

# **Organization of the business process of a real estate agency, using an information system supported by a regression model**

**Egor Gulakov**

Work carried out under the supervision of

**Prof. Paulo Jorge Matos**

**Prof. Marina Rasskazova**

**Master in Information Systems**

**2017**



**Organization of the business process of a real estate agency, using an information system supported by a regression model**

Dissertation for the title of  
Master in Information Systems  
High School of Technology and Management

Egor Gulakov

June 2017

The High School of Technology and Management is not responsible for the opinions expressed in this report.

# Acknowledgments

I would like to express my deep gratitude to the ICM Erasmus + International Credit Mobility for the opportunity to study in one of the best European institutions. This program allowed me to gain invaluable knowledge from highly qualified professors, improve my English and find new friends from all over the world. Separately, I would like to thank my supervisor - Professor Paulo Jorge Matos, for his patience and tremendous help in writing my master's thesis. Thank supervisor of the home institute - Marina Rasskazova for good ideas. Also thanks to my parents - Olga Gulakova and Konstantin Moroz, for investing in me time, love and care, brought up in me a person. I love you and will try to live up to your expectations! I thank my friend Vahan Vahanyan and neighbour Zhenya Selivonchik for their support and friendly motivation for writing the dissertation



# Abstract

The effectiveness of the functioning of the real estate market depends on the level of development of the management structure of enterprises. It is necessary to create an information and analytical base that meets the requirements and wishes of the client. This product should have functions that simplify the process of searching and selling real estate, and also have a positive effect in the activities of the real estate agency. [B 15]

We created an information system of the real estate agency, which is aimed at satisfying the needs of the client and increasing the comfort when using the product.

The first step was to conduct an analytical study of the real estate agency "Galleon", the main organizational charts were reflected, the agency's activities were described in detail. Next, it was necessary to make demands on the future information system. On UML - Diagrams, the functions and processes of the future program were described in detail, the roles of the main actors were characterized. All the diagrams were built in the program "IBM Rational Rose".

The next step was the development and construction of a mathematical model for calculating the cost of the apartment. At this stage, we used the package for regression analysis included in the MS Office Excel.

The final stage was the development of the main product - the information system of the real estate agency. The program was created using Visual Studio 2017, on the ASP.NET platform. The C # programming language was used. The information system database was implemented using markup language – XML.

**Keywords:** Real estate agency, informational system, UML diagram, XML, C# language





# Index

<b>Acknowledgments</b> .....	<b>v</b>
<b>Abstract</b> .....	<b>vii</b>
<b>Index</b> .....	<b>ix</b>
<b>List of Figures</b> .....	<b>xii</b>
<b>List of Abbreviations</b> .....	<b>xiii</b>
<b>Introduction</b> .....	<b>1</b>
1.1. Goals .....	1
1.2. Structure of the document .....	2
<b>Inspection of the Galleon agency</b> .....	<b>3</b>
2.1. Description of activity of real estate agency Galleon .....	3
2.2. Technical model of the Galleon real estate agency .....	4
2.2.1. Program model of the Galleon real estate agency.....	4
2.2.2. Information model of the Galleon real estate agency .....	5
2.2.3. Organizational scheme of the Galleon real estate agency .....	6
2.2.4. Topological model of real estate agency .....	7
2.2.5. “Problems tree” of real estate agency .....	8
2.2.6. Justification of need of creation of an information system .....	9
<b>Beginning of design</b> .....	<b>10</b>
3.1. Description of subject domain .....	10
3.1.1. Purposes and system designations.....	11
3.1.2. Deployment diagram .....	11
3.1.3. Justification of the choice of ASP.NET platform .....	12
3.1.4. Justification of the choice of XML and database .....	13
3.1.5. Comparison of XML data and relational data.....	14
3.1.6. Determination of the data required for work.....	15
3.2. Design of information system software .....	18
3.2.1. Justification of the choice of CASE tools at the design stage .....	18

3.2.2.	Designing a business model for a real estate agency .....	18
3.2.3.	Description of the transaction process .....	19
3.2.4.	Building a detailed model of business.....	20
<b>Model for forecasting prices.....</b>	<b>27</b>	
4.1.	Theoretical justification of the method .....	27
4.1.1.	The classical normal linear model of multiple regression.....	28
4.1.2.	Estimation of the parameters of classical regression of the model by the method of least squares .....	28
4.1.3.	Evaluation of the significance of multiple regression. The coefficients of determination $R^2$ .....	29
4.1.4.	Multicollinearity.....	31
4.1.5.	Selection of the most significant explanatory variables in the regression model	32
4.1.6.	Linear regression models with variable structure. Dummy variables.....	32
4.2.	Implementation of the mathematical model for forecasting prices.....	33
<b>Implementation.....</b>	<b>37</b>	
5.1.	Create a spreadsheet.....	41
5.2.	Add an entry to the database .....	42
5.3.	Delete an entry from the database.....	42
<b>Quick start guide .....</b>	<b>45</b>	
6.1.	Introduction.....	45
6.2.	Preparing for work .....	45
<b>Conclusion.....</b>	<b>53</b>	
7.1.	Problems .....	54
7.2.	Future work.....	54
<b>Bibliographic references .....</b>	<b>55</b>	
<b>Attachment A.....</b>	<b>1</b>	
<b>A Create a spreadsheet .....</b>	<b>2</b>	

# List of Tables

Table 1 - Technical characteristic of the workstation .....	4
Table 2 - Software used by the real estate agency .....	5
Table 3 - "Registration apartment" and "Search for objects" .....	16
Table 4 - Maps.....	16
Table 5 - Feedback .....	17
Table 6 - Employees.....	18
Table 7 - Client model description .....	21
Table 8 - The director.....	23
Table 9 - Employee model .....	24
Table 10 - Use Cases description .....	25

# List of Figures

Figure 1 - Information model of the Galleon real estate agency .....	6
Figure 2 - Organizational scheme of management of Galleon real estate agency .....	7
Figure 3 - Topological model.....	7
Figure 4 - The tree of problems.....	8
Figure 5 - Deployment diagram .....	12
Figure 6 - Conceptual Business Model .....	19
Figure 7 - Deal process.....	20
Figure 8 - Client model .....	21
Figure 9 - The director .....	22
Figure 10 - Employee model.....	24
Figure 11 - Statistical data for Omsk .....	34
Figure 12 - Matrix of coefficients of pair correlation .....	35
Figure 13 - Regression statistics.....	35
Figure 14 - The variance analysis .....	36
Figure 15 - Default page.....	37
Figure 16 - Pages.....	38
Figure 17 - Default .aspx.....	38
Figure 18 - Registratation.....	46
Figure 19 - Log in.....	46
Figure 20 - Registration apartment page .....	47
Figure 21 - Search an apartments .....	48
Figure 22 - Search results.....	49
Figure 23 - Calculation of the apartment's cost .....	50
Figure 24 - Feedback page .....	51
Figure 25 - Administrator page .....	52
Figure 26 - Employee page .....	52

# List of Abbreviations

AIS - Automated Information System

ASP.NET - Active Server Pages for .NET

CASE - Computer Aided Software / System Engineering

C# - Object-oriented programming language

HTML - HyperText Markup Language

RDF - Resource Description Framework

UML - Unified Modeling Language

XML - Extensible Markup Language



# Chapter 1.

## Introduction

The principle of an integrated approach to customer needs is the cornerstone activity of Galleon real estate agency. The list of the services provided by the agency covers all parts of the relations arising upon transition of the property rights, privatizations of the land and structures, commissioning of the objects of the real estate to rent, the lawsuits connected with the real estate and the land. Also real estate assessment is carried out, professional photography of the object offered to sale is carried out.

Relevance of this work is that registration and timely processing of a large number of data, including data storage about objects, is necessary for carrying out any kind of transactions. Constant control of updates and the changes in price, characteristics and existence of objects to sale, is necessary for optimization of work of real estate agency.

### 1.1. Goals

This work aims the development of an Automated Information System (AIS) to the Galleon real estate agency. The object of the research is the Galleon real estate agency.

The creation of an AIS to a real estate agency will allow:

- to reduce operating time on search of objects for the client;
- to reduce the number of the routine operations carried out by the experts;
- to reduce the labor required to collect and register the input information;
- to organize the output information;
- to create a database of the sold objects;
- to increase overall performance.

To achieve this goal, it is necessary to solve the following problems:

- research of real estate agency "Galleon" before the beginning of designing;
- definition of the organizational scheme of the agency;
- identify and characterize the requirements of the Galleon real estate agency;
- drawing up functional, information, technical, program and topological models;
- drawing up a problems tree;
- design of an information system;

## **1.2. Structure of the document**

The work includes a theoretical, analytical and practical part, that are described in this dissertation in 5 chapters.

The first chapter includes an introduction to the work and describes the need to create a product.

In the second chapter, we described the main activities of the agency, identify the main problems and, proceeding from them, the goals and objectives.

In the third chapter, the requirements for future work are presented and describe the structure of the program using the UML diagram. It is also justified the choice of tools and programming languages, used in the development of the information system.

The fourth chapter was devoted to regression analysis. We calculated the importance of the factors necessary for constructing a mathematical model for calculating the cost of an apartment.

The fifth chapter includes a description of the code and basic formulas used in writing the program.

In the sixth chapter, it is presented a guide showing how to use the finished product.



# Chapter 2.

## Inspection of the Galleon agency

### 2.1. Description of activity of real estate agency Galleon

The real estate agency is a specialized commercial organization that supplies, as rule, intermediary services between the seller and the final buyer in the real estate market.

The purpose of a real estate agency is to facilitate and promote business among sellers and buyers of real estate properties, guaranteeing that both parts have a good deal and that the real estate agency wins money with that.[B 15]

The main goals of the real estate agency are:

- improving the quality of work;
- customer satisfaction;
- increase in positive feedback about the agency;
- receiving a profit;
- development and expansion of activities.

The main services provided by real estate agency:

- purchase - property sale;
- search of sellers and buyers;
- legal support of the transaction;
- check of purity of a real estate object;
- registration in the public or other authorities of an object according to the transaction;
- consultations concerning transactions with the real estate;
- assistance in receiving the credits;

- legitimization of changes in a real estate object.

A real estate agency present several advantages to the potential clients. A wide range of properties and a highly specialized and dedicated service. The staff of real estate agency daily works on extension of the database of apartments, and well in it is guided. The real estate agency daily adds new and new options to the base, and the employees offer them to the clients for viewing. The agency undertakes all work on the organization of meetings between sellers and buyers of apartments.

Therefore possibilities of acquisition or property sale by means of the agency are much wider, than to look for necessary option independently.

## **2.2. Technical model of the Galleon real estate agency**

Technical model of the agency - hardware of jobs at office or in other workplace. In the agency workplaces, the personal computers characteristics are the ones presented in the Table 1.

Table 1 - Technical characteristic of the workstation

Work station	Characteristics
Processor	Intel (R) Core (TM) i3 CPU 2.40 GHz
RAM volume	DDR 3 - 6,00 gb
HardDisk volume	500 gb
Drives for optical disks	DVD-RW drive
Motherboard	MSI B85M-P33 V2
Video card	ATI Radeon HD4850 2gb

### **2.2.1. Program model of the Galleon real estate agency.**

A minimum set of the generic and specific software is required for the employees execute their duties. The software used in the agency is presented in Table 2.

Table 2 - Software used by the real estate agency

Software
Microsoft Windows 7 Professional
Anti-virus program ESET Smart Security
Archiver – WinRAR
Microsoft Office 2010
ConsultantPlus

The Microsoft Windows 7 Professional - an operating system with which specialists of the agency work. Anti-virus program ESET Smart Security - the anti-virus program that provides data protection on the computer and at the internet access, blocking attempts of malefactors to take control of data. Archiver - WinRAR - widely used archiver thanks to whom the employees can squeeze large volumes of data, for example, sending to the client a selection of objects with a large number of photos. Microsoft Office 2010 is used to prepare documents, namely reports. ConsultantPlus is an electronic version of Federal Laws of the Russian Federation and the Civil Code of the Russian Federation. They are necessary for drawing up the agreements, contracts and other documents participating in various processes of real estate agency and governing the relations between the client and the agency.

### **2.2.2. Information model of the Galleon real estate agency**

In the course of studying of document flow of real estate agency Galleon, and also obtaining additional data from the staff of the agency, it was built an information model in which the Director and Employees take part, being guided by federal laws of the Russian Federation and the Civil Code of the Russian Federation. As entrance, information serves the application from the client for search or sale of a real estate object, or the analysis of data on real estate prices. At the exit, we receive the contract of purchase and sale, the forecast of the prices or the announcement in an information system and on various advertising websites. The information model of the real estate agency "Galleon" is reflected in Figure 1.

