk/index.html>.

The next step is to create Android Virtual Devices and connect hardware smart-phone.

2. The main phase of developing applications is the *Project development* step. Now the Android project can be create using source code, resources files and graphical tools. The project will contain one or more different types of modules, such as application modules (the container for the app's source code), library modules (the shareable Android source code) and test modules (the code for testing the project). The detailed steps of creating an Android project will be described in what follows.

3. The next step is *building and running the application* in debug mode, using the Android Debug Monitor, and testing the application using Android SDK testing tools.

4. The last phase is *publishing the application* that means configuring, building and testing the application in release mode, and publishing, selling and distributing it to users.

The first step in *Project development* is to design the interface of the app and its structure. The interface must be simple, clear and efficient (see Figure 7 a). The next two interfaces (Figure 7 b and Figure 7 c) provide the input data of the application: the values of input data and a description of variables. The navigation through those interfaces can be done using the forward and backward arrows. Further, the app shows some results. The next interface (Figure 8 a) asks the user to choose one of the tasks to find the answer of the task. The first task ensures a predicted value for the dependent variable  $y$  associated with a given value of  $x$  (Figure 8 b). The description of the variables  $x$ ,  $y$  are taken by the algorithm from the data indicated in the Figure 7 c. The forward arrow displays the next interface with the predicted value (Figure 8 c). The algorithm is based on the well known results from the regression theory: finding the parameters

$\alpha, \beta$  of the linear regression equation  $= \alpha + \beta \cdot x$   
 $\alpha, \beta \in \mathbb{R}$ , using the formula

$$(1) \quad \beta = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

$$(2) \quad \alpha = \bar{y} - \beta \bar{x},$$

where  $\bar{x}, \bar{y}$  are the mean values of the given variables  $x_i, y_i \in \mathbb{R}$ .

Going back to the task interface, the user may want to find a prediction of the other independent variable,  $x$ , associated with a known value  $y$ . The interfaces are in the same manner as above.

The next step is to investigate the extent to which the prognoses are achievable (Figure 8 c). With a probability of 95%, the predicted value  $y$  lies in the interval  $(\alpha + \beta x_p - t_a \cdot \sigma_\varepsilon^2, \alpha + \beta x_p + t_a \cdot \sigma_\varepsilon^2)$ , where  $\sigma_\varepsilon^2 = \frac{1}{n-2} \sqrt{\sum_{i=1}^n \varepsilon_i^2}$  is the standard error of the estimation. Here,  $t_a$  is the student's  $t$  distribution with 0.05 level of significance and  $n-2$  degree of freedom. We obtain similar answer in the case of  $x$  variable prognosis, with the mention that the predicted interval is given by  $((y_p - \alpha - t_a \cdot \sigma_\varepsilon^2)/\beta, (y_p - \alpha + t_a \cdot \sigma_\varepsilon^2)/\beta)$ .

The final question is the most important in prediction: what happens if we choose different sample from the entire population? Do the values change? Is the chosen sample well identified? Is the predicted value still valid, whatever the chosen sample is? Is it indicated to go further with the business investigation? The answer is given by the last task from the list Figure 7 a: Is it possible the effort to be covered within a convenient time? (See the designed interface with the validity of the results: Figure 1). The effort of obtaining the input data for the sample, of processing the data input using the linear regression is explained in percentage  $R^2 \cdot 100\%$ , where the coefficient of determination  $R^2 = 1 - \sum_{i=1}^n \varepsilon_i^2 / \sum_{i=1}^n (y_i - \bar{y})^2$  is very close to 1. The percentage also shows how much the variation of  $y$  is explained by the influence of  $x$ .

The evolution of variables  $x$  and  $y$  is given by the correlation aspect. The correlation may be positive, strongly negative correlation or negative correlation, depending on the correlation coefficient

$$R_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}. \text{ If } R_{xy} \text{ is too small,}$$

there is a low connection between  $y$  and  $x$ , if  $R_{xy} \rightarrow 1$  both variable  $x$ ,  $y$  vary in the same direction and if  $R_{xy} \rightarrow -1$ , the variables vary in opposite. There is no correlation if  $R_{xy} \rightarrow 0$ .

