MEDICAL DATA MANAGEMENT USING BUSINESS ANALYTICS

Loredana MOCEAN
Miranda Petronella VLAD

Abstract. *The issue of data management has been felt in the last decade by more and more fields of activity, including medical one. The need to operate with large sets of information, determined the emergence of systems to help with the optimal development of processes specific to each environment. The project in question proposes a solution to this issue, following the stages of developing an information management system in hospitals. The software manages to solve the problem of patient administration, hospitalizations, analyzes, etc., in order to support the process that leads to the generation of the analysis bulletin and to allow it to be sent via email. With a user-friendly interface, the proposed project aims to facilitate the work of staff working with this system and add value to existing products on the market.*
Keywords*: Data, Clinical hospital, Management system, Administration, User-friendly*

SCIENTIFIC DESCRIPTION

The importance of the evolutionary process of humanity lies in the ability of interoperability that allows a symbiosis appropriate to all areas involved in it. Such an association is between the medical branch and the technology branch, the latter aiming at optimizing the procedures encountered in the former. In order to solve the problem of managing the large data sets that medical clinics face, today's medicine uses management information systems that manage and control data, but the unforeseen assault of the medical sector with a significantly increased volume of data is not excluded. To manage the effects of the ambush it has been subjected to, medicine still needs software to turn chaos and support conglomerates of data.

This paper addresses the issue of a system that solves the management of incoming and outgoing data in hospitals, aiming to add value through accuracy and speed, but also a new perspective on what already exists.

The main goal of process automation is based on the desire to eliminate human error and increase the speed of generating the desired result. If we refer to the current context, where more and more branches are adopting technology as a tool for optimizing manual work, the field of clinical hospitals is no exception, enjoying the benefits that such an approach entails.

This mechanism of adjustment to the needs of a clinical laboratory, has led to the emergence of a software called MIMS-Medical Information Management Systems, which support the development of the environment for which they were designed. The use of such a system reduces the occurrence of errors caused by the human factor and eliminates the possibility of confusion between the test in progress and its result, by implementing functionalities that track its path throughout the procedure. (D. O. Skobelev, 2011) Thus, at the intersection of medicine and technology, we find a generous number of applications aimed at information management in clinical laboratories. Existing Medical Information Management Systems software offers a number of benefits, which ensure their popularity among their users, but also have aspects that make it difficult to navigate the application flow.

This paper focuses on the realization of a user-friendly system, which should support the activity carried out and eliminate the aspects that can cause complications when performing medical procedures. A MIMS-type system must be constructed in accordance with the requirements of the segment for which it is dedicated, so that it can be adjusted and improved during its use, without affecting its functionality. Consequently, the antithetical link between a science completely dedicated to man and technology, which manages to automate manual labor, is a challenge that deserves to be taken on and that can open new horizons in both fields.

GOALS AND OBJECTIVES

The need to implement such a system was generated by the obstacles that appear in the field of medical hospitals, which without the existence of such software, at present, would risk an overload of the human factor. In order to be able to talk about the purpose of this project and the objectives it is trying to achieve, we must first take a look at the state of innovation present at national and international level in the field of activity.

The market for management information systems is presented to its users, with a wide range of possibilities, which will be tailored to the needs and shortcomings that the requirements of clinical laboratories have. Current applications provide certain functionalities that lead to a number of positive aspects, namely: reducing the number of errors of the human factor, data quality

and integrity, ensuring communication between the laboratory and other departments of the medical institution, easy access to information, storage of a large volume of data without the need for physical archives. On the other hand, there are situations that make it difficult for the user when using such a system, and this paper focuses on eliminating these unfavorable situations. If a web application has been implemented, the system will be dependent on the connection to an internet network and will be vulnerable to attacks of any kind. At the same time, both the information and the data backup will be stored by the system vendor, which can increase the maintenance costs of MIMS. In an on-premise application, the system will be independent when accessing information from another network, and it will have to be configured and deployed on each user's computer, which will lead to an extended installation time. In most cases, applications of this type do not have an interface that makes it easier to navigate the software, through a simplistic and minimal design, but rather have a loaded and suggestive graphics. Also, when displaying output data, the results of medical procedures performed in a hospital are often presented in a format that does not promote a clear understanding of the message they convey.