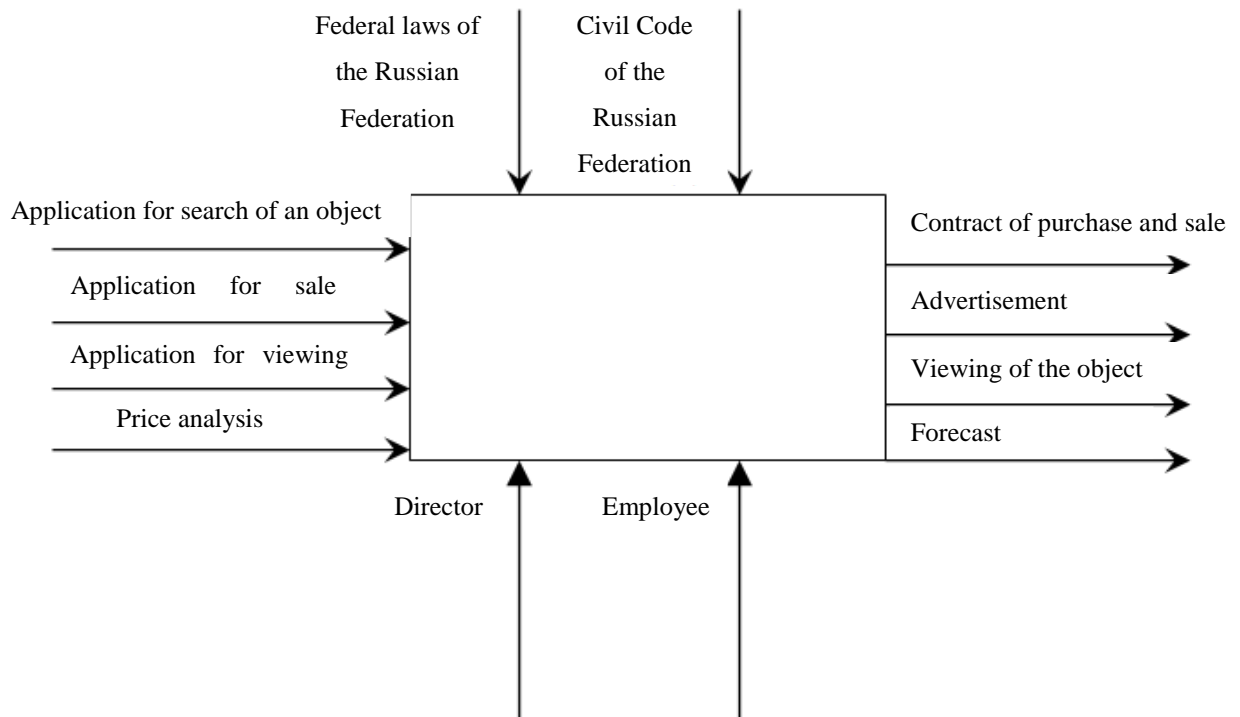


Figure 1 - Information model of the Galleon real estate agency

### 2.2.3. Organizational scheme of the Galleon real estate agency

During performance of analytical work, it was made the organizational scheme of management in the Galleon real estate agency presented at Figure 2.

The organizational scheme in real estate agency Galleon includes 2 elements: The main element is the Director of real estate agency who bears responsibility for all operations with the real estate performed by him or the staff of the agency. The Director also provides training in the agency, exercises supervision of work of the employees. The second element is the team of experts. Under sensitive supervision of the director, they carry out operations with the real estate and with the information system, entering necessary data about the objects. The experts also carry out all communications with the client, advising them, holding meetings and providing verification of documents when carrying out the transaction.

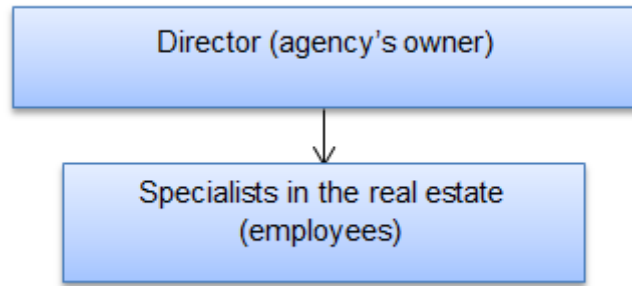


Figure 2 - Organizational scheme of management of Galleon real estate agency

### 2.2.4. Topological model of real estate agency

The topological model is a conditional representation of the room in which object of automation settles down Figure 3.

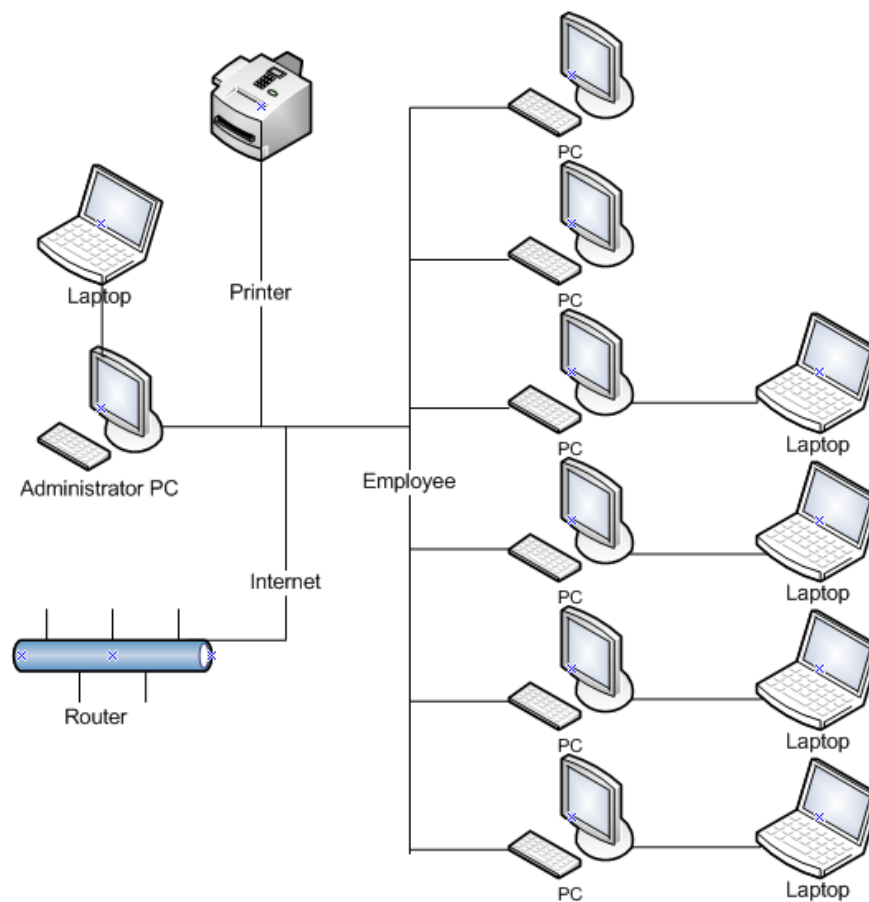


Figure 3 - Topological model

The real estate agency includes 10 jobs of realtors. Six personal computers with the characteristics specified in the Figure 3, and 4 laptops. Existence of one personal computer and one laptop for the administrator (director) is provided. Existence of the laptop is necessary for mobility and for holding meetings abroad with clients. Also at office, a printer is necessary to print documentation and contracts. Wi-Fi router is necessary for ensuring access to Internet.

**2.2.5. “Problems tree” of real estate agency**

"The tree of problems" is a multilevel structure on which the revealed problems are located in a hierarchical order of Figure 4.

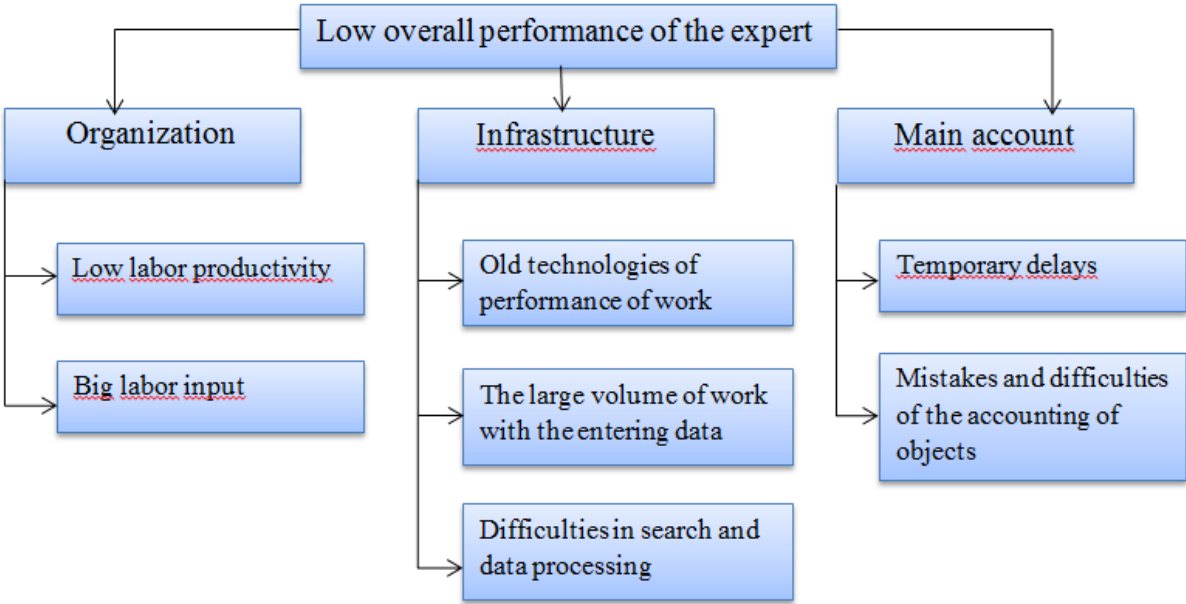


Figure 4 - The tree of problems

This tree represents the set of problems structured, constructed by the hierarchical principle in which the main problem and subproblems of the second and subsequent levels subordinated to it is allocated.

Perhaps, not all problems influencing achievement of goals are revealed, but the most common factors are considered.

## 2.2.6. Justification of need of creation of an information system

Before starting development of personalized software solution, it is necessary to be convinced that this decision is proved. [M 08]

At the moment, the Galleon real estate agency has not a database with the data about the solded objects. Employees process all data manually and on the basis of the obtained information fill in tables. In this case, it is very high the probability of commit a mistake when fill a document or loss/damage a document.

Besides, at the moment, the staff of the agency has only local access to data that creates additional difficulties and big loss of time in the course of work.

The developed AIS of real estate agency will allow:

- to reduce operating time on creation of announcements;
- to reduce the number of the operations carried out by the experts;
- to reduce the labor to collect and register the information about new objects and to fulfill the applications to search for an object;
- to structure output information;
- to create a database of the real estate objects which are available on sale;
- to reduce time necessary for work with a real estate object or the client;
- to provide online - access to data that will provide mobility and besides will reduce time;
- to increase quality of feedback supplied by the clients.
- it is easy and fast to change the personal data and privilege rights of the employees, namely to assign responsibilities and tasks.

Thereby, the information system will allow reduce considerably amount of the time spent by the employees searching the necessary object, it will allow to structure data that considerably will increase productivity of work of the department, and also it will grant to the client the right for independent selection of housing.

# Chapter 3.

## Beginning of design

### 3.1. Description of subject domain

This system is intended for the accounting of the properties available to sell - data recorded by the staff of the enterprise. The system allows to issue the client application, to provide it the catalog of apartments and the price list of all real estate properties.

Description of business processes of subject domain

Adding a property to the database:

- the expert receives a call from the client;
- depending on the type of client (seller/buyer), he fills in an application for the purchase or sale of a property;
- the client receives the invitation in office (time is discussed);
- the client comes to office in due time;
- the client describes the premises of the intended real estate object (the location, the number of rooms, existence of property, etc.)
- the expert enters these premises in the database.

At the future work, the client should be able to introduce directly the data of the real estate object in the database, and the administrator will only need to confirm the conditions by phone and formally accept them in the information system.



### **3.1.1. Purposes and system designations**

The system is intended for automation of an information system of real estate agency. Users of system can perform the functions more effectively, thereby increasing the working capacity. Less time will be spent with proceedings, increasing work efficiency.

The list of the functions of system providing achievement of goals

- the accounting of the arriving real estate objects;
- the accounting of the existing real estate objects;
- the accounting of the sold real estate objects;
- granting statistics of the sold real estate objects;
- search of real estate objects;
- function of feedback with the client;
- viewing of an arrangement of an object on maps;
- sorting of objects according to any criterion;
- viewing and editing information on employees;
- calculating the cost of an apartment.