The significance property of the model tests if there is a linear connection between  $x$  and  $y$ . As long as the parameter  $\beta$  from the regression equation is not zero, whatever the sample is chose, the equation is linear. Because we cannot verify all the samples of the population, or the entire population, the inequality  $t_a < |\beta / \sigma_\beta^2|$  establishes that  $\beta$  is not zero in every possible choice of the sample. Here

$\sigma_\beta^2 = \sigma_\varepsilon^2 / \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}$  is the standard error of  $\beta$ . Moreover, the *Details* button (from Figure 1) displays the equation of regression model and the confidence intervals of the parameters  $\alpha, \beta$ :

$$\alpha \in (\alpha - t_\alpha \sigma_\alpha^2, \alpha + t_\alpha \sigma_\alpha^2), \beta \in (\beta - t_\beta \sigma_\beta^2, \beta + t_\beta \sigma_\beta^2)$$

(see the designed interface of the significance of the model, Figure 2). Here  $\sigma_\alpha^2 = \sigma_\varepsilon^2 \cdot \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$  is the standard error of  $\alpha$ . If the model is not significant, the case of  $|\beta/\sigma_\beta^2| \leq t_\alpha$ , the interfaces are as follow in Figure 3 and Figure 4.

All these formula are implemented in the computation algorithm of the app using the Java language environment.

1. The creation of the project. When you create a new project (*File* → *New Project* pane, with *Blank Activity* and with the name *LinearRegressionCalculation* in our case), the project is a minimal example application project that displays the words: "Hello World!" (An *activity* is "a window within your App that displays content the user can interact with") (Buyliss, 2014);

2. Designing the layouts. All the layouts of the app were created by:

- changing the text in the file *activity\_main.xml*, which is located under *app* → *res* → *layout* → *activity\_main.xml* in the project file hierarchy and

- creating new files using *Alt+Insert* → *Activity* → *Blank Activity* (*activity\_ecran\_secundar.xml* etc.). The *Palette menu* provides a "what you see is what you get" (WYSIWYG) environment from where the user may select a tool (layout template, widget, text field, containers, date and time fetures, expert or custom properties) and place it onto the app. The *xml* resource file is ready to be modified by clicking the *Text* button. (see the Android Studio window from Figure 5)

The contact to the next layout is done by the "ImageButton" from the Widgets list. The code that links the button with the next layout of the app is

```
android:id="@q+id/imageButtonEcran2"
```

Open the *strings.xml* file (*app* → *res* → *values* → *strings.xml*) to give a title for the screen (Figure 6).

3. Running the app. To run the app on the phone, the phone must be plugged into the computer using a USB data cable. Android Studio installs the *LinearRegressionCalculation* app on it by clicking on *Run* button (on the Android Studio Toolbar) and choosing the device. Another way to run the app is to use the Android Emulator. For example, the virtual device Nexus 5 API 22.

4. Programming the app. The app has three main screens for introducing the input data (Figure 7) and one screen for choosing the desired task in the regression calculation (Figure 7 a.). If someone chooses to find a prognosis of a scale  $y$ , the app asks for one more value (Figure 7 b.). This screen already constructs a phrase for better understanding

of the necessary input data in the regression theory. The answer may be seen by clicking next button (Figure 7 c.).

The Java script for the predicted value of a scale  $y$  is exemplified below. The script is located in *app* → *java* → *com.example.mm.linearregressioncalculation* → *Ecran6.java* file. The array type variable *xValues* contains the known independent values  $x_i, i=1, \dots, n$  and the string type variable *textX* contains the word which identifies what  $x_i$  are.

These input variables are used in the *protected void onCreate* subroutine (*Bundle savedInstanceState*) (Appendix A-B). First, the variables are populated with the given correspondent values that were declared in the previous interfaces (Java language uses the function *getIntent()* which returns the intent that started this activity). After that, the algorithm computes the values of the parameters  $\alpha, \beta$  by the formulas given in (1) - (2). (Appendix D).

A Java language characteristic is the summation of the values  $y_i$  for each  $i=1, \dots, n$  made by the combining the symbols  $+=$  and specifying the type of the result (*Double*) (Appendix C). The final formula (1)-(2) are expressed in Java language as it is shown in Appendix D.

The big challenge during the programming of this app was to create the interface of reading input data (Figure 7 b.). This screen offers users the possibility of deleting the wrong value and the possibility of adding more lines for complete the variables  $x$  and  $y$ . For this purpose a *GridViewAdaptor* java file was created. It manages the lines from the grid.