With a perspective on the systems currently operating on the market, this project aims to deliver a user-friendly finished product, easy to handle and allow the user to focus on the medical process they maintain, and not on solving the confusion that would arise in the case of implementing a system that is difficult to understand. The proposed project also addresses patients to the same extent, who will be able to view the results of the tests to which they have been subjected, in a clear, intelligible and accessible manner. Being a field that requires the organization to carry out a beneficial activity, the motivation for designing a management information system dedicated to hospitals, comes precisely from the desire to maintain and maintain the order of the data it faces. Although the purpose of the system is to streamline work in clinical laboratories, it is not only beneficial in it, but also in the departments and segments that depend on it. In a century in which the speed of information circulation breaks the spatio-temporal barriers, it is mandatory to belong to such software in the medical field, in order to increase the mobility of the circuit that each hospital follows.

Starting from this premise, several tasks can be decomposed which must be performed and which in turn will be decomposed until all aspects related to the proper development of the software are considered. Thus, we find in the list of secondary objectives the following:

- ✓ displaying the results in an intelligible format,
- ✓ entering valid data in the computer system,
- ✓ easy navigation through the application flow

- ✓ proper configuration of the system according to the requirements.

However, in order for the system to enjoy these aspects, it is necessary to address in more detail the tasks involved in the development process, such as:

- ✓ display the names of the persons who validated and worked on the input data,
- ✓ building a mechanism for validating the data entered,
- ✓ logical ordering of interface pages,
- ✓ modeling the database according to the real requirements,
- ✓ creating accounts for users, etc.

Once the objectives that form the hierarchical structure mentioned above are met, the team that develops the system can assess whether it responds appropriately to the purpose for which it was implemented, subsequently being delivered to potential customers.

THE CONCEPT OF THE PRODUCT AND TECHNOLOGY

The software developed according to the Medical Information Management Systems model was implemented out of the desire to meet the requirements of stakeholders and respond to them with a user-friendly application and easy to handle throughout the medical process, to which the analyzed tests are subjected. The direct beneficiaries of this system are certainly the employees of the laboratory, whose needs were the first arguments in the development of such software.

Therefore, the whole process of recording and processing data is subjected to a metamorphosis of manual work into an automatic one, assisted by MIMS, increasing the capacity of sample processing. Indirect beneficiaries, on the other hand, include both patients and hospital managers, each of whom feels differently the benefits generated by such a system. If we refer to patients, they can be confident that the medical tests are performed using high-performance, innovative tools and that the application will generate an analysis bulletin that is easy to interpret. In the case of managers, there are the advantages of a satisfied staff and patients who will again use the services of the clinical hospital, but there may be a number of risks that must be considered, such as: the possibility of not adapting staff to the new way of working, financial losses, the adoption of a system that does not meet expectations, etc. Following the bidding process, the main requirements that the software must meet in order to bring customer satisfaction have been outlined, and they can be structured in two

categories: functional requirements and quality requirements. Being the ones that dictate the way the system responds to the user, among the functional requirements are:

- The system must allow the user to enter data in the database: patients, hospitalizations, diagnoses, analyzes;
- The system must allow access to the data with which the user works;
- The system must differentiate the user-doctor from the user-assistant;
- The system will allow the validation of the results exclusively by the doctor;
- The system will generate the results in PDF format, after which they can be printed or sent via e-mail to reach the patient;
- The system will provide the user with personal details of his profile;
- The system will allow changing the password of the account that the user uses;

The functional requirements mentioned above, capture only a part of the set of needs that the customer reported during the development and that the system managed to manage in the best way. At the same time, a complex computer software does not only meet the criteria of actual functionality, but also aims to meet the quality requirements that an application must meet. These requirements refer to the operationality of the system and not to its behavior, captured through the functional requirements. Thus, the current application meets the following qualitative aspects:

- Data security and integrity;
- Flexibility;
- Performance relative to time and cost;
- Reliability;
- Easy navigation through the system flow;
- Installation with a low degree of difficulty;

Due to the fact that we want to work with a model that encourages collaboration between the client and the development team, but at the same time can easily adapt to unforeseen situations that may arise along the way, the Agile methodology was chosen, due to its transparency and focus on users. and business values. Being a model that is based on an incremental and iterative

development, the fragmentation of the project into sections and the separate operation with them, allow the approximation in advance of both the time spent in the actual development and the costs that the client has to bear. Agile presents itself as a response to traditional methodologies and brings a new perspective on the application lifecycle.