### **3.1.2. Deployment diagram**

The deployment diagram is intended for visualization of elements and components of the program, existing only at the stage of its execution (runtime). In this case, only the component-instances of the program that are executable files or dynamic libraries are represented. Those components that are not used during the execution phase are not shown in the deployment diagram. [M 08]

The deployment diagram contains graphical representations of processors, devices, processes, and connections between them (Figure 5). Unlike diagrams of logical representation, the deployment diagram is unified for the system as a whole, since it must completely reflect the features of its implementation. [W 08]

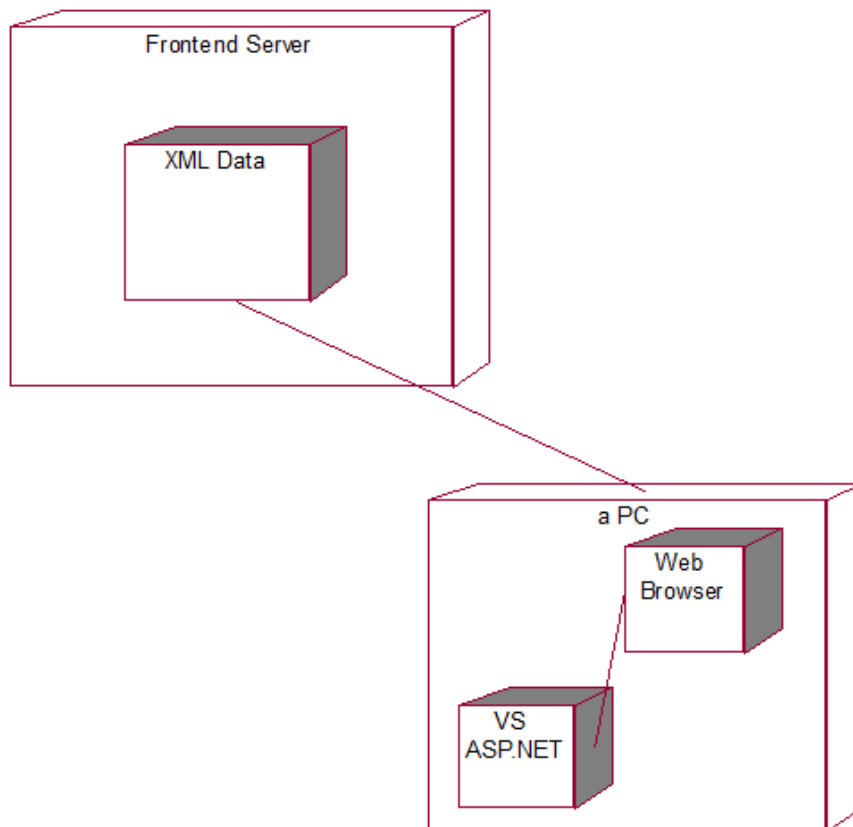


Figure 5 - Deployment diagram

### 3.1.3. Justification of the choice of ASP.NET platform

We use the ASP.NET platform to develop the information system. ASP.NET (Active Server Pages for .NET) is a technology for creating Web applications and web services from Microsoft. Structure of Pages and Controls ASP.NET is a programming structure that runs on a Web server for dynamically creating and displaying ASP.NET Web pages. ASP.NET Web pages can be viewed in any web browsers or client devices, ASP.NET displays markup (such as HTML) in the requesting browser. Typically, you can use the same page for different browsers, because ASP.NET displays the appropriate markup for the requesting browser. ASP.NET Web pages are completely object-oriented. The structure of ASP.NET pages and controls also encapsulates the general functionality of the user interface into convenient reusable controls. Controls written once can be used in many pages. They are embedded in an ASP.NET web page, on which they are placed during rendering. The structure of ASP.NET pages and controls also provides the ability to control the display and behavior of the Web site with themes and covers. Also, we can define the main pages that allow us to create a page layout that can be used for all pages in the application. One master page defines a layout and

standard behavior that can be used for all pages (or a group of pages) in the application. Next, we can create separate content pages that include the content associated with the page that we want to display. When users request a content page, the output page is a combination of the structure of the main page and content from the content page. We can also manage the authentication of a Web application using Windows groups or its own role database using ASP.NET roles. These schemes are easy to add, remove or replace, depending on the requirements for the application. This platform was chosen for development because it includes the "designer" mode, which is easy enough to use at the initial level of programming.

Also during the creation of the information system, we used the programming language C#. To date, the programming language C# is one of the most powerful, fast growing and demanded languages in the IT industry. At the moment, it writes a variety of applications: from small desktop programs to large web portals and web services that serve millions of users every day.

Compared to other languages, C# is quite young, but at the same time it has already gone a long way. The first version of the language was released with the release of Microsoft Visual Studio .NET in February 2002. C# is object-oriented. C# supports polymorphism, inheritance, operator overload, static typing. The object-oriented approach allows solving the tasks of building large, but at the same time flexible, scalable and extensible applications. And C# continues to develop actively, and with each new version there are more and more interesting functionalities, such as lambda, dynamic binding, asynchronous methods, etc.

#### **3.1.4. Justification of the choice of XML and database**

Now various database management systems, differing with both the opportunities and requirements to computing resources, are applied to creation of information systems.

The most important characteristics of this DBMS are:

- simplicity of administration;
- possibility of connection to Web;
- speed and functionality of the mechanism of the DBMS server;
- existence of means of remote access [C 08].

XML is the new Standard Generalized Markup Language (SGML) derivative markup language of documents, allowing to structure information of different types, using for this purpose any set of instructions.

Today XML can be used in any applications which need the structured information - from difficult geographic information systems, to huge volumes of transmitted data. A set of tasks for which XML can be used:[C 08]

- First of all, this technology can be useful to developers of difficult information systems, with a large number of the appendices connected by flows of information the most various structures. In this case, XML documents carry out a role of a universal format for exchange of information between separate components of the big program.
- XML is the basic standard for the modern resource description language - Resource Description Framework (RDF), allowing to simplify many problems in Web connected with search of information, ensuring control of contents of network resources, creations of electronic libraries, etc.
- The XML language allows to describe data of any type and is used for submission of specialized information. It means that XML can serve as powerful addition to HTML for distribution of "non-standard" information in the Web. Perhaps, in the near future, XML will completely replace HyperText Markup Language (HTML), at least, the first attempts of integration of these two languages already become (extensible hypertext markup language - XHTML).
- Information, which is contained in XML documents, can change, be transferred to the client machine and be updated in parts. The developed XML Linking Language (XLink) and Xpointer specifications is tumbling down to refer separate elements of the document, with the accounting of their enclosure and values of attributes.
- XML can be used in usual applications for storage and processing of the structured data in a uniform format. [C 08]

The main advantage of XML documents is that it is easy to create and process (the usual text can be edited by any text processor and be processed by standard XML analyzers), but it is also easy to produce structured information which computers well "understand".

Data of XML are internally ordered as an example from our program.

### **3.1.5. Comparison of XML data and relational data**

Before starting the use of XML in modern databases, briefly we will compare the structure of relational data and XML. In the relational database, information is stored in the

tables. Each type of data is stored in columns. Each record is presented as a line on the table. The order of lines in the table is not connected with any streamlining of data, unlike XML where internally present order of the document influences, for example, it affects the data returned. Only the elementary relational data can be stored in one table; The typical relational database has a data in different tables. For example, in the table Customers can be a field (or a column) CustomerID. Identification of orders for the specific user is facilitated by the existence of the corresponding value in the table Orders CustomerID column. The relationship between data can be "one-to-one", "one-to-more", more-to-more. Each of these relations can be represented by data storage in two or more connected tables. Relational databases usually have no hierarchy per se unlike XML documents. [C 08]

Summarizing, we can briefly list the difference between XML and relational databases:

- The XML data is hierarchical; Relational data are presented in the model of logical relationships. The XML document contains information about the relationships between data items in a hierarchical form.
- The XML document contains not only data, but also tags that explain this data. One XML document can contain different types of data. In the relational model, the data content is specified in the column definition. All data in the column must be of the same type.
- XML data has its own order; relational data does not have this order. For the order of the data in an XML document is received the order in which you specify the data items. As a rule, there is no other way to specify the order of data in the document. [C 08]

### **3.1.6. Determination of the data required for work**

For further development of the automated system, it is necessary to define and describe entities.

In the program there will be forms, one of which (Registration apartment) is necessary for addition of an object with use of certain criteria, and second (Search for objects) is necessary for search of objects in base by the set criteria. This is shown in Table 3. These forms will be almost identical on structure, as the criteria used in the description of an object at its registration in an information system are also used by search of a necessary object.

Table 3 – “Registration apartment” and “Search for objects”

<b>Name of attribute</b>	<b>Description of attribute</b>
ID_Area	District of the city
Type_object	Housing type: primary or secondary
Type_of_Apartment	Number of rooms
Material_Construction	Material of construction of the house
Number_of_floors	The number of floors in the house
Floor	Floor
Area	Area of the object
Address	Object address
Street	Object street
House_No	House number
Building_No	Building number
Price	Object cost
E-mail	E-mail of the client
Phone	Phone number of the client

In the tab to Search for objects, it is necessary to add a viewing with online maps containing the available apartments. This is a very useful feature – for example, a client moved to our city recently that badly knows names of streets or areas, could easily find an object for purchase visualizing online map and choose the option which is most suitable for him. A description of the main elements of this menu is shown in the Table 4.

Table 4 – Maps

<b>Name of attribute</b>	<b>Description of attribute</b>
Address	Address form
ID	Object ID

The Feedback menu will be available to the administrator and workers. Very often happens so that the client finds for himself suitable option and the following step which he should make – is establishment of communication with the agency or the expert in more detail, to study characteristics of the chosen object, to ask necessary questions or to agree about viewing. Thus, without function of feedback, a large number of clients is lost. But there

are other good arguments, if the time zone of client is distinct of the agency, direct contact could not be easy to do. The expert, in a hurry, loses contacts of clients which it is necessary to contact for examination or specification of details and a set of other options. Therefore it was decided to organize a simple form of feedback for clients. The client will choose a suitable object then the form with already entered data on an object will be available in base. The client will need only to add a contact information for feedback and to leave the comment, or to submit its question in the corresponding window. Then the administrator or the employee will receive the application which will be stored in the information system. A description of the main elements of this menu is shown in the Table 5.

Table 5 - Feedback

<b>Name of attribute</b>	<b>Description of attribute</b>
Name	Name of the client
Phone	Phone number of the client
E-mail	E-mail address of the client
Comments	Comments of the client on an object
Message	The message for the client

As it was told before, in our information system users are divided into three types: administrator, employee and guest. We will create another menu for the administrator. Here will be the data of registered users: Name, Surname, Phone number, Login, Password and E-mail of each user. The administrator can register new users, change data on the available users and delete data of the dismissed user. This is the table where will be stored the registration data of all users authorized to access the system. A description of the main elements of this menu is shown in the Table 6.

Table 6 - Employees

<b>Name of attribute</b>	<b>Description of attribute</b>
Name	Name of the employee
Surname	Surname of the employee
Phone	Phone of the employee
E-mail	E-mail of the employee
Login	Login of the employee
Password	Password of the employee

## **3.2. Design of information system software**

### **3.2.1. Justification of the choice of CASE tools at the design stage**

CASE-tools (from Computer Aided Software / System Engineering) allow us to design any system on the computer. A necessary element of system and structural analysis, CASE tools allow us to model business processes, databases, software components, activity and structure of organizations. Practically applicable in all areas of activity. The result of applying CASE-tools - optimization of systems, reducing costs, improving efficiency, reducing the probability of errors. [W 08]

The most suitable is Rational Rose, this tool has the greatest functionality.

### **3.2.2. Designing a business model for a real estate agency**

The main purpose of the real estate agency is the sale and purchase of real estate. As actors, there are: worker, director, client. Figure 6 presents a conceptual business model, which reflects the actors' connections to the functions of the information system.



