

# Business Indicator Management System: How to Help the Companies to Manage Their Result with A Web Application

Paulo Roberto Martins De Andrade

Dept. Of Computer Science University of Regina Wascana Parkway Regina, SK, Canada S4S 0A2

**ABSTRACT:** The technology is growing faster each day, and the business needs to follow this growing to keep the competitiveness. The companies do it establishing metrics to aim their "Future View" using the Key Performance Indicators (KPIs). Typically, the KPIs give an overview of the health of the business, getting data from several different systems, like EAP, Commercial, Call Center and others. Normally this information is in separate databases and doesn't have communication with each other (data warehouse). The most companies use the Excel to put the consolidated data and present to managers and directors, to check where the company is today. The BIM System (or Business Indicator Management System) comes as a support for these works, putting all the information together and giving a visual information about the business health using graphs and live information. It can help in the decision making and the principles of the availability and reliability of information. This paper shows the design and implementation of a web-based application that makes it possible with two implemented design patterns (Observer Pattern and Factory Pattern).

**Keywords.** KPI, Design Patterns, ROI, Result Management, Software engineering, Business Metrics.

## I. INTRODUCTION

The KPI (Key Performance Indicator) is used to measure the processes performance of a company. It is a technique often used by executives and leaders in their management because it enables the communication of the company's development to the other employees who are below them in a hierarchical level. Through the results presented in the KPI, it is possible to quantify the performance of the company and allows workers understand how their activities collaborate to the success of these numbers. [1]

The KPIs help the managers to analyze and decide on a change of action if the numbers are pointing to a future bad scenario. When this occurs, company administration does a new scenario analysis and sets new goals, where each of these goals used KPIs, and allow an immediate monitoring in the course of actions. They provide information that facilitates the understanding of business progress. [2]

Even if the literature has already extensively explored the importance of the KPIs, especially when we talk about Business Intelligence, companies still resort to using spreadsheets in Excel to perform the analysis of these indicators. Despite being an effective solution, it is not efficient. It is a solution that becomes slow and requires some work to be produced. Also, as stated earlier, should information is available whenever necessary, one of the goals, engage employees so that they help achieve the business vision. [3]

A properly deployed IT tool can generate information on the strategic level, resulting in the reduction of people involved in the decision-making process, in the improvement of internal and external communication, in the improvement of organization's productivity. [18] [13]

We know that the information is something precious in a company, it is a factor that can determine the survival or even the discontinuation of its activities. Information technology can assist the company in making better use of information. The best way to it is with the development of a robust web system that can provide the necessary support in the monitoring and control of strategic and performance indicators [18]. As a solution, it proposes the creation of a new system that aims to fill this gap in existing companies, thus simplifying management. The new system, called BIM System, will work as a part of the Joomla Framework (a Content Management System – CMS, available in <http://www.joomla.org>).

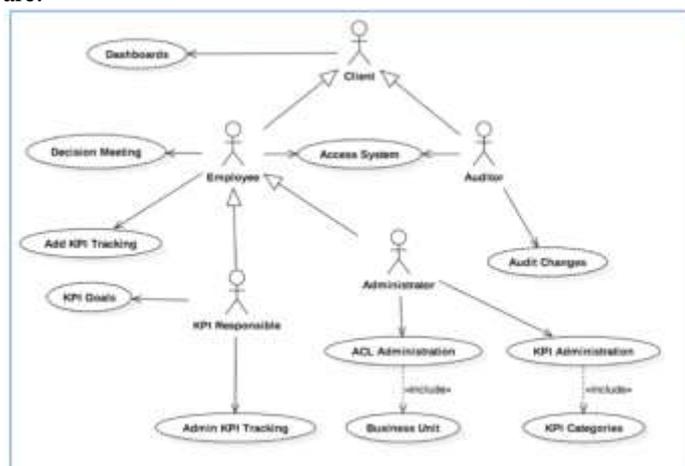
We organized the rest of the paper as follows. Section II shows details about the requirements. Section III presents the alternative solutions, describing what the system brings new compared to existing systems. Section IV shows the architectural design and patterns used in the software. Section V shows some details about the development. Section V also give us details about the deployment. Finally, Section VI offers some concluding remarks and outlines our future work.