The three areas on which the software operates have been separated into specific tasks, giving developers the chance to identify and improve the problems that occurred, quickly and efficiently, without affecting the final product. Agile aims to achieve the objectives imposed by the requirements of stakeholders by dispersing the project into smaller pieces, which treated separately add efficiency and utility to the application. However, in order for the system design to meet the standards imposed by the customer, a hybrid model was used between the Agile-Kanban model and aspects of incremental and iterative theory.

Kanban, the specific Agile methodology, is used in software development due to the directions in which the project is heading, with the adoption of this method. Thus, it is encouraged to view the workflow by all parties involved, limit the tasks that are in progress and approximate the time in which they are performed. Applying these aspects and using the dashboard specific to the model, each developer knows his duties and tasks in the task, and what is sought is to prioritize communication within the team and identify obstacles that arise along the way. (M. O Ahmad, 2013) The current system was developed on the principles of this model, which accordingly complied with the form of customer requirements and the structure of the team that implemented the project. The small number of people involved led to a favorable dynamic between teammates, and this left a positive imprint on the finished product. Limiting the number of tasks according to the capacity of the team, helped to balance the delivery rate of product fragments and maintain a constant performance. The versatile nature of the methodology, favored the possibility of changes in the functionality and design of the application, during its implementation, as a result of feedback received from the customer. Thanks to Kanban, it was possible to notice an increase in both the productivity and the quality of the delivered products.

Therefore, the choice of the Kanban methodology was justified by the key factors, specific to it, which streamlined the teamwork and provided an overview of the whole process, throughout the development. This method not only ensured a good communication with the client, but also gave him satisfaction and improved the quality of the software, responding to the needs of stakeholders.

The above points out the importance of using the Kanban model, together with the flexibility and transparency it has and the advantages obtained from developing the project according to this pattern. However, in order to optimize and improve the functionalities of the system, an iterative and incremental approach was chosen, so the work is sectioned into smaller fragments, and then refined by repetitive cycles.

Regarding the software for data management in a hospital, this model offered the possibility to repeatedly test the implemented project sections, but also to update the functionalities to the point where they can provide the best user experience. If we refer to the login functionality, the first iteration only checks the existence of the user, the password corresponding to him and its correctness, in the database. In the following iterations, it was verified both the role that the user has-user or system administrator, and the position he holds in the hospital-doctor or nurse, to determine whether or not the possibility of validation of results will appear in graphic interface. Regarding the rest of the functionalities that the system has, the process went in the same way: the first iteration was presented with a minimum of activities that the user can undertake, and the following iterations developed the existing functionalities and added new ones. If in the early stages, the user could only add the patient and view hospitalization data, analysis histories and diagnoses, at the end of the project, he can generate an entire analysis bulletin and display it in PDF format or send it via e-mail the patient, view and modify their profile, or in the case of doctors, validate the analysis bulletins. The incremental and iterative nature of this model made it possible for the team to pay equal attention to each implemented sequence and also to test the completed work in several iteration cycles.

Consequently, the iterative and incremental approach helped to grind the project, until the finished product brought satisfaction to the customer. Opting for a hybrid model of Agile methodology, both Kanban and the iterative and incremental pattern, fluidized the entire development process, provided an overview of each stage the application went through, and by fragmenting the whole unit and perfecting each sequences, they extended the success to the level of the final result.

PROJECT IMPLEMENTATION

In order to correctly organize the input data, the system requires the construction of a database, which can be easily modeled, according to the rules that the system architecture implies. The software in question processes the information received from the perspective of the database, not

the storage files, thus producing a "system that will meet the information needs of the institution for which the software is developed." (Thomas M. Connolly, 2005) For this approach to be successful, developers must first consider database design, which is crucial in the application implementation process. (Thomas M. Connolly, 2005)

Starting from the premise that the start of software design is given by the structure and architecture of the database, it must be framed in one of the two categories that the theory of Database Management Systems (DBMS) proposes: the one of centralized systems and the one of distributed systems.

Although the affinity for a distributed DBMS is constantly growing in the IT market, due to the advantage of treating fragmented information as a whole in the company's entire network, the system in question was developed according to the format of centralized DBMS.

Thus, the option was folded according to the needs, requirements and equipment provided by the customer, as "centralized design is productive when the data entered is relatively small and can be processed by a simple database." (Peter Rob, 2009)

Compared to a distributed DBMS, in the case of centralized, the user can have a broad view of the database, which is local and unfragmented.

SYSTEM ARCHITECTURE

Taking into account the requirements and needs of the client, the development team opted to develop a desktop application that communicates with the server where the data with which it operates is stored and which is thus subject to the Client-Server architecture.