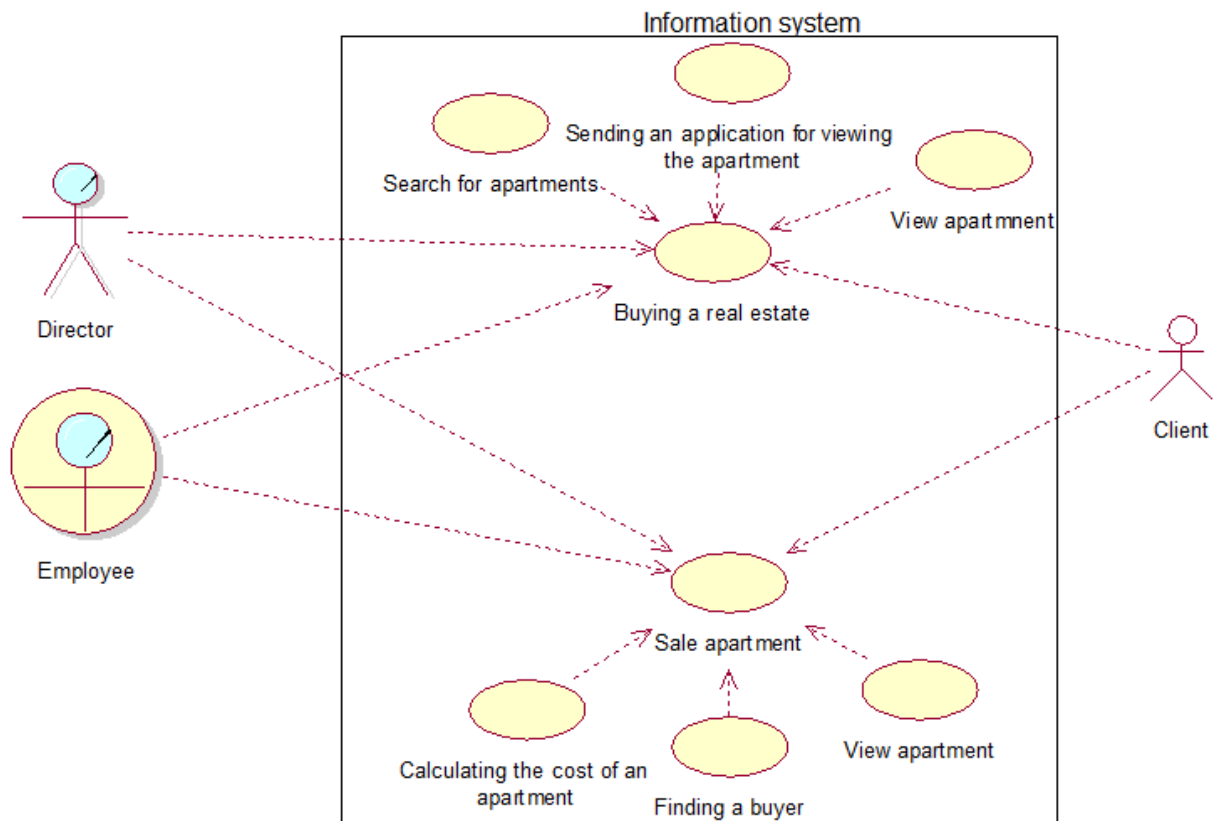


Figure 6 - Conceptual Business Model

### 3.2.3. Description of the transaction process

The main purpose of the real estate agency is the purchase and sale of apartments. We need to understand how the process of buying and selling an apartment from the beginning to the end. In practice, this process looks like this: The first thing the seller do is to register in our system and calculate the cost of his apartment using our price calculation function. Further, he fills in the necessary fields and sends an advertisement on the sale of his apartment to our base. After receiving the application, the agency employee contacts the seller and agrees the selling price and all the conditions. If the seller agrees, the agency employee confirms and adds the ad to the main database. Further, the buyer appears in the process and is similar to the seller, being registered to gain access to the system. After this, the buyer carries out an independent search or immediately appeals to the agency. Once the suitable apartment is found, the agency employee shows the apartment to the buyer. If after viewing the apartment, the buyer is satisfied with everything, then the employee calls the seller, negotiates all the details, and then the buyer and seller conclude a deal. This process is described in Figure 7.

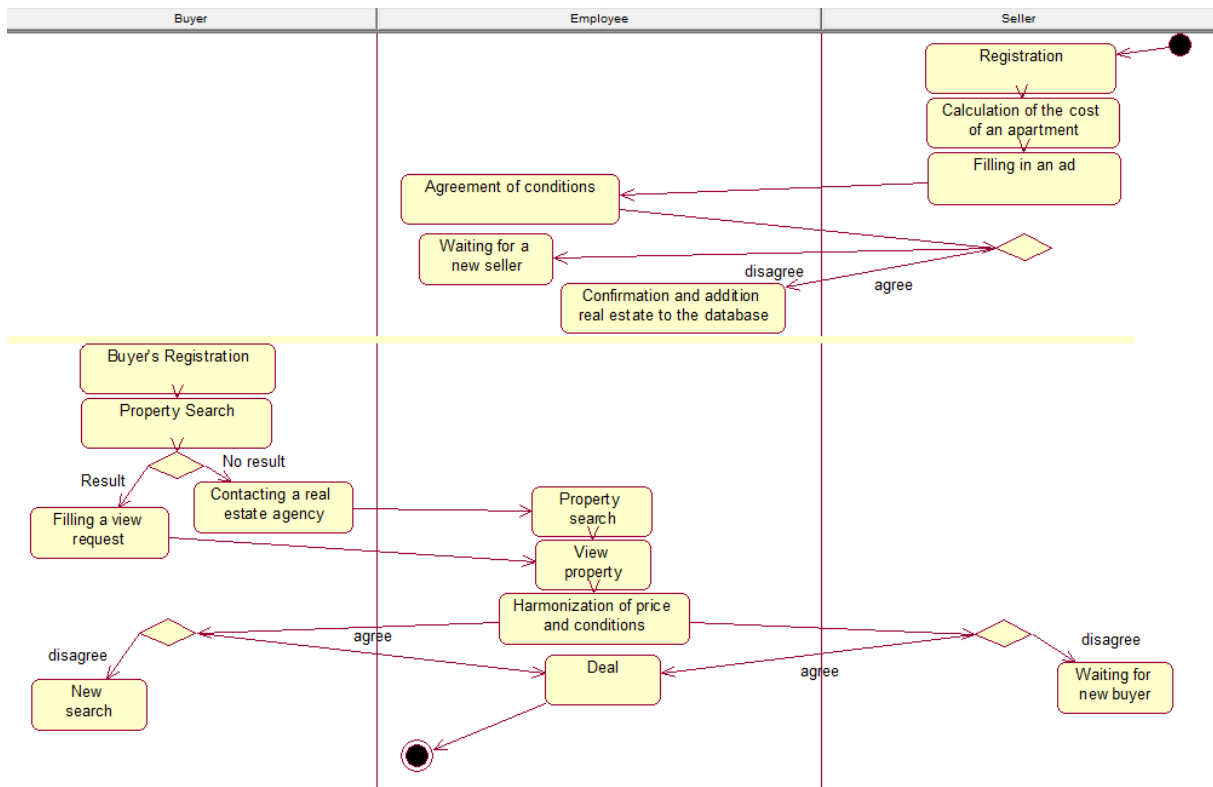


Figure 7 - Deal process

### 3.2.4. Building a detailed model of business

Functional requirements must be in accordance with the context and objectives set by the Business Requirements. Requirements that do not contribute to achieving the Business Objectives of the project should not be included in the project. The problems concerning the image and boundaries must be resolved to the specification of the detailed functional requirements.[M 08]

As mentioned earlier, our project involves: a client, an employee and a director. UML diagrams will be used to display the roles, functions and dependencies of each actor. As you can see, Figure 8 shows the Client model. Description of the actor, its main functions and rights can be seen in Table 7. [W 08]

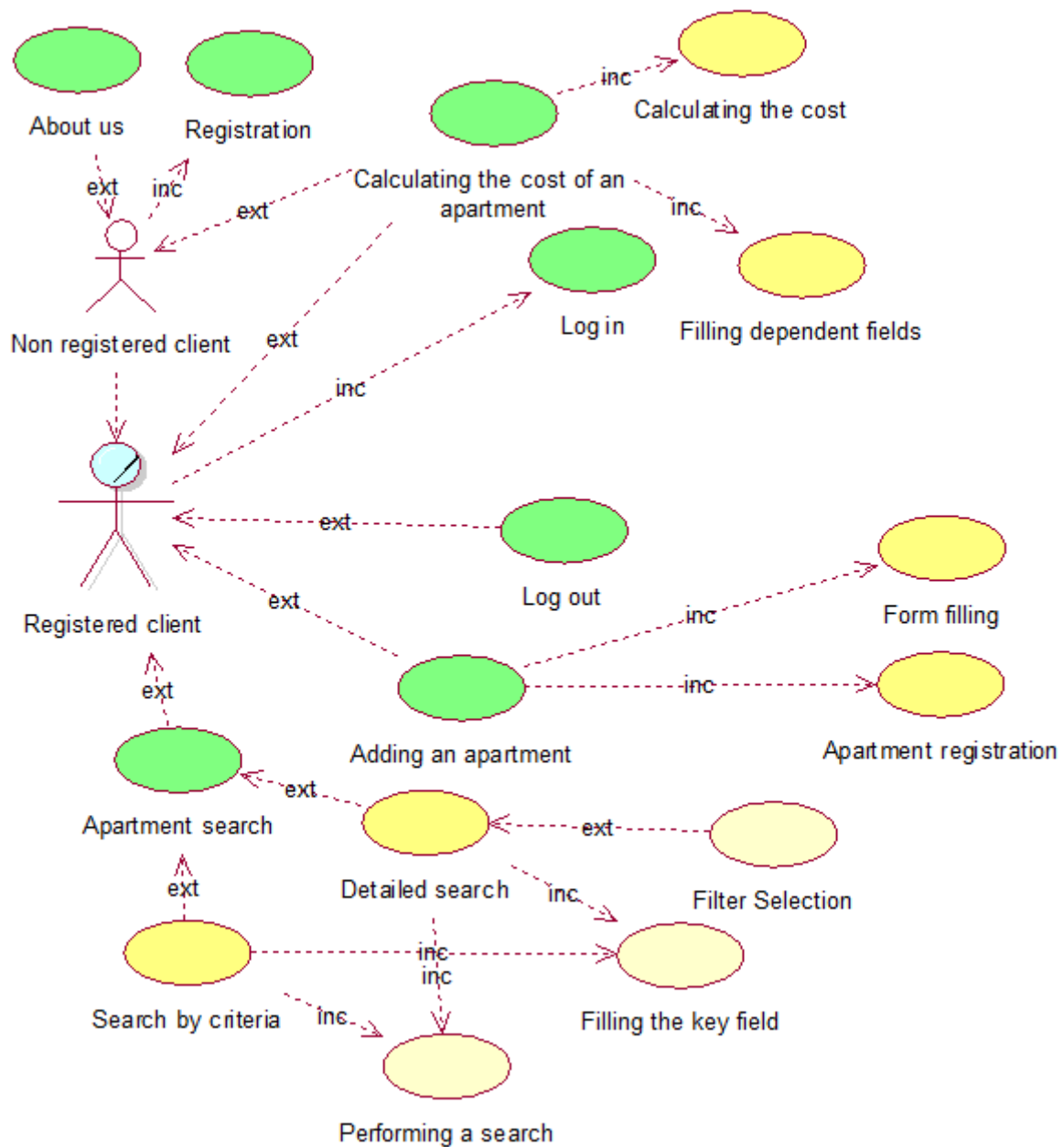


Figure 8 - Client model

Table 7 - Client model description

Actors		Specification
<b>Client</b>	Describe	Uses agency services. After registration can perform the necessary operation with real estate: to look for an apartment; Sell an apartment;

		Send an application for an inspection of the apartment; Calculate the cost of an apartment and so on
	Attributes	Full name, phone number, e-mail, login, password, comments
	Privilege	Significant restriction on access rights

Figure 9 shows the Director's model. This is the main actor who has access to any function, can change the rights for other participants, can change the structure and functionality of the program itself. Description of the model "The director" is in Table 8.