## Requirements Specification

Currently, the majority of software have a very high complexity, which creates the need to define precisely the relationship between the various components as well as the internal operation of each of the components constituting the software, so that no failures. In this context, the specification languages by allowing to define precisely the mathematical problems to which the software is expected to respond, allow to create software with high reliability.

After the observation of some companies in Brazil and Canada, we verify what is the existents gaps in the business management and what kind of services will be necessary to change for a web-based application. For this work, was performed a short interview with some managers to ask what kind of functions will be required for a system that aims to manage all KPIs of the business and give tools to make the meeting more productive [1].

The system requirements were defined to supply all the company's need for information organization, agility, availability, and security. The use case diagram illustrated in Figure 1 summarize the functional requirements of the software.



**Fig. 1.**Summary Use Case Diagram

**To facility the general comprehension of the system, we can use some general definitions:**

- KPI: Key Performance Indicator – Main object.
- ACL: Access Control List – Manage users.
- CRUD: Create, Read, Update, Delete.
- KPI Administration: CRUD for KPIs.
- ACL Administration: CRUD for users.
- The client is everyone that can access the system.
- Employee and Auditor also are Clients. Administrator and KPI Responsible also are Employee.

### 1.1 Functional Requirements

#### Client

- Screen with details of one KPI, all values (months) and details of each month (decisions meeting).
- KPIs Dashboard: one dashboard with all marked (choose in the creation) KPIs to make easy the visualization (for everyone)

#### Administrator

- Access Control List (ACL): Just authorized people can add, edit or delete data. Administrator is who can add, edit and remove users from the system (access control)
- KPIs Database: Possibility to create, to read, to update and to delete KPIs (CRUD)
- KPIs categorization: Possibility of divide the KPIs and categories and subcategories
- Business Unit: Manage units to put users and KPIs

#### Responsible for KPI

- KPIs Database: Possibility of edit, read, enable and disable KPIs
- Each KPI has a responsible (role) and all the editions needs to send by email (workflow).
- KPI Goals: set what value is the goal of each KPI for each period of time.

#### Employee

- KPIs Values: Add values for each month of the KPIs (tracking), and keep tracking with the possibility to compare with others months or with the same month from the previous year.

- Decision meeting: Have meetings to discuss the results and after, register the action plan for each KPI (Facts, causes, and actions)

Auditor

- Possibility to view all changes in the information of the system using a login. The auditor screen includes the options to check who made the changes, when the changes occurred, filter changes by type, check previous values, and check new values.

## 1.2 Non-Functional Requirements

- **Portability:** Works on desktops, tablets and smartphones
- **Internationalization:** the system needs to work in English and French
- **Easy:** Tips around the system to help the users

## II. COMPARISON WITH ALTERNATIVE SOLUTIONS

To show the benefits of this new system, we compared the initial requisites and concepts of the system with some alternatives found on the web. The first option for the system found was the "ICG KPIs Management" from the Qualitin Company (<http://www.qualitin.com/en/solutions/icg-kpis-management>). The ICG system delivers the "Strategy Map" that is not foreseen in the first version of the BIM System. This functionality is not very useful because for now because it is a part of the Business Strategic Plan and usually, can be accessed quickly. Maybe the company decided not to put all the indicators in the system for any reason and the "Strategy Map" will be incomplete. Another point is that the ICG don't have the role of Indicator Responsible and because that, the information might be not audited before someone update the indicator. In the same way, the ICG don't have one workflow behind the system.

The second alternative is the "SimpleKPI" from Iceberg Software Company (<http://www.simplekpi.com>). This system has a very friendly design and is very easy to use. You can easily change between different KPIs and update more than one KPI at the same time. But it has some disadvantages when we talk about system and quality requirements. In this kind of software (and others like ActPoint KPI, etc.) we don't have the total control of the system. Without this control, it is not possible to give any guaranty about the uptime, capacity or extensibility. All the system is hosted outside of the company and we, as a client, don't have access to the code or the server. It is not possible to create new functionalities or extend an existent one. The SimpleKPI also don't use the idea of the KPI responsible. If the company needs this ability, this system isn't a good idea.