Under this approach, the "client" and the "server" are software rather than hardware entities that operate on the following principle: the client makes a request to the server, and the server responds according to the needs of the former. (John M. Gallaugher, 2007)

The data used in the implementation of the current work is stored in the local Microsoft SQL Server, and their retrieval by the other components is done through the development environment, Visual Studio. A first argument underlying the choice of this type of architecture is the speed with which such a system develops, giving the development team the opportunity to deliver the final product in a much shorter time, compared to other ways of working, possible less flexible. (John M. Gallaugher, 2007) Also, the client-server architecture has an advanced security, because the server has total control over the data, and the access of unauthorized persons is not

allowed. At the same time, if the technology according to which the server is developed undergoes changes over time, it is sufficient to execute these changes at the server level, without affecting the client application.

In order for the chosen architecture to fulfill its purpose within the application, the components of the system must be taken into account, which represent the raw material on the basis of which the Client-Server approach operates. Thus, within the workspace there will be several stations on which the system together with its architecture will run, each of them being equipped with the Desktop application and the centralized database server. The possibility for users to process data stored on the server is conditioned by the existence of a graphical interface and in the case of the system proposed in this paper, a design pattern that mediates the connection between the server and UI components, which use stored data.

The design pattern chosen in the development of the system in question is Model-View-ViewModel, which fits perfectly on the type of desktop application implemented. The structure proposed by MVVM involves the conceptual fragmentation of the software into three segments that communicate with each other, but that do not intersect their duties: Model, View and ViewModel.

The first segment, Model, is represented by the data server, which stores the information with which the system operates and which it modifies or not. The model does not process the information in any way, it only deals with keeping the data in the best form, so that it can be taken over by the other components of the pattern.

The View component refers to the graphical interface of the application and is the only segment with which the user has contact throughout the experience with the system. It presents the events that will be implemented by ViewModel, but also the links (data bindings) between the UI elements and the information in the database. Last but not least, the ViewModel segment represents the bridge between the components mentioned above, maintaining a clear separation between them. ViewModel can add information received from View to the database and can also display data from the server.

Therefore, establishing a structure of the system that suits all the characteristics of the chosen type of application, is essential for its dynamics to be good and prosperous. On the one hand, we have the Client-Server architecture that encourages the clear delimitation between the component that deals with structuring, organizing and manipulating the data involved in the process and the component that provides the user strictly with the information he needs. On the other hand, the system uses the MVVM design

pattern to ensure the granularity of the written code, so that each segment of it can be treated separately and benefit from proper maintenance, which ensures its operation over time.

THE PHYSICAL STRUCTURE OF THE DATA

In order for information to be translated into data useful to the software, it needs to be stored in a database, but not before structuring it into entities and outlining the relationships that are established between them.

Thus, the data categories become entities of the database, and subsequently the information will be stored in the tables related to it. At the same time, in the implementation of the application, according to the system architecture and the MVVM model, the information stored in the database will communicate with the system interface, all changes occurred at its level, affecting the data server.

The design of the database diagram started from the documentation on the set of information needed to be used and their division into appropriate entities. In order to capture exactly the aspects that interest us in relation to employees, patients, the set of analyzes, etc., each of the entities was assigned a series of specific attributes, which summarize the information necessary for the optimal operation of the application.

The storage of information in the database is done according to the type of data that the attribute established in the entity implies. Thus, the data can appear in the graphical interface in all the forms in which they can be displayed: strings, ints, data, boolean, etc. Segmenting the information into entities and establishing the relationships between them, offers a better perspective on how it should be organized, but also on how the functionalities are implemented in order to use that data.

It is essential that during the entire data manipulation process, they maintain their integrity and that their structure is not distorted in any way, in order to avoid generating errors that could lead to false-positive or false-negative results and that further influence the medical approach. The database must also fully comply with the diagram modeled for this purpose, in order to be able to easily identify which are the main actors with which the system operates and the relationships established between them.

SPECIFIC TECHNOLOGIES

This LIMS is based entirely on C # technology, which offers the possibility to design applications based on the Windows Presentation Foundation (WPF) framework, a subtype of the .NET platform. Being an object-oriented language, the application can be developed quickly, we are given the chance to reuse the written code, and the maintenance of the entire system is easy, compared to other existing language types. In addition to creating desktop systems, WPF brings with it a number of advantages in terms of structuring the application, each framework-specific window being built from two separate files, whose areas of interest are relatively opposite. Therefore, a clear distinction is made between the XAML file, which deals exclusively with the creation and stylization of the graphical interface and the CS file, which operates with the C # code itself and considers the back-end part of the application. Although different in terms of the language they use and the tasks they have in implementing the system, the two files can communicate with each other, using the MVVM model, described above.