These are only few aspects that show how the Android Studio works. The app may be further improved and uploaded on Google Play site in order to be used by people interested in business values prediction.

## CONCLUSION

The paper presents two main aspects concerning the new age of programming – the mobile Android programming. The first aspect refers to the existing library of applications in economic domain. The paper extracts the most useful financial and business apps that run on Android smart-phones (see Table 1). For example when you need to make a decision on important personal business matters you could simply use *Business Viability Calculator* or *Business Loan Calculator*; or if you want to find the payments of a mortgage loan and amortization values you can benefit from the quick calculations of *Mortgage Calculator Free* and so on.

The second aspect refers to the big opportunity of programming an app using an Android applications development IDE and a computer language. The paper describes the steps in developing an

economic app with Android Studio programming environment. The code uses the Java language. The developed app calculates the prognosis of a business size starting from known statistic data sets and evaluates the effort of the business implementation, using the linear regression method. The app differs from other Android applications by the explicit answers that result from the regression calculus (see Figure 8 c.).

The objective of the paper was to underline the property of the Android System Operator: to develop (or to improve) any kind of app (games, books, restaurant's menus, news, calculations and various other apps). In this regard, the paper stops at an economic calculator-app because the calculator-apps are software with simple graphical interface and they are perfect tools in finding very quickly the desired economic (or math) results. The *LinearRegressionCalculation* app shows how to create such an app. In this regard, the paper explains first the econometrical formulas (that are needed in the linear regression process) and where those formula are used in the concrete screens of the app. The paper continues with the app's implementation in the Android Studio environment using Java Language: the creation of the project, the designing of the interfaces and the programming of the app. Programming the app was not entirely described in this paper. The app addresses to any kind of user who wants to investigate the business domain.

The future steps are to improve the design of the interfaces and to publish the app on Google Play. The paper may also be used for developing future apps.

Thus Android is a solid platform for building innovative mobile solutions which may be utilized by most of the people who have some programming notions.

With Android Studio, businesses, consumers and programmers have embraced this innovative technology. Based on the business modern rule "Connected means Informed" (Cisco, 2016), the next challenge is to implement business apps for smart phones connected together (IoT - Internet of Things – a concept for the internetworking "multitude of devices developed for smart and innovative applications") (Apatean, Dunca, 2016) in order to collect and exchange data with other intelligent devices using different programming

environments. An implementation of an intelligent system for the IoT is presented in (Apatean, Dunca, 2016).

## REFERENCE LIST

- [1] Android Developer – The Official site of Android, Android Studio. *Android Developer*. Retrieved from <http://developer.android.com/sdk/index.html>.
- [2] Apatean, A & Dunca F. (2016). An Intelligent Eye-Detection Based, Vocal E-Book Reader For The Internet Of Things. *Acta Technica Napocensis* 57.2 (2016), 9-14.
- [3] AppBrain/Stats. (2015). Android Statistics - Number of Android applications. *AppBrain/Stats*. Retrieved from <http://www.appbrain.com/stats/number-of-android-apps>.
- [4] Buyliss, D. (2014). Android Tutorial for Beginners - Part 1. *Raywenderlich Tutorials for Developers & Games*. Retrieved from <http://www.raywenderlich.com/78574/android-tutorial-for-beginners-part-1>.
- [5] Chandnami, P. & Wadhvani, R. (2012). Evolution of Android and its Impact on Mobile Application Development. *International Journal of Scientific Engineering and Technology*. 2012, 1(3), 80-85.
- [6] Cisco (2016). Internet of Things. <http://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf>
- [7] Fowler, D.(2014). List of IDEs for Android App Development, Which is Best for You? *Tek Eye - Programming Android and Computing Technology*. Retrieved from <http://tekeye.biz/2014/list-of-android-app-development-ides>.
- [8] Google Play (2015). Google Play Apps. *Google Play*. Retrieved from <https://play.google.com/store/apps>.
- [9] Wikipedia. (2015). Android application package. *Wikipedia, The Free Encyclopedia*. Retrieved from [https://en.wikipedia.org/wiki/Android\\_application\\_package](https://en.wikipedia.org/wiki/Android_application_package)