CallCopy [4] delivers information starting on the beginner. They give us information about the importance of the KPIs and also give some examples of Use Case to help in the creation of KPIs. They describe the cycle life of the KPI: collect from enterprise data → analyze → report/dashboards → engage. In the paper, we can check some critical requirements for one system of KPIs like the role-based views of operation (help to see just what is important to you and not all the indicators at the same time) and the idea of individual KPI (just applied to few companies). The author complements talking about the best practice in this kind of system: incorporate data from all relevant source, embrace adoption at all levels of the organization and drive quality through a process of continuous improvement. In the another hand, the paper doesn't give the idea of workflow, necessary for this system or the idea of categorization, for example, the possibility of the creation of different indicators for the same department, but divided in sub-department. In this way, we can have the division of company → directory → department → local supervision.

## III. DESIGN SPECIFICATION

### 3.1 Architectural Style

For the architectural style, we choose the Model–view–controller (MVC). It is a software architectural pattern mostly (but not exclusively) for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. The system was implemented using the framework Joomla, and that choice has influence in the style.

With MVC you can:

- Develop a system with **loose coupling** and **high cohesion** that supports high reliability, robustness, and functional correctness.
- Hide the complexity of a system to the end user.
- Avoid code monotonicity and duplication of objects.

#### IV. DESIGN PATTERNS

Design patterns are common reusable solutions for an existing problem in the context within a software design. They are a way of structuring your code which often acts as an intermediate between a programming paradigm and a concrete algorithm.

Design patterns are crucial in the phase of software project development as it eases to achieve the target within the deadline and also design patterns techniques reduce code complexity by promoting reusability. Apart from its list of advantages, it also provides an efficient platform to realize the relationship between different entities in the software project.

In this project, we have used a factory design pattern and an observer design pattern. We described the details in the following section below.

##### 4.1 Factory Pattern

It is a creational pattern where objects can be created without exposing the creation logic to the end user and refer to the newly created object using a common interface. It is one of the most used design patterns in complex architectural styles.

Creating an object usually requires complex processes not appropriate to include within a composing object. The object's creation may lead to a significant duplication of code, may require information not accessible to the composing object.

The factory pattern handles these problems by defining a separate method for creating the objects, which subclasses can then override to specify the derived type of product that would be created. Figure 2 illustrates this pattern in the BIM System.

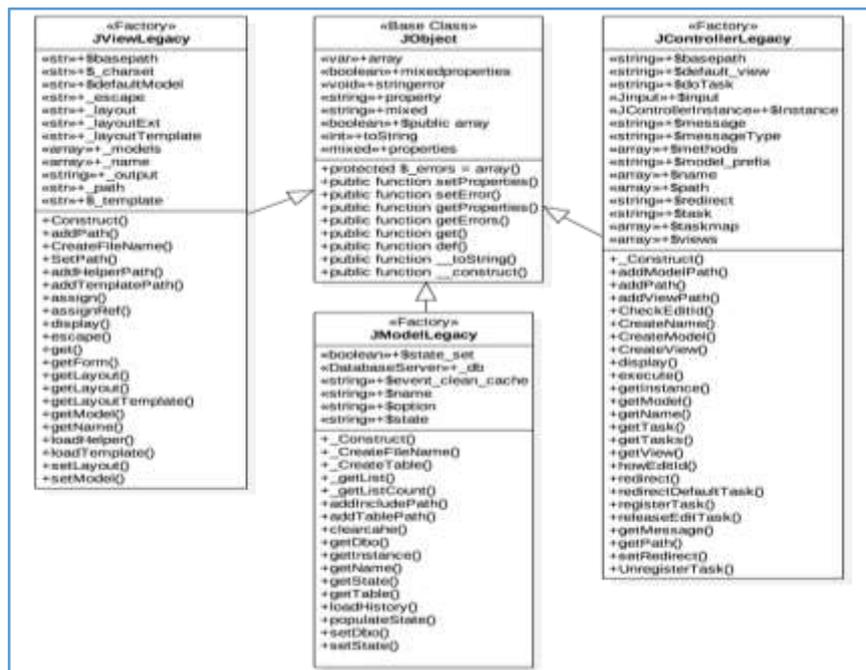


Fig. 2. Class Diagram in Bin System for the Factory Pattern

##### With this pattern you can:

- Develop a software system where requirements are to create many objects according to the need of the problem, but without exposing the creational logic to the client or the end user.
- Present the newly created object to the end user only through a standard interface.
- Avoid duplication in the objects created.