Once the WPF window is created, there is a need felt for the existence of the data with which the two files can operate and which can either be processed or displayed. The database where the entire set of information specific to a clinical hospital is stored was built in Microsoft SQL Server, with which the application implemented in Visual Studio, communicates permanently. Database tables were not built manually, but using Code First specific Entity Framework (EF). This framework is a technology provided by the ADO.NET platform, which helps to map the relationships between the relational database and object-oriented programming. (Klein, 2010) When we use Code First, we should not worry about the structure of the tables in the database, but it is enough to implement classes specific to the objective ideology, which mimic these tables, which will later be created qith the help of EF in the local database. Each attribute of the implemented class will be transformed into a column belonging to the table and will respect the type of date which was declared in the C # code. In order to use the data stored in the database, we need to use queries to select the pieces of information necessary for the functionality we are implementing, and .NET has managed to simplify this with the help of Language Integrated Query. LINQ is an addition to the improvements developed by Microsoft, which make operating with information in the database much simpler and more intuitive. (Magennis, 2010) This library avoids writing SQL-specific queries, using simplified structures that make the code more readable and accessible to understand the mechanism being followed.

MVVM STRUCTURE

The model is the basic component of the design pattern, ensuring both the information needed to be displayed by View and the communication with the system database. In the case of the described application, the Model uses the Entity Framework (EF), to optimize the communication with the server and to build the data storage tables, through the Code First method. Therefore, based on this approach, the developer does not have to build tables based on SQL queries, but it is enough for him to model their structure in the development environment, in the form of classes, and they will be generated later with the help of EF . The relationships established between the tables of the database are developed by declaring an instance from the table with which it is desired to establish a relationship, within the class specific to the table from which the established link starts.

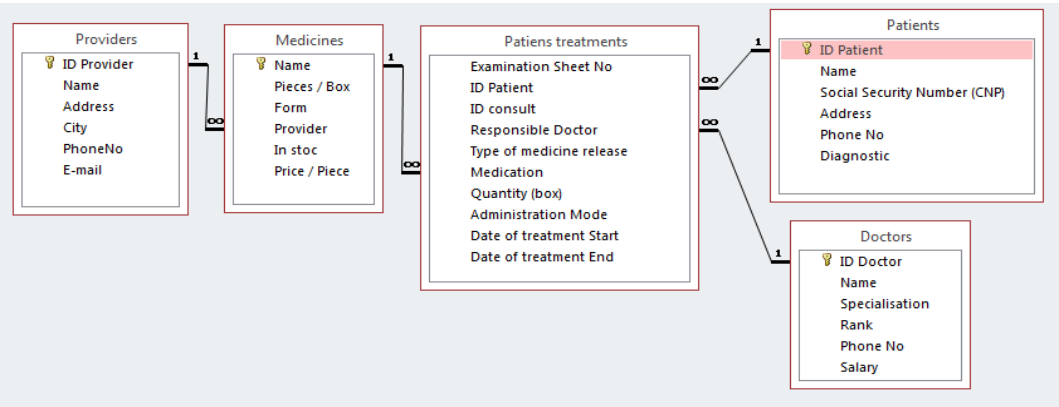

Figure 1. Database structure

The View component manages the graphical interface of the application and represents the bridge between the user and the developed system. View retrieves data from the Model using the ViewModel, giving the user the opportunity to process information from the database. Changes to the data level in the interface also spill over into the information on the server, as they are synchronized through data binding. Without setting security rules, any user of the application would be able to adjust the stored information. Thus, it is the duty of the system administrator to restrict users' access to modify sensitive data on the server, such as information about employees, users, etc. Also, depending on the position held by the user in the hospital, there are certain features that can only be used by doctors and for which nurses are not authorized.

The two components mentioned above give the software two of the aspects that make it a usable application for the customer, namely: database management, along with ensuring the integrity of information and the ability to operate with this data through the graphical interface. However, for communication and synchronization between the two components, it is necessary to use the ViewModel which acts as a link between them. Consequently, this level dictates the conduct of the View, by establishing connections with the Model. The current component has a dual purpose: to provide information and its specific processing operations at the interface level and to direct the logic and behavior of the View. (Anderson, 2012) Within ViewModel takes place the implementation of functionalities that are fully applied and contribute to the improvement of the work in the clinical hostpital. Thus, here we can see how the CRUD operations were developed, the generation of the analysis bulletin and its sending via e-mail and other properties that meet the customer's needs (see Figure 2).