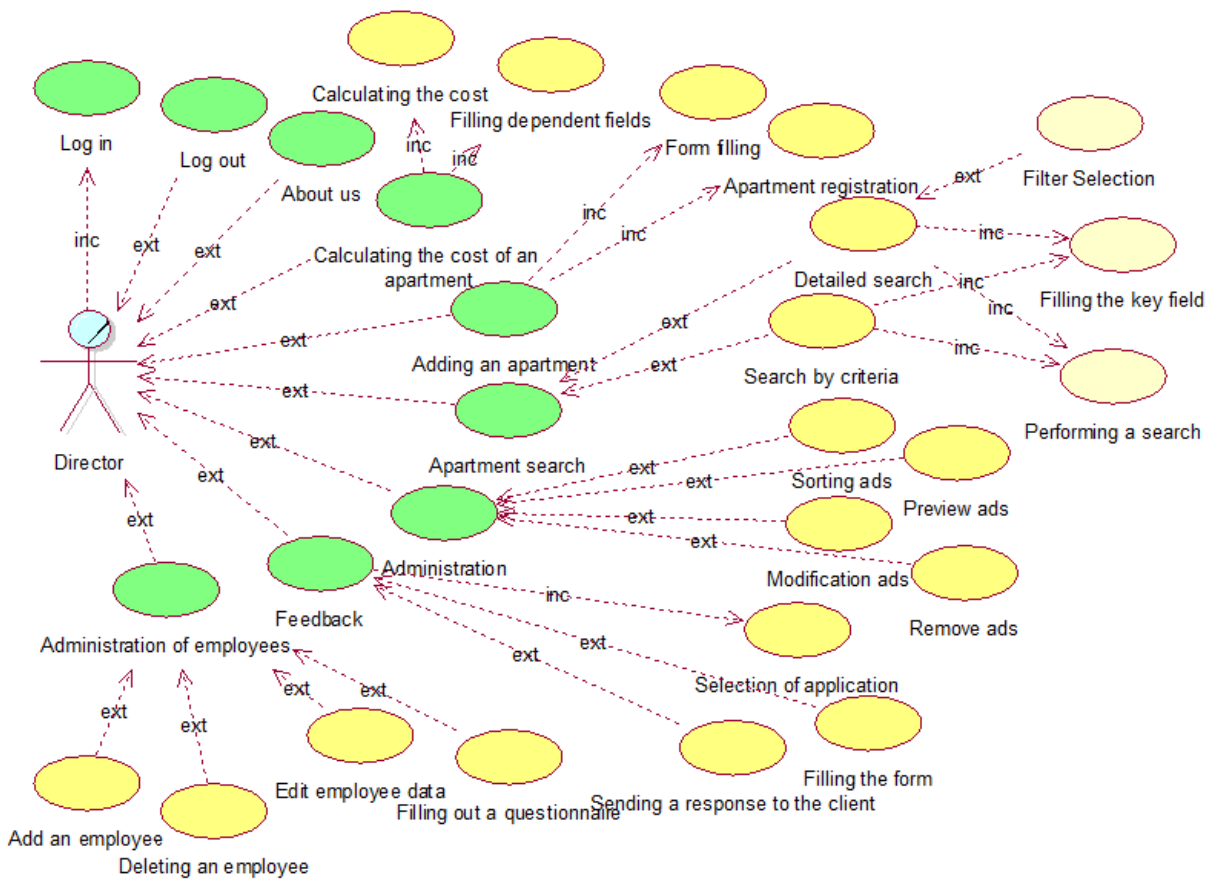


Figure 9 - The director

Table 8 - The director

Actors		Specification
<b>Director</b>	Description	Sets rights for other users; Changes or limits the functionality of the program; Confirms the registration of users, has direct access to the database
	Attributes	Full name, phone number, e-mail, login, password
	Privilege	Highest level, full access

Next, consider the model employee. This actor does not require a self-registration in the system. The administrator needs to add a questionnaire and an account for the new employee. The employee has access to most of the functions. Consider this model in Figure 10 and below its description in Table 9.

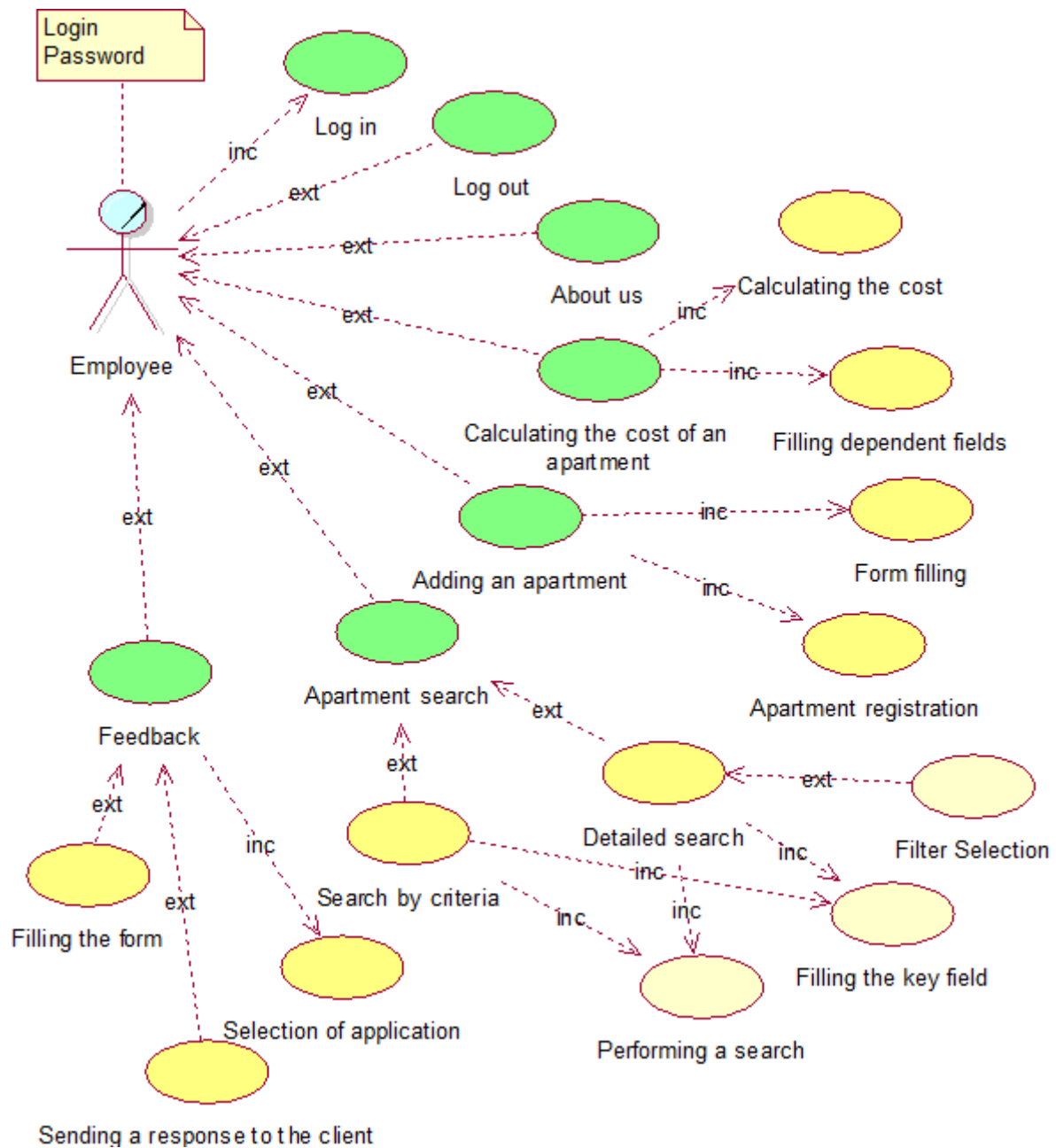


Figure 10 - Employee model

Table 9 - Employee model

Actors		Specification
Employee	Description	Processes incoming requests from customers; Adds an apartment to the base; Has a set of functions inherent in the client; Helps in finding an

		apartment
	Attributes	Full name, phone number, e-mail, login, password
	Privilege	Limited access

After we have built UML diagrams for each actor, we need to describe the meaning of main use cases. Brief description we reflected in the table 10.

Table 10 - Use Cases description

UseCase name		Description
<b>Log in / Log out</b>	Action	User authorization in the system
	Actors	Client, Employee, Director
<b>Calculating the cost of an apartment</b>	Action	Calculating the cost of an apartment based on regression statistics
	Actors	Client, Employee, Director
<b>Adding an apartment</b>	Action	Fill out the form and registration of the apartment in the database
	Actors	Client, Employee, Director
<b>Apartment search</b>	Action	Search for an apartment according to the specified criteria. Detailed search includes the ability to use filters.
	Actors	Client, Employee, Director
<b>Administration</b>	Action	View the database of real estate, change and delete apartments from the database. The ability to sort by any criterion, as well as the actor who added an apartment.

	Actors	Director
<b>Feedback</b>	Action	Receipt of applications from customers, filling out the form and sending a response.
	Actors	Director, Employee
<b>Administration of employees</b>	Action	Add, edit and delete any information about employees. Including login data
	Actors	Director



# Chapter 4.

## Model for forecasting prices

### 4.1. Theoretical justification of the method

This chapter will describe and build a mathematical model for forecasting prices. While writing this part of the thesis, additional literature was used for a more accurate description of the mathematical model.

The effectiveness of the functioning of the real estate market, largely dependent on the development of its administrative structure and the level of information security of the governance structure, acquires a strategic importance at the macroeconomic level. Therefore, for the solution of regulatory and management problems, it is necessary to fulfill such conditions as the creation of an information and analytical database that satisfies the requirements of reliability, completeness and accuracy of the source information that facilitates the adoption of well-founded management decisions, the optimization of the distribution of real estate objects by types, Formation of strategy and management programs, minimizing the costs of maintaining the management system. [KP 02]

Fulfillment of the proposed conditions and, as a consequence, the solution of the tasks posed require information about trends in the market development, which can only be obtained by combining qualitative and quantitative approaches to forecasting the main indicators of the real estate market revealed as a result of systemic studies. They contribute to the development of integrated thinking in the process of developing and making managerial decisions. [KP 02]

System analysis of any component of the socio-economic system, including the real estate market, should begin with a qualitative study of the system itself, which allows to

determine which trends are activated and relevant in a given situation. However, a qualitative analysis of the relevant tendencies and variables is not able to provide an objective assessment of the behavior of the system due to the limited examples of the systems being studied. This determines the need to use, along with a qualitative approach, quantitative methods and models that can take into account the complexity of the systems under study and the processes that take place in them. [R 15]

At the present stage, the needs for the forecast information required for effective management are not fully met. This indicates the relevance of setting and solving the problems of developing and testing models for forecasting the main indicators of the real estate market. Forecasting, as one of the forms of state regulation, serves as a base stage, precedes the development of programs, plans, strategies for socio-economic development. The main function of forecasting is the formation of scientific prerequisites for making managerial decisions in market conditions [R 15].

#### **4.1.1. The classical normal linear model of multiple regression**

Economic phenomena are usually determined by a large number of both the cumulative effects of factors. In connection with this, the problem frequently arises of investigating the dependence of one dependent variable  $Y$  on several explanatory variables  $X_1, X_2, \dots, X_n$ . This problem is solved with the help of multiple regression analysis.

$i$ -th observation of the dependent variable  $y_i$ , and  $x_{i1}, x_{i2}, \dots, x_{ip}$  are explanatory variables. The multiple linear regression model is presented as:

Equation 1

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \varepsilon_i.$$

where  $i = 1, 2, \dots, n$ . [KP 02]

#### **4.1.2. Estimation of the parameters of classical regression of the model by the method of least squares**

To estimate the unknown parameter vector  $\beta$  is applied the method of least squares.

Build a system of normal equations in matrix form:

Equation 2

$$X'Xb = X'Y$$

The solution of the equation is the vector

Equation 3

$$b = (X'X)^{-1} X'Y$$

Where  $(X'X)^{-1}$  is the matrix inverse to the coefficient matrix of system (Equation 2)

$X'Y$  is a matrix-column, or vector, of its free terms.

Finding the matrices entering into equation (Equation 2). The matrix  $X'X$  represents the matrix of sums of the first powers, squares and pairwise products of  $n$  observations of the explanatory variables:

Equation 4

$$X'X = \begin{pmatrix} n & \sum x_{i1} & \dots & \sum x_{ip} \\ \sum x_{i1} & \sum x_{i1}^2 & \dots & \sum x_{i1}x_{ip} \\ \dots & \dots & \dots & \dots \\ \sum x_{ip} & \sum x_{i1}x_{ip} & \dots & \sum x_{ip}^2 \end{pmatrix}$$

The matrix  $X'Y$  is a vector of products of  $n$  observations of explanatory and dependent variables: [KP 02]

Equation 5

$$X'Y = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_{11} & x_{21} & \dots & x_{n1} \\ \dots & \dots & \dots & \dots \\ x_{1p} & x_{2p} & \dots & x_{np} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i x_{i1} \\ \dots \\ \sum y_i x_{ip} \end{pmatrix}$$

### 4.1.3. Evaluation of the significance of multiple regression. The coefficients of determination $R^2$

As in the case of the paired regression model, in the multiple regression model, the total variation of  $Q$  is the sum of the squares of deviations of the dependent variable from the mean: [KP 02]

Equation 6

$$Q = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n y_i^2 - \frac{(\sum_{i=1}^n y_i)^2}{n} = Y'Y - n\bar{y}^2$$

Equation 7

$$Q_e = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = Y'Y - 2b'Y'Y + b'X'Xb = Y'Y - b'X'Y$$

Equation 8

$$Q_R = Q - Q_e = Y'Y - n\bar{y}^2 - (Y'Y - b'X'Y) = b'X'Y - n\bar{y}^2$$

The multiple regression equation is significant (H0 hypothesis that the parameters of the regression model are zero, that is, H0:  $\beta_1 = \beta_2 = \dots = \beta_p = 0$  is rejected), if

Equation 9

$$F = \frac{Q_R (n - p - 1)}{Q_e p} > F_{\alpha; p; n-p-1},$$

where  $F_{\alpha; p; n-p-1}$  is the tabular value of the Fisher-Snedekor F-test, and  $Q_R$  and  $Q_e$  are determined by formulas (Equation 8) and (Equation 7). [KP 02]

Evaluation of the significance of regression coefficients using Student's t-test is reduced to the calculation of the value:

Equation 10

$$t_{b_i} = \frac{b_i}{m_{b_i}} = \sqrt{F_{xi}}$$

where  $m_{b_i}$  is the mean square error of the regression coefficient  $b_i$ , it can be determined by the formula:

Equation 11

$$m_{b_i} = \frac{\sigma_y \sqrt{1 - R_{yx1...xp}^2}}{\sigma_{xy} \sqrt{1 - R_{xi1...xp}^2}} * \frac{1}{\sqrt{n - m - 1}}$$

The coefficient of determination (or the multiple coefficient of determination)  $F^2$  is determined by the formula:

Equation 12

$$R^2 = \frac{Q_R}{Q} = \frac{b'X'Y' - n\bar{y}^2}{Y'Y - n\bar{y}^2}$$

$R^2$  characterizes the proportion of the variation of the dependent variable due to the regression or variability of the explanatory variables; the closer  $R^2$  to unity, the better the regression describes the relationship between explanatory and dependent variables.