##### How the implementation works:

- Location: All the Models, View, and Controllers.
- Factory pattern in our BIM system is implemented using extend function in PHP. We used this kind of implementation to inherit properties from the base class to the subclass. It also helps to provide a common interface to the newly created object to the end user. In the source code, it is possible to observe that **JControllerAdmin** and **JControllerForm** are the base classes for all the controllers in the BIM System (lists and items view). Both classes extend the general class **JObject**.

**Implementation Guidelines:**

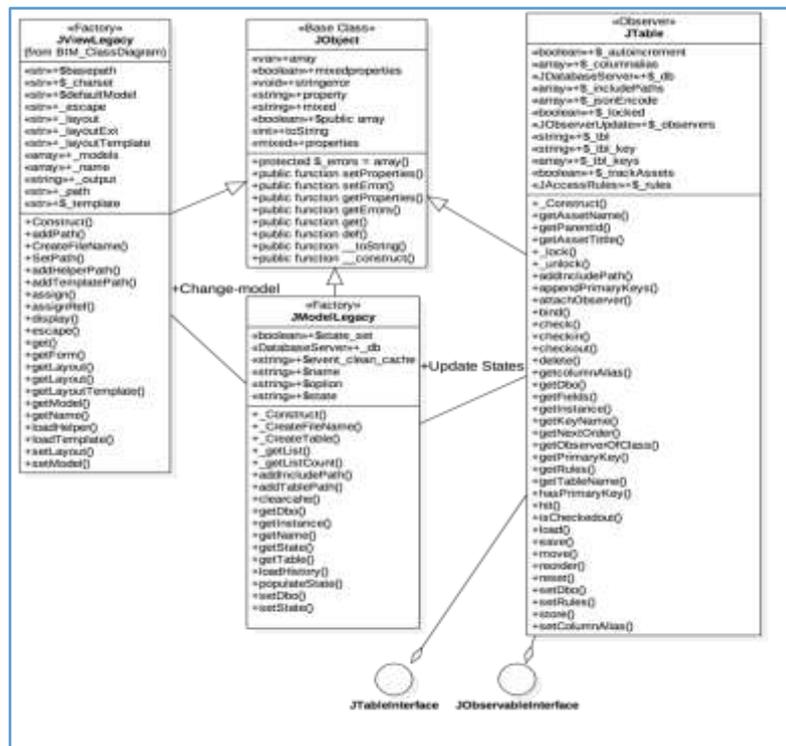
1. Create a base class.
2. Declare base class variables.
3. Create functions of the base class.
4. These functions should describe the property settings, error function definitions and also should be able to get these property definitions.
5. Then, create sub-classes that inherit these properties from the base class using the function "extend".
6. Initial inherited class acquires properties and functions from the base class.
7. Apart from that, we can create new functions the will be used for this new kind of object (class). These sub-classes can be the final layer or the intermediate layer or another sub-class.
8. Final layer class is then created and it will inherit the functions for all other previous classes.
9. In this class, we create new functions or override existent ones for a specific object. The functions and attributes for previous classes can be called using the object \$this (self-referenced) or using the command parent::functionName().

**4.2 Observer Pattern**

The observer pattern is a software design pattern which defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically. The interface class is used in the subject since it provides the considerable promise to provide the required functionality.

It is optimal to create a particular function in a system in which an object, called the subject, maintains a list of its dependents, called observers. This implementation reduces the monotonous function of individual updates when dependencies become more than one to a particular subject. It is also very helpful in supporting loosely coupled designs in interacting with the objects.

In a scenario where maintaining one-to-many dependencies between the objects become an integral part of a development process, such that a single notification from one state (presumably subject-object) notifies the changes to all the observers of that state (presumably observer object), this pattern is the best option. We required it in our system in several cases. A simple instance where an admin user (subject-object) creates or updates a KPI or Business Unit values and the system should be designed such that this notification successfully reaches its dependents i.e. to all the non-admin users (observer object) who have access to that respective KPI in which modification or creation is made. Figure 3 show us how is organized the observer pattern in the BIM System.



**Fig. 3.**Class Diagram in Bin System for the Observer Pattern

**How the implementation works:**

- **Location:** plugins and table folder
- Observer pattern in our BIM system is implemented using extend and implements functions in PHP. This is helpful in letting the publish function know of the subject observer and then object observer dependencies.
- In the source code, it is possible to observe that JTable as the base class through which their properties are inherited through extending to subclasses used. The publish function in parallel checks of observer objects in the subject-object through \$this->\_db->execute(). This class extend the general class JObject and implements two interfaces, the JObservableInterface and JTableInterface.