Figure 2. Free-Mind diagram of the project development

CONCLUSIONS

Starting from results and notes managed in registers and journals, the growing volume of data currently used in medicine, makes it difficult to process them in this way and requires the use of management information systems, which have adapted to the requirements and operations performed. The application developed and described in this paper aims to optimize the processes in the medical field. The entire implementation followed a well-established plan, which originated in documenting the requirements and needs of the customer, the user who will operate in the future with the proposed system. As the software market that aims to solve the problems faced by clinical laboratories is constantly growing, the challenge has been to identify a new way for the current software to assert itself and satisfy the customer. Using the latest technologies and development environments, the application is presented to the user through a user-friendly graphical interface, an easy to follow flow and accessible to all users of the system, the ability to transmit test results via e-mail and many other features that reserve its place among the preferences of those working in the field.

Consequently, medical clinics have undergone the evolutionary process and have chosen to take advantage of what technology can offer them, in terms of optimality, speed and reduction of human error. Data management information systems today manage to store and process the entire data set with which medical laboratories operate, being a tool to combat the challenges of the unknown, which was, is and will be medicine.

References

Anderson, C. (2012). *Pro Business Applications with Silverlight 5., h*ttps://doi.org/10.1007/978-1-4302-3501-9_13

Berger, D. (1999). *A brief history of medical diagnosis and the birth of the clinical laboratory.* https://www.ascls-pa.org/uploads/2/4/2/1/24211033/meddiagandlab.pdf

Brian Sauser, J. R.-M. (2006, April). *From TRL to SRL: The Concept of Systems Readiness Levels.*https://www.researchgate.net/profile/Brian_Sauser/publication/228652562_From_TRL_to_SRL_The_concept_of_systems_readiness_levels/links/0c96051598e3c0b3b7000000/From-TRL-to-SRL-The-concept-of-systems-readiness-levels.pdf

D. O. Skobelev, T. Z. (2011). *Laboratory information management systems in the work of the analytic laboratory, Measurement Techniques.* 10.1007/s11018-011-9638-7

Darlene, B. (1999). *A Brief History of medical diagnosis and the birth of the clinical laboratory.* https://www.ascls-pa.org/uploads/2/4/2/1/24211033/meddiagandlab.pdf

John M. Gallaugher, S. C. (2007). *Choosing a Client-Server Architecture.* http://dx.doi.org/10.1080/10580539608906981

Klein, S. (2010). *Pro Entity Framework 4.0.*

M. O Ahmad, J. M. (2013). *Kanban in software development: A systematic literature review, 2013 39th Euromicro Conference on Software Engineering and Advanced Applications.h*ttps://www.researchgate.net/publication/260739586_Kanban_in_Software_Development_A_Systematic_Literature_Review

Magennis, T. (2010). *Linq To Objects Using C# 4.0.* Boston.

Oppel, A. J. (2009). *Databases: A Beginner's Guide.*

Peter Rob, C. C. (2009). *Database Systems: Design, Implementation and Management, Eight Edition.*

Thomas M. Connolly, C. E. (2005). *Database Systems: A Practical Approach to Design, Implementation and Management.*

NOTES ON THE AUTHORS

Loredana MOCEAN has graduated Babes-Bolyai University of Cluj-Napoca, the Faculty of Computer Science, she holds a PhD diploma in Cybernetics and she had gone through didactic position of assistant and lecturer, since 2000 when she joined the staff of the Babes-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration. She is associate professor. Ph.D., Babes-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration, Cluj-Napoca. Also, she graduated Faculty of Economics and Business Administration. She is the author of more than 20 books and over 40 journal articles. She is member in more than 20 grants and research projects, national and international. Email*:* loredana.mocean@econ.ubbcluj.ro

Miranda Petronella VLAD has graduated Babes-Bolyai University of Cluj-Napoca, the Faculty of Mathematics and Computer Science and she holds a PhD diploma in Cybernetics and Economic Statistics. She is associate professor. Ph.D., „Dimitrie Cantemir" Christian University Bucharest, Faculty of Economic Sciences, Cluj-Napoca. She is the author of numerous scientific papers and books in the field of computer science. Email: mirandavlad@gmail.com