At the same time, the use of only one determination coefficient  $R^2$  for selecting the best regression equation may not be sufficient. In practice, there are cases when a poorly defined regression model can give a relatively high coefficient  $R^2$ . [R 15]

The disadvantage of the determination coefficient  $R^2$  is that it increases with the addition of new explanatory variables, although this does not necessarily mean an improvement in the quality of the regression model. In this sense, it is preferable to use the corrected determination coefficient  $\bar{R}^2$ , determined by the formula:

Equation 13

$$\bar{R}^2 = 1 - \frac{n-1}{n-p-1}(1-R^2)$$

#### 4.1.4. Multicollinearity

Multicollinearity is understood as a high mutual correlation of explanatory variables. Multicollinearity can be manifested in functional (explicit) and stochastic (latent) forms.

In the functional form of multicollinearity, at least one of the paired links between explanatory variables is a linear functional dependence. In this case, the matrix  $X'X$  is special, since it contains linearly dependent column vectors and its determinant is zero, that is, the precondition for regression analysis is violated. This leads to the impossibility of solving the corresponding system of normal equations and obtaining estimates of the parameters of the regression model. [KP 02]

However, in economic studies, multicollinearity is more often manifested in a stochastic form, when there is a close correlation between a minimum of two explanatory variables. The matrix  $X'X$  in this case is nonsingular, but its determinant is very small.

To eliminate or reduce multicollinearity used a number of methods. The simplest of these is that of two explanatory variables that have a high correlation coefficient (greater than

0.8), one variable is excluded from consideration. At the same time, what variable to keep and what to remove from the analysis, decide primarily based on economic considerations. If from the economic point of view none of the variables can be given preference, then keep one of the two variables that has a greater correlation coefficient with the dependent variable. To eliminate multicollinearity can be used the transition from the initial explanatory variables  $X_1, X_1, \dots, X_n$  interconnected fairly close correlation, the new variables representing linear combinations of the original. At the same time, new variables must be weakly correlated, or generally uncorrelated. As such variables, take, for example, the main components of the vector of the initial explanatory variables studied in the component analysis. Next, regression is considered on the main components, in which the latter act as generalized explanatory variables that are subject to further meaningful (economic) interpretation [KP 02].

#### **4.1.5. Selection of the most significant explanatory variables in the regression model**

Another possible method for eliminating or reducing multicollinearity is the use of step-by-step procedures for selecting the most informative variables. For example, in the first step, only one explanatory variable is considered that has the largest coefficient of determination with the dependent variable  $Y$ . In the second step, a new explanatory variable is included in the regression, which, together with the originally selected one, forms a pair of explanatory variables, which has the highest (corrected) coefficient of determination with  $Y$ . At the third step, another explanatory variable is introduced into the regression, which together with the two originally selected forms a triple of explanatory variables, having the largest (corrected) coefficient of determination with  $Y$ , and etc.

The procedure for introducing new variables continues as long as the corresponding (corrected) determination coefficient  $\widehat{R}^2$  (more precisely, the minimum value  $\widehat{R}_{\min}^2$ ) increases. [R 15]

#### **4.1.6. Linear regression models with variable structure. Dummy variables**

In practice, it is often necessary to study the influence of qualitative features that have two or more levels (gradations). Among such attributes are: floor (first or last, others), building type (primary, secondary), etc. [KP 02]

Qualitative characteristics can significantly affect the structure of linear relationships between variables and lead to a discontinuous change in the parameters of the regression model. In this case, it is necessary to study regression models with a variable structure or to construct regression models from heterogeneous data.

There is an approach that allows one to evaluate the influence of the values of quantitative variables and levels of qualitative characteristics using a single regression equation. This approach is associated with the introduction of fictitious (manikin) variables, or mannequins.

A qualitative difference can be formalized using any variable that takes two different values, not necessarily "0" or "1". However, in econometric practice, fictitious variables of type "0-1" are almost always used, since the interpretation of the results obtained here is the simplest. [R 15]

If the considered qualitative feature has several (k) levels (gradations), then a discrete variable is introduced into the regression model, taking the same number of values.

## **4.2. Implementation of the mathematical model for forecasting prices**

There are various tools for implementing similar tasks such as: Statistics views, economic views, Statgraphics, etc. Easy to use add-in Excel «Analysis Package" because it provides all the essentials, and also gives us the opportunity to continue to program in VBA and build interface. [R 15]

We have chosen the data of Omsk's real estate to build a model for forecasting prices of real estate.

Dependent variable - the cost of one square meter of real estate. Independent - the number of rooms, area of the city, floor, total area, type of house, material. [B 15]

Figure 11 presents a part of the statistical data for Omsk

Y	X1	X2	X3	X4	X5	X6	X7
Price p/m2	ID_area	Type_Object	Type_of_Apartment	Material_Construction	Area	Floor	Number_of_Floors
53846,15385	5	2	2	2	65	0	5
35000	2	2	2	2	77	1	9
37000	2	2	1	2	33	0	9
49761,90476	1	1	2	3	42	1	5
51123,59551	5	1	3	2	89	0	9
36551,72414	3	1	1	2	29	0	5
38529,41176	3	1	1	2	34	1	9
47666,66667	3	1	1	2	30	0	9
41818,18182	2	1	1	2	22	0	5
33666,66667	2	1	1	3	30	1	9
29000	5	1	1	3	30	0	9
34482,75862	5	1	1	2	29	0	5
33870,96774	5	1	1	2	31	1	5
36666,66667	5	1	1	2	30	1	5
34090,90909	5	1	1	3	33	1	9
55142,85714	1	1	1	2	21	0	5
41555,55556	4	1	2	3	45	1	5
43333,33333	4	1	2	3	48	0	9
50000	3	1	2	2	42	1	5
32619,04762	2	1	2	2	42	0	5
31795,45455	2	1	2	2	44	0	5
31818,18182	2	1	2	3	44	1	5
35128,20513	5	1	2	2	39	1	5
30888,88889	5	1	2	3	45	1	5

Figure 11 - Statistical data for Omsk

Figure 11 introduces the following notation:

Y – the price of real estate (thousand rubles per square meter);

X1 – area of the city;

X2 – type of building;

X3 – number of rooms;

X4 – material;

X5 – total area (sq.m.);

X6 – floor;

X7 – number of floors;

It is necessary to build correlation-regression models. A matrix of coefficients of pair correlation is constructed to eliminate insignificant factors (Figure 12)



Y	1							
X1	-0,132	1						
X2	-0,1436	0,04281	1					
X3	-0,0419	-0,0266	-0,1196	1				
X4	-0,0872	0,02565	-0,1956	0,08681	1			
X5	0,14297	-0,0666	0,06341	0,76267	-0,0255	1		
X6	0,11245	-0,0125	0,03768	-0,0164	-0,0318	0,01586	1	
X7	0,20183	0,00984	0,27244	-0,0124	-0,0439	0,20199	0,0944	1

Figure 12 - Matrix of coefficients of pair correlation

A multi-factor model is being constructed where Y - the price of real estate is the response, X1 - area of the city, X2 - type of building, X3 - number of rooms, X4 - material, X5 – total area, X6 – floor, X7 – number of floors, - factors.

The multifactorial model has the form:

$$Y=48460,28984-662,178713*X_1-9787,767077*X_2-3873,487154*X_3-1164,713507*X_4+201,1990145*X_5+1750,138678*X_6+879,0563569*X_7$$

Figure 13 shows the regression statistics.

<i>Regression statistics</i>	
Multiple R	0,933320979
R-square	0,889200651
The normalized R-square	0,888722807
Standard error	510,6363723
Observations	139

Figure 13 - Regression statistics

The multiple correlation index (multiple R) is equal to 0.93, which means a pronounced tightness of the influence of factors on the result.

The coefficient of multiple determination (R-square) is 0.88, which shows the high quality of the model built.

The adjusted coefficient of determination (normalized R-square) is 0.88. This means that the correction of the R-square had a high influence on the value of the coefficient of determination. [R 15]

The standard error is 510.6, which means a good forecast quality.

Figure 14 shows the variance analysis:

	<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>The significance of F</i>
Regression	7	257399787,1	36771398,16	141,0276	8,28E-58
Balance	131	34156810,06	260739,0081		
Total	138	291556597,2			

Figure 14 - The variance analysis

$F = 141.03$ , the significance of  $F = 8.28$ , which means a high significance of the result.

The constructed model has a high coefficient of determination - 88.28%, which allows using it in forecasting.

During the analysis of the coefficients can be concluded that the constructed model can be used for future predictions.

# Chapter 5.

## Implementation

The program is implemented in C # ASP # in Visual Studio 2017. The program is located on a hosting server that supports Windows. Visual Studio allows to run the program on the local computer. In this case, there is no need to install a virtual server. Visual Studio does this automatically and opens the program in the WEB browser, installed on your computer by default (Figure 15): [V 07]

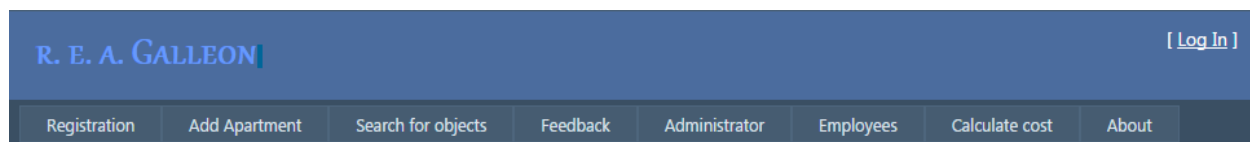


Figure 15 - Default page

As you can see from the figure, the program consists of seven pages:

- Registration
- Add Apartment
- Search for object
- Feedback
- Administrator
- Employees
- Calculate cost
- About

Figure 16 shows the specified pages with the extension .aspx.