## ANNEXES

*Appendix A.* The variables declaration body for the calculation of the predicted value  $y$

```
package com.example.mm.linearregresioncalculation;

import ...

public class Ecran6 extends ActionBarActivity {
    private ArrayList<String> xValues;
    private ArrayList<String> yValues;
    private String textX;
    private String textY;
    private int optiune = 0;
    private String xp = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {...}

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {...}
}
```

*Appendix B.* The population of the variables

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ecran6);

    xValues = getIntent().getStringArrayListExtra("x-values");
    yValues = getIntent().getStringArrayListExtra("y-values");

    textX = getIntent().getStringExtra("textX");
    textY = getIntent().getStringExtra("textY");

    optiune = getIntent().getIntExtra("optiune", 0);

    xp = getIntent().getStringExtra("xp");
}
```

*Appendix C.* The calculation of the mean value for the values from  $y$  array

```
double mediaAritmeticaY = 0;
for (int i = 0; i < yValues.size(); i++)
{
    mediaAritmeticaY += Double.valueOf(yValues.get(i));
}
mediaAritmeticaY = mediaAritmeticaY/yValues.size();
```

*Appendix D.* The calculation of the parameters  $\alpha$ ,  $\beta$

```
double raportBeta = 0;
if (sumXX != 0)
{
    raportBeta = sumXY/sumXX;
}

double alpha = mediaAritmeticaY - raportBeta*mediaAritmeticaX;
double predYValues = alpha + raportBeta * Double.valueOf(xp);
```

**Tables**

Table 1  
*The list of the Andorid calculatior-apps and their featur*s

<b>Name</b>	<b>Description</b>	<b>Features</b>	<b>Useful (in economy)</b>
<b>Percentage Calculator v1</b>	Solves various problems related to percentages. <ul style="list-style-type: none"> <li>▪ simple percentage calculator;</li> <li>▪ percentage increase / decrease;</li> <li>▪ restaurant tip amount;</li> <li>▪ discount rate;</li> <li>▪ markup;</li> <li>▪ profit margin;</li> </ul> Convert fractions to percentages.	<ul style="list-style-type: none"> <li>▪ Free</li> <li>▪ 877 kb file size</li> </ul>	finances and business: <ul style="list-style-type: none"> <li>▪ investment return rate,</li> <li>▪ loan interest rate,</li> <li>▪ stock exchange quotes,</li> <li>▪ markup,</li> <li>▪ profit margin,</li> <li>▪ company profits changes,</li> <li>▪ shopping (discounts).</li> </ul>
<b>Business Calculator</b>	Calculator for business administration and economics. <ul style="list-style-type: none"> <li>▪ Gross profit and markup calculator (data input: the cost price and selling price of the product);</li> <li>▪ Price and markup calculator (data input: the cost and gross margin);</li> <li>▪ Operating margin calculator (data input: the operating income and revenue);</li> <li>▪ Effective interest rate calculator (data input: the nominal interest rate and the number of compounding periods per year);</li> <li>▪ Value added tax VAT (data input: the price without VAT and the VAT rate of the country);</li> <li>▪ Break-even point (data input: fixed cost and price of a unit).</li> </ul>	<ul style="list-style-type: none"> <li>▪ Free</li> <li>▪ 175 kb file size</li> </ul>	School: <ul style="list-style-type: none"> <li>▪ help students learn finance and economy;</li> </ul> Business administrations.
<b>Business calculator LITE</b>	Calculates profitability and efficiency. <ul style="list-style-type: none"> <li>▪ Gross measures;</li> <li>▪ Break-even point</li> <li>▪ Demond elasticity;</li> <li>▪ Discounts;</li> <li>▪ Budget (for small businesses, individuals);</li> <li>▪ Campaign prices for goods;</li> <li>▪ Business liquidity.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Free</li> <li>▪ 2552 kb file size</li> </ul>	Business and finance: <ul style="list-style-type: none"> <li>▪ profitability</li> <li>▪ efficiency.</li> </ul>