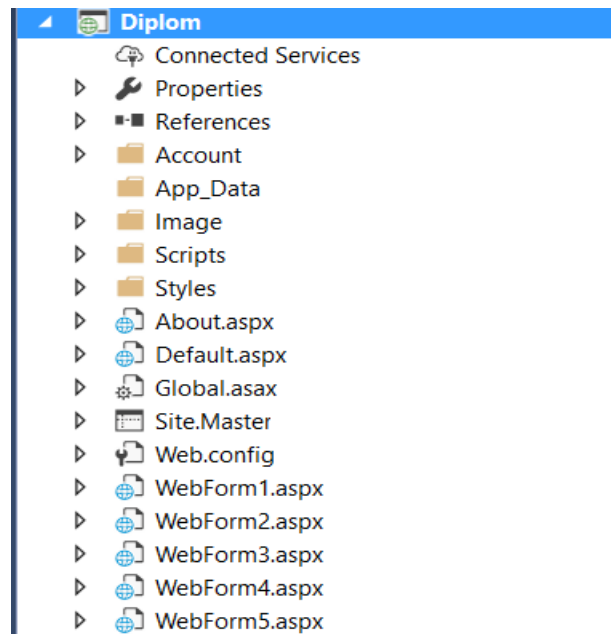


Figure 16 - Pages

When the program starts, the registration window of the Default.aspx page appears (Figure 17):

## Registration

Name	<input type="text" value="egor"/>
Log In	<input type="text" value="Gulakov"/>
Password	<input type="password" value="..."/>
	<input type="text"/>
<input type="button" value="To register"/>	<input type="button" value="Sign In"/>

Figure 17 - Default .aspx

The program provides three types of users: "Administrator", "Employee", "Guest". In the Global.asax file, users are registered:

```
void Session_Start(object sender, EventArgs e)
{
    // Code that runs when a new session is started
}
```

```

Application.Lock();
if (Application["mode"] == null)
    Application["mode"] = "";
Application.Unlock();
}

```

If Application["mode"] = 1 - is an administrator, if Application["mode"] = 2 - is an employee, if Application["mode"] = 3 - is a guest. For synchronization sessions used blocking flow Application.Lock(); And Application.Unlock();

The user enters his data into the text fields "Name", "Log In", "Password" (Figure 17). The program reads them from the database and sets the corresponding Application ["mode"]. If this is an administrator, then all tabs of the pages become available (see Figure 15). [O 05]

Logins and passwords of registered employees are stored in the file Emploes.xml:

```

<?xml version="1.0" standalone="true"?>
- <NewDataSet>
  - <Employees>
    <ID>2</ID>
    <Name>Kirill</Name>
    <Surname>Postevoy</Surname>
    <Phone>933-677-375</Phone>
    <E_Mail>kirillpost@mail.ru </E_Mail>
    <Login>levon</Login>
    <Password>987</Password>
    <Date>5/15/2017</Date>
  </Employees>
  - <Employees>
    <ID>3</ID>
    <Name>Vahan</Name>
    <Surname>Vahanyan</Surname>
    <Phone>010-55-38-65</Phone>
    <E_Mail>vah@mail.ru</E_Mail>
    <Login>vahan</Login>
    <Password>375</Password>
    <Date>5/15/2017</Date>
  </Employees>
</NewDataSet>

```

The ReGister.xml file where it was wrote down, it was edit and stored data about real estate:

```

<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Egor>
    <ID>13</ID>
    <ID_Area>Central</ID_Area>
    <Type_Object>Secondary</Type_Object>
    <Apart>4</Apart>
    <Mat_Const>Panel</Mat_Const>
    <Num_Floors>10</Num_Floors>
    <Floor>5</Floor>
    <Area>300</Area>
    <Street>Lenina</Street>
    <House>114</House>
  </Egor>

```

```

    <Building>2</Building>
    <Price>1790000</Price>
</Egor>
<Egor>
  <ID>14</ID>
  <ID_Area>Kirovsky</ID_Area>
  <Type_Object>Primary </Type_Object>
  <Apart>3</Apart>
  <Mat_Const>Brick</Mat_Const>
  <Num_Floors>10</Num_Floors>
  <Floor>5</Floor>
  <Area>100</Area>
  <Street>Pushkina</Street>
  <House>10</House>
  <Building>1</Building>
  <Price>1420000</Price>
</Egor>
</NewDataSet>

```

The first line XML of the document is called the announcement of XML (declaration) it is the optional line specifying the version of the XML standard. [O 05]

Let's pay attention to a tag of "NewDataSet". As we work with DataSet, at preservation of contents of DataSet in the XML file, this tag is added automatically.

Besides the file has tags of "ID" - the identifying number of record (it is unique), "ID\_Area" - the area, "Type\_Object" - Primary or Secondary housing, etc.

Customer applications are stored in the file Admin.xml:

```

<?xml version="1.0" standalone="true"?>
  <NewDataSet>
    <Admin>
      <ID>3</ID>
      <ID_Area>Sovetsky</ID_Area>
      <T_Object>Secondary</T_Object>
      <Apart>3</Apart>
      <MatConst>Panel</MatConst>
      <N_Floors>5</N_Floors>
      <Floor>3</Floor>
      <Area>75</Area>
      <Street>Lenina</Street>
      <House>12</House>
      <Building>2</Building>
      <Price>2300000</Price>
      <Name>Dmitry</Name>
      <Phone>933974583</Phone>
      <Mail>dim-di@mail.ru</Mail>
    </Admin>
  </NewDataSet>

```

The data on the registered guests is in the Log\_pass.xml file:

```

<?xml version="1.0" standalone="true"?>
  <NewDataSet>
    <Regis>
      <ID>4</ID>
      <login>Ivan</login>
    </Regis>
  </NewDataSet>

```

```

    <password>qwerty</password>
    <name>Ivan</name>
</Regis>
<Regis>
    <ID>6</ID>
    <login>Dima</login>
    <password>17412g</password>
    <name>Dmitry</name>
</Regis>

```

## 5.1. Create a spreadsheet

Consider the source code of the program for ReGister.xml.

Three class objects are described:

```

DataTable table = new DataTable("Egor"); - creates a table named
"Egor"
DataSet dataSet ; - Manages the database (reads, writes, edits)
DataColumn column ; - creates a table column
The initXML() function is called once when the page loads:
protected void initXML()
{
    column = new DataColumn();
    // We need to initialize it. The type of the column, we need
to specify the type of .Net
    column.DataType = Type.GetType("System.Int32");
    column.ColumnName = "ID"; // column name
    // Field autoincrement, and if AutoIncrement set to "true"
and the field type is not "int", then we get exception
    column.AutoIncrement = true;
    column.ReadOnly = false; // can not be changed
    column.Unique = true; // unique
    table.Columns.Add(column); // add to the table

```

See the full code in the attachment A.

Function Read(); - reads data from the server and writes them to the GridView1 control:

```

private void Read()
{
    try
    {
        dataSet.Clear();
        dataSet.ReadXml(Server.MapPath("~/ " + "ReGister.xml"));
        GridView1.DataBind();
    }
    catch
    {
    }
}

```

The Write() function writes data to the server:

```
private void Write()
{
    dataSet.WriteXml(Server.MapPath("~/\" + "ReGister.xml"));
    GridView1.DataBind();
}
```

## 5.2. Add an entry to the database

```
protected void Button_Registration_Click(object sender, EventArgs e)
{
    DataRow row; // a class object for adding one object
    row = table.NewRow();

    row["ID_Area"] = TextBox3.Text;
    row["Type_Object"] = TextBox4.Text;
    row["Apart"] = TextBox5.Text;
    row["Mat_Const"] = TextBox6.Text;
    row["Num_Floors"] = TextBox8.Text;
    row["Floor"] = TextBox9.Text;
    row["Area"] = TextBox7.Text;
    row["Street"] = TextBox10.Text;
    row["Building"] = TextBox11.Text;
    row["House"] = TextBox12.Text;
    row["Price"] = TextBox2.Text;
    row["Mail"] = TextBox13.Text;
    row["Phone"] = TextBox14.Text;
    row["User"] = Label_user.Text;
    row["Buy_Sell"] = false;
    row["Date"] = DateTime.Now.ToShortDateString();
    table.Rows.Add(row);

    GridView1.DataBind();

    Write();
    ClearTextBox(); // Clear all text fields
}
```

## 5.3. Delete an entry from the database

```
protected void Button_Delete_Click(object sender, EventArgs e)
{
    try
    {
        if (table.Rows.Count != 0 && index != -1)
        {
            table.Rows[index].Delete();//Delete the selected object

            table.AcceptChanges();
            Write(); // Write to the database after removal
        }
    }
}
```



```
        else
        {
            Label_Info.Text = "Error: Selected Row";
        }
    }
    catch
    {
        Label_Info.Text = "Error: Selected Row";
    }
}
```



# Chapter 6.

## Quick start guide

### 6.1. Introduction

#### **Brief description of the possibilities**

The developed information system is designed to automate the processing of information on real estate, the implementation of forecasting prices for real estate, as well as maintaining a database of real estate.

#### **Purpose and conditions for the application of the information system**

The developed information system will allow to automate the processes of gathering information about real estate; to calculate the preliminary cost of an apartment to any user; and will also provide the ability to independently search for real estate for any user

Work with the information system is available to all users with the set access rights.

### 6.2. Preparing for work

#### **The procedure for downloading data and programs**

To start working with the information system on the user's workplace, follow these steps:

After starting the program - the Login and Registration page will open.

The new user logs in and presses the “To register” button as shown in Figure 18

The screenshot shows the registration interface of the R. E. A. GALLEON system. At the top, there is a navigation bar with the system name and a [ Log In ] link. Below this is a menu with options: Registration, Add Apartment, Search for objects, Feedback, Administrator, Employees, Calculate cost, and About. The main content area is titled 'Registration' and contains a form with the following fields and buttons:

Name	<input type="text" value="Eena"/>
Log In	<input type="text" value="Elena"/>
Password	<input type="password" value="....."/>
<input type="button" value="To register"/>	<input type="button" value="Sign In"/>

Figure 18 - Registratation

After passing the registration, the user needs to log in to the system. To do this, the user must enter his registration data and press the "Sign In" button, as shown in Figure 19.

This screenshot shows the same registration form as in Figure 18, but with a confirmation message below it. The form fields are: Name (Eena), Log In (Elena), and Password (masked with dots). The buttons are 'To register' and 'Sign In'. Below the form, the text 'Registration successfully completed' is displayed.

Registration successfully completed

Figure 19 - Log in

After a successful login, the user enters the real estate registration section. This section is available to any type of user. If the user wants to sell the apartment - he fills the main form fields and presses the "Registration" button. This is shown in Figure 20.

After clicking the "Registry" button, the generated advertisement is published in the real estate database.

## Registration Apartment

Guest: elena

ID Area	Sovetsky	Sovetsky	<input type="checkbox"/>
Type Object	Secondary	Secondary	<input type="checkbox"/>
Type of Apartment	2-room	2-room	<input type="checkbox"/>
Material Construction	Brick	Brick	<input type="checkbox"/>
Number of Floors	9		<input type="checkbox"/>
Floor	4		<input type="checkbox"/>
Area	78		<input type="checkbox"/>
Address	Street: Pushkina House №: 67 Building №: 1		<input type="checkbox"/>
Price	2680000		<input type="checkbox"/>
E-Mail	elena55@mail.ru		<input type="checkbox"/>
Phone	86584629		<input type="checkbox"/>
		<input type="button" value="Registration"/>	

Figure 20 - Registration apartment page

If the user wants to buy an apartment, he chooses the page "Search for objects". In this section, he performs an independent search for real estate by specified filters. A user can search for an apartment using one or more filters. It is possible to search for an apartment by any numeric or text value. Fields "floor", "area" and "price" have the ability to set the range. When using this range - the list of found apartments will be larger.

It will be necessary to find the added ad. To do this, it is necessary to enter three values: "ID Area", "Material Construction", and set the price range. After filling in the required fields - the user should tick the box on the right and click the "Search" button, as shown in Figure 21.

Registration

Add Apartment

Search for objects

Feedback

Administrator

Employees

Calculate cost

About

## Search for objects

Guest: elena

ID Area	Sovetsky	Sovetsky	▼	<input checked="" type="checkbox"/> Select
		Type Object	▼	<input type="checkbox"/> Select
Type of Apartment		Type of Apartment	▼	<input type="checkbox"/> Select
Material Construction	Brick	Brick	▼	<input checked="" type="checkbox"/> Select
Number of Floors		Number of Floors	▼	<input type="checkbox"/> Select
Floor	<input type="text"/>	from - until	<input type="text"/>	<input type="checkbox"/> Select
		not the first	not the last	
Area	<input type="text"/>	from - until	<input type="text"/>	<input type="checkbox"/> Select
Street	<input type="text"/>	Street	▼	<input type="checkbox"/> Select
House №	<input type="text"/>	House №:	▼	<input type="checkbox"/> Select
Building №	<input type="text"/>	Building №	▼	<input type="checkbox"/> Select
Price	<input type="text"/>	<input checked="" type="checkbox"/> from - until	2000000 3000000	Price <input type="checkbox"/> Select
	<input type="button" value="Clear"/>	<input type="button" value="Search"/>		


Figure 21 - Search an apartments


If at least one apartment answers the given conditions, it will appear in the search results, as shown in Figure 22.

The results are displayed in a tabular form. After selecting the desired Ad - opens a detailed description.

We can see the following information about the ad: location of the object on the map; application form, which includes a detailed description of the advertisement. The user fills in the fields of this form, indicating contact information for feedback, as well as a comment to the advertisement. Having filled all fields, the user presses the button "Send Admin". The application leaves in section "FeedBack", which we will consider further.

Выбор	5	Sovetsky	Secondary	2-room	Brick	9	4	78	Pushkina	67	1	2680000
-------	---	----------	-----------	--------	-------	---	---	----	----------	----	---	---------





Омск, Pushkina, 67

17 ▼ 73.387355, 54.974308

Yandex Maps

<pre> ID: 5 ID Area: Sovetsky Type Object : Secondary Type of Apartment: 2-room Material Construction: Brick Number of Floors: 9 Floor: 4 Area: 78 Street: Pushkina House №: 67 Building №: 1 Price: 2680000           </pre>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; padding: 2px;"><i>Name</i></td> <td><input style="width: 80%;" type="text"/></td> </tr> <tr> <td style="padding: 2px;"><i>Phone</i></td> <td><input style="width: 80%;" type="text"/></td> </tr> <tr> <td style="padding: 2px;"><i>E-mail</i></td> <td><input style="width: 80%;" type="text"/></td> </tr> <tr> <td style="padding: 2px;"><i>Comments</i></td> <td><input style="width: 80%; height: 30px;" type="text"/></td> </tr> </table>	<i>Name</i>	<input style="width: 80%;" type="text"/>	<i>Phone</i>	<input style="width: 80%;" type="text"/>	<i>E-mail</i>	<input style="width: 80%;" type="text"/>	<i>Comments</i>	<input style="width: 80%; height: 30px;" type="text"/>
<i>Name</i>	<input style="width: 80%;" type="text"/>								
<i>Phone</i>	<input style="width: 80%;" type="text"/>								
<i>E-mail</i>	<input style="width: 80%;" type="text"/>								
<i>Comments</i>	<input style="width: 80%; height: 30px;" type="text"/>								

Figure 22 - Search results

If the user wants to calculate the cost of the apartment - he can use the function of the program "Calculation of the apartment's cost".

To calculate the cost - the user must enter the characteristics of the apartment in the appropriate fields and click the "Statistics date" button, as shown in Figure 23.

This function is implemented on the basis of regression analysis, where the significance of factors influencing the total cost of an apartment was revealed and a model for calculation was constructed. This function is also available to any user.

## Calculation of the apartment's cost

Guest: elena

Kirovsky ▾	Secondary ▾	2 - room ▾	Brick ▾	Another ▾
X1 = 6	X2 = 2	X3 = 2	X4 = 1	X6 = 2
X5 = Area	75	X7 = Num. floors	5	
			Statistical data	
Cost per square meter		Y = 38985		
Full amount		Full sum = 2923911		

Figure 23 - Calculation of the apartment's cost

As a result of the calculation, we received 2 prices: the price per square meter and the total cost of the apartment. Above we described the actions of all users who can work with the above pages.

Use of the pages that we discuss below is only allowed to the employee or administrator. To do this, we need to log in with administrator or employee rights, but we'll look at how the program works with administrator rights, so any page and function will be accessible to us.

Above we wrote that the user can send an application for the apartment, which he chose. This application falls on the page "FeedBack", which is available to the employee and administrator, it can be seen in Figure 24.

We see that the page contains orders from buyers. For the answer, we need to select the required application, then the answer form will open. We can write an answer and send it to the mail to the buyer.



R. E. A. GALLEON [ Log In ]

Registration   Add Apartment   Search for objects   Feedback   Administrator   Employees   Calculate cost   About

## Send Mail

Admin: egor

	N_Floors	Floor	Area	Street	House	Price	Comment	Date
<a href="#">Выбор</a>	4	6	50	Ул. Строителей	10	100000	Hello Хочу купить квартиру	
<a href="#">Выбор</a>	23	1	30	1000	10	700000	Можно понизить цену	
<a href="#">Выбор</a>	4	6	50	Ул. Строителей	10	100000	Желаю купить	5/7/2017
<a href="#">Выбор</a>	10	7	300	Ул.Баграмяна	10	5000000	Цена 5000000 окончательная ?	5/7/2017

⏴ ⏵ ⏶ ⏷
✎ 🗑

<b>Name</b>	Ernest	<b>Phone</b>	010-77-61-36	<b>E-mail</b>	ernest@mail.ru
<b>Comments</b>	Цена 5000000 окончательная ?	<b>Message</b>		<input type="button" value="Send Mail"/>	

Figure 24 - Feedback page

In order to edit the record in the database, you need administrator rights. To do this, use the "Administrator" page, as shown in Figure 25. The administrator can view information and perform actions such as:

- view all ads available in the database;
- view the author of the advertisement (guest, employee or administrator himself);
- change of any announcement;
- change of the status of the apartment from "available" to "sold";
- sort ads by: the author, any ad parameter, by decreasing and increasing.

R. E. A. GALLEON [ Log In ]

Registration Add Apartment Search for objects Feedback Administrator Employees Calculate cost About

## Administrator Admin: egor

	ID	ID_Area	Apart	Floor	Area	Street	House	Price	User	Buy_Sell	Date
<a href="#">Правка</a> <a href="#">Выбор</a>	4	Kirovsky	1-room	12	221	Lenina	1	1231234	3. Guest: elena	<input type="checkbox"/>	01.06.2017
<a href="#">Правка</a> <a href="#">Выбор</a>	5	Sovetsky	2-room	4	78	Pushkina	67	2680000	3. Guest: elena	<input type="checkbox"/>	01.06.2017
<a href="#">Правка</a> <a href="#">Выбор</a>	6	Kirovsky	1-room	4	35	10 let oktyara	61	1680000	1. Admin: egor	<input type="checkbox"/>	01.06.2017
<a href="#">Правка</a> <a href="#">Выбор</a>	7	Sovetsky	2-room	7	74	Mira	82	2470000	1. Admin: egor	<input type="checkbox"/>	01.06.2017
<a href="#">Правка</a> <a href="#">Выбор</a>	8	Kirovsky	3-room	3	75	Stepanca	21	3200000	1. Admin: egor	<input type="checkbox"/>	01.06.2017

- [All](#)
- [Admin](#)
- [Employee](#)
- [Guest](#)

- [All](#)
- [ID Area](#)
- [Apart](#)
- [Floor](#)
- [Area](#)
- [Street](#)
- [Price](#)

Sort   Ascending  Descending

Figure 25 - Administrator page

The administrator needs to change the data about his employees from time to time. For example, a new employee was employed or the old employee was fired, then the administrator needs to add or delete the account information and provide the login and password for the entry, or delete. To do this, the Employees page was created, as shown in Figure 26. The administrator can add new users, edit information about them and completely delete the profile from the program, thereby blocking the access of the employee.

R. E. A. GALLEON [ Log In ]

Registration Add Apartment Search for objects Feedback Administrator Employees Calculate cost About

## Employees Admin: egor

Name	Dima	Surname	Rostov
Phone	933974873	E-Mail	Dimarostov@mail.ru
Login	drostov	Password	<input type="text"/>
<input type="button" value="Add"/>		<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

	ID	Name	Surname	Phone	E-Mail	Login	Password	Date
<a href="#">Выбор</a>	3	David	Petov	933987392	erika@mail.ru	Petov	1231	01.06.2017
<a href="#">Выбор</a>	4	Ivan	Ivanov	010-55-38-65	iv@mail.ru	Ivan		01.06.2017
<a href="#">Выбор</a>	6	Dima	Rostov	933974873	Dimarostov@mail.ru	drostov	qwerty	01.06.2017

Figure 26 - Employee page

# Chapter 7.

## Conclusion

In conclusion, we will talk about the work done, about the achievement of previously set goals and objectives, consider some comments on the work, and also describe the future work on the project.

The main goals of the work were to reduce the number of routine operations that the agency's employees commit daily and spend a very large amount of time on this work. We were able to create an information system in which the client can independently search for an apartment to buy or publish an advertisement on the sale of his apartment. This allows to reduce the time and money needed to collect incoming information. Previously, in order to publish an apartment, it was necessary to phone the client, make a meeting at the apartment, take a photo session, check all the documents, and only after all the above-mentioned actions, we could publish the apartment for sale. Now it is enough for us to process the application and appoint a meeting in the office to check the documents. The same goes for buying an apartment: before, the agency staff could spend a lot of time looking for an apartment for the client. And now he can independently find the option he is interested in and send us an application for viewing, indicating contact information and any comments. The next goal was to increase the efficiency of working with clients. We created a regression model for calculating the price of an apartment and integrated it into a website. This function positively influenced the time of working with clients, since now we do not need to make a report for the customer to convince him of the correctness of the specified price.

## **7.1. Problems**

Summarizing, we can also say about the problems of our information system. The information system does not have sufficient protection against unauthorized access. This is due to a lack of knowledge in this area. Also, the user interface should be more functional and convenient. This part of the work needs further work.

## **7.2. Future work**

We also have some plans for the future of this project, since the information system should not be limited to the set of functions that we have at the moment. In the beginning of the work we planned to create a parser that will collect data about apartments from other larger sites. We could not implement these plans for some reasons. First, the use of parsing is not entirely fair, because in fact the parses collect data from other sites without the permission of their owners. Secondly, the administrator can not verify the correctness of this data, because they are collected from a third-party source. This can affect the rating of the real estate agency, and may entail a large number of negative reviews about our product. Therefore, in the future, we will try to refine our system to a full-fledged advertising site, where our database will be the primary source of data. This will increase attendance, the number of transactions conducted with the help of our real estate agency, as well as increase the recognition of the company. For each type of user will be introduced the function of working with documents. This is necessary to speed up the process of publishing an apartment and its subsequent sale. So, together with basic information about their apartment, the seller will be able to attach the necessary documents that may be required for authentication. There will also be an electronic version of contracts between the agency and the client, which you can sign without leaving home. At the moment, the information system is ready for use and fully confirms the fulfillment of our goals.

## Bibliographic references

- [M 08] Matslyashek, L. Analysis and design of information systems with UML 2.0. Tutorial – M. Williams – 2008.
- [KP 02] Kremer, N. Putko, B. Econometrics. Tutorial - 2002.
- [W 08] Williams, I. Object-oriented analysis and design with examples. M. Phoenix - 2008.
- [V 07] Vendrov, A. Designing software for economic information systems. Tutorial - M. Finance and Statistics - 2007.
- [O 05] Ostrinskaya, L. Automated information systems and technologies in economics: Tutorial - SibADI - 2005.
- [C 08] Chubukova, I. Data Mining. Tutorial - University of Information Technologies - 2008.
- [R 15] Rasskazova, M. Econometrics. Tutorial - Omsk State Institute of Service - 2015.
- [B 15] Bogdanov, A. Comparative analysis of influence of various methods of accounting spatial information on the precision of models of estimation of value of residential property on the example of two rooms of Tomsk. Journal - Bulletin of Tomsk State University. Economy – 2015.



# **Attachment A**

# A Create a spreadsheet

```
DataTable table = new DataTable("Egor"); - creates a table named "Egor"
DataSet dataSet ; - Manages the database (reads, writes, edits)
DataColumn column ; - creates a table column
The initXML() function is called once when the page loads:
protected void initXML()
{
    column = new DataColumn();
    // We need to initialize it. The type of the column, we need to
specify the type of .Net
    column.DataType = Type.GetType("System.Int32");
    column.ColumnName = "ID"; // column name
    // Field autoincrement, and if AutoIncrement set to "true" and the
field type is not "int", then we get exception
    column.AutoIncrement = true;
    column.ReadOnly = false; // can not be changed
    column.Unique = true; // unique
    table.Columns.Add(column); // add to the table

    // further similarly for "ID_Area"
    column = new DataColumn();
    column.DataType = Type.GetType("System.String");
    column.ColumnName = "ID_Area";
    column.AutoIncrement = false;
    column.ReadOnly = false;
    column.Unique = false;
    table.Columns.Add(column);

    column = new DataColumn();
    column.DataType = Type.GetType("System.String");
    column.ColumnName = "Type_Object";
    column.AutoIncrement = false;
    column.ReadOnly = false;
    column.Unique = false;
    table.Columns.Add(column);

    column = new DataColumn();
    column.DataType = Type.GetType("System.String");
    column.ColumnName = "Apart";
    column.AutoIncrement = false;
    column.ReadOnly = false;
    column.Unique = false;
    table.Columns.Add(column);

    column = new DataColumn();
    column.DataType = Type.GetType("System.String");
    column.ColumnName = "Mat_Const";
    column.AutoIncrement = false;
    column.ReadOnly = false;
    column.Unique = false;
    table.Columns.Add(column);

    column = new DataColumn();
```



```

column.DataType = Type.GetType("System.String");
column.ColumnName = "Num_Floors";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Floor";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Area";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Street";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "House";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Building";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Price";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Mail";
column.AutoIncrement = false;
column.ReadOnly = false;

```

```

column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Phone";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "User";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

// new
column = new DataColumn();
column.DataType = Type.GetType("System.Boolean");
column.ColumnName = "Buy_Sell";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

column = new DataColumn();
column.DataType = Type.GetType("System.String");
column.ColumnName = "Date";
column.AutoIncrement = false;
column.ReadOnly = false;
column.Unique = false;
table.Columns.Add(column);

// Columns added to the table, now it's time to add the table
to the "DataSet"
dataSet = new DataSet();
dataSet.Tables.Add(table);

GridView1.DataSource = table; // bind the table to
"dataGridView1"
Read();

```