<b>Business calculations</b>	Facilities the process of economic calculation. <ul style="list-style-type: none"> <li>▪ Prediction of the payback of any business;</li> <li>▪ Breakeven point calculator (minimum sales amount below which the business becomes unprofitable);</li> <li>▪ Build a detailed financial plan for business development.</li> </ul>	<ul style="list-style-type: none"> <li>▪ \$3.36</li> <li>▪ 321 kb file size</li> </ul>	Specifically for entrepreneurs.
<b>Business Loan Calculator</b>	Helps to calculate monthly loan payments for fixed/rate loans (data input: the loan amount, interest rate and loan term).	<ul style="list-style-type: none"> <li>▪ Free</li> <li>▪ 991 kb file size</li> </ul>	Finance
<b>Business Viability Calculator</b>	Assesses the viability of new business ideas or business expansion plans. The app includes: <ul style="list-style-type: none"> <li>▪ Earning before interest tax and depreciation;</li> <li>▪ Net cashflow;</li> <li>▪ Internal rate of return;</li> <li>▪ Indicative valuation of the idea;</li> <li>▪ Business viability report.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Free</li> <li>▪ 10650 kb file size</li> </ul>	Business
<b>Mortgage Calculator Free</b>	Calculates total interest and total payments over the life of the mortgage loan. The app includes: <ul style="list-style-type: none"> <li>▪ Amortization graph</li> </ul>	<ul style="list-style-type: none"> <li>▪ Free</li> <li>▪ 172 kb file size</li> </ul>	Finance
<b>Finance-Investment Calculator</b>	Focuses on 100 useful and important finance and investment formulae: <ul style="list-style-type: none"> <li>▪ Basic finance formulae;</li> <li>▪ Depreciation and cash flow;</li> <li>▪ Corporate finance formulae;</li> </ul>	<ul style="list-style-type: none"> <li>▪ \$1.99</li> <li>▪ 4590 kb file size</li> </ul>	Finance and Investment field.

Name	Description	Features	Useful (in economy)
	<ul style="list-style-type: none"> <li>▪ Stocks and bonds;</li> <li>▪ Currency convertor.</li> </ul>		

Source: Author's view using the data from the AppBrain Android market (<http://www.appbrain.com/>)  
The calculator-apps underline the main aspects of business/finance. Many other apps-calculator oscillate around the same themes.

**Figures**

 **Answer:**  

*The results are valid in ...%, whatever of chosen sample.*

*Yes! The model is significant:*

- ✓ *the input data are well identified*
- ✓ *the input data are strongly positive correlated (a growth with one ... generates a grows with ... ..).*

Details  


Figure 1. The validity of the results: positive case

 **Answer:**  

*There is a linear connection between the data input x, y*

$y = \dots + \dots x$

*Moreover, ... ∈ (... , ...) and ... ∈ (... , ...), whatever the chosen sample you had made*

*\*the confidence percentage is 95%*



Figure 2. The significance of the model: positive case

 **Answer:**  

*The results are valid in ...%.*

*No! The model is not significant:*

- ✓ *the input data are bad identified.*

*Please make another data input statistical identification.*

Details  


Figure 3. The validity of the results: negative case

 **Answer:**  

*There is no linear connection between the data input x, y*

$y = \dots$

*In this case, the best prediction is given by the mean model of the independent variable y.*



Figure 4. The significance of the model: negative case

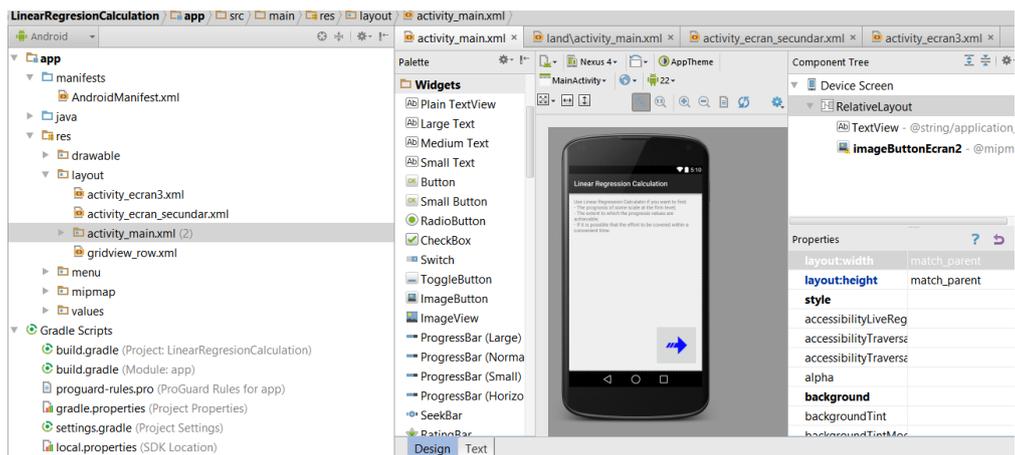


Figure 5. The Android Studio window. It contains three parts: the folder structure (left), a preview of the layout (middle) and the layout hierarchy and its attributes (right)

```
<string name="title_activity_ecran4">Find</string>
<string name="text1Ecran5">Write a value {x} (x) to find out the predicted value for {y} (y).</string>
```

Figure 6. The code for the title of the layout Ecran 4

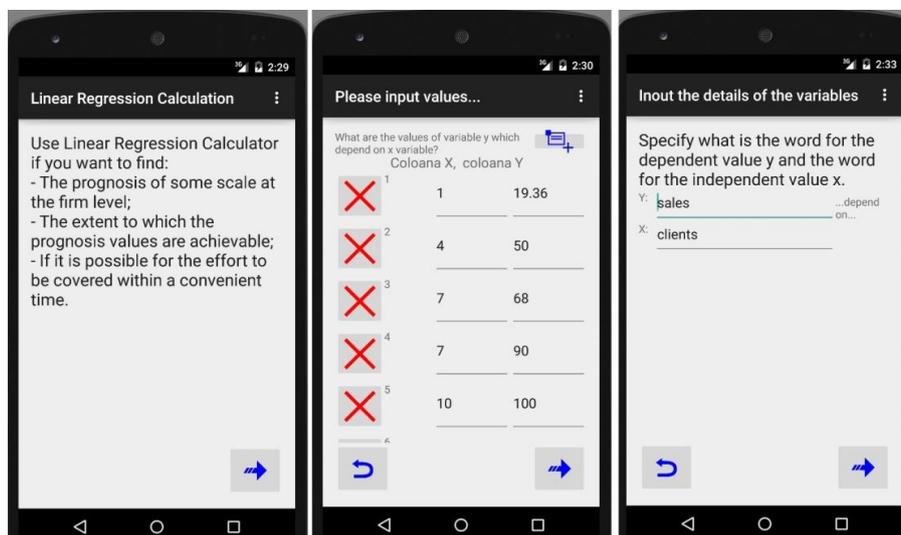
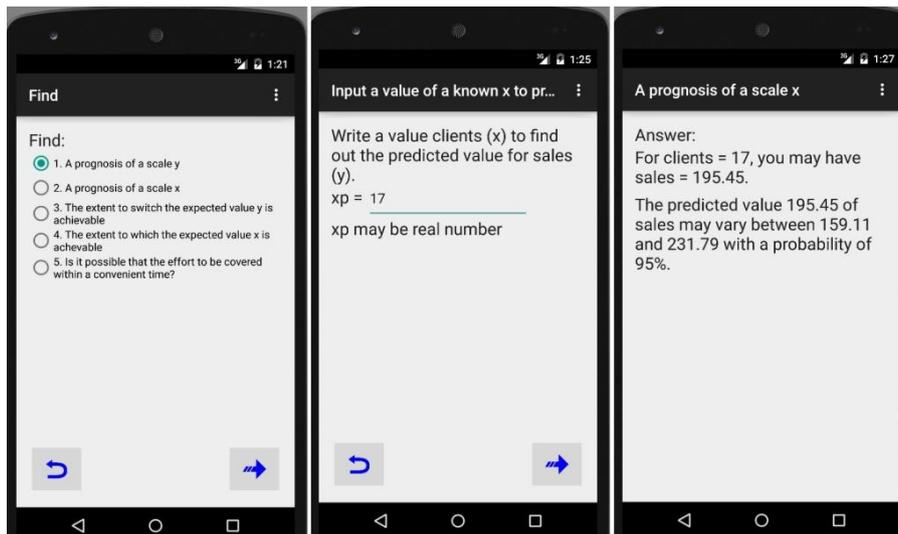


Figure 7. The first three screens ask for the input data in order to establish the linear relation between the number of clients  $x$  (thousand) and the values of sales  $y$  (thousand RON); a) The introducing interface; b) The entering of input data interface; c) The entering of details



*Figure 8.* The next three screens compute a prognosis value for *sales* (thousand RON) for a given number of *clients* (thousand): a) The interface with the tasks that the app can do; b) The interface asks the value of a known *x*; c) The predicted value *y* and the extent to which the predicted value *y* is achievable