**Implementation Guidelines:**

1. Create a base class.
2. Declare base class variables.
3. Create functions of the base class.
4. These functions should describe the property settings, error function definitions and also should be able to get these property definitions.
5. Now, create an Interface that will be the observer for the tables / models (generalization).
6. Along with them, we declare a public function to return observer to the created observable interface.
7. The next step will be to create a class that implements the interface and extend the base class.
8. Now is possible to create classes (Tables) for each kind of object (sub-classes)
9. In MVC, the model makes procedure calls to the tables to do the database initializations and to perform the table check to map subject object and observer object relationship. Normally used inside of the models classes.
10. Final layer class is then created and it must be able to inherit DB Checks and table initialization of its immediate upper class.
11. In this class, we create an object that calls the publish function that implements notification of subject observers to object observers.

**V. DEVELOPMENT AND DEPLOYMENT**

**5.1 Minimal Requirements**

This software was designed to be used on the web, and there are some minimal server requirements (Table 1) for this application works well.

**Table 1.**List of supported software

Software	Recommended	Minimum
PHP (Magic Quotes Off)	5.6+ or 7.0+	5.3.10
Supported Databases		
MySQL (with InnoDB)	5.5.3+	5.1
SQL Server	10.50+	10.50
PostgreSQL	9.1+	8.3.18
Supported Application Servers		
Apache	2.4+	2.0
Nginx	1.8+	1.0
Microsoft IIS	7	7

The system is also browser friendly. In the other words, it works in several web browsers, like Safari 6+; Google Chrome 15+; Mozilla Firefox 10+; Internet Explorer 9+; Android Browser 4.0+.

**5.2 Summary of development**

We created this system as a part (a component) of Joomla Framework. In this way, we can easily install the result component on any website or intranet that use the Joomla Framework with the minimum version 3.0.

- Total of Classes developed: 58.
- Total of Lines for the classes: 9569 lines.

**5.3 Deployment Diagram**

The Figure 4 illustrates how the system was divided to be a distributed system. In the server view, the system was hosted by the company Hostinger with a server located in the United Kingdom. The application server is located on a different server of the database. Also, the SMTP server used is located at the University of Regina.

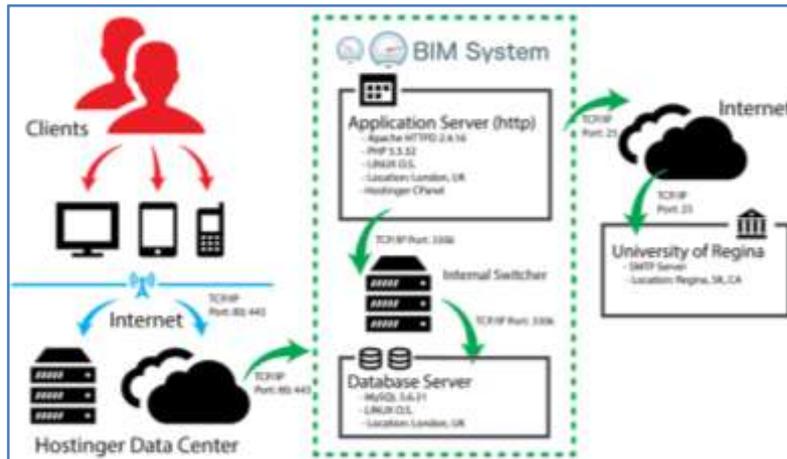


Fig. 4. Deployment Diagram – Server View

### 5.4 Perform Test

Performance testing is a crucial part of a quality process and software testing in a Web application. In most software engineering process models, software testing plays a significant role in ensuring the quality of a software product. In most projects, most of the effort goes into functional tests and some managers tend to ignore Performance tests altogether.

For websites and web applications, especially an e-commerce situation, Performance tests are fundamental. Even a bug-free application will be doomed to failure, only "endure" a medium traffic, but is not able to handle a significant spike in real life of a situation. To ensure that a Web application meets certain criteria such as performance, data throughput or response time, testing in an environment similar to production will be necessary.

Because of this importance, the BIM System also passed for some of these tests. For the below observational performance results, WEB Page Test [10] were used and give us the results in Figures 5 and 6.

	Document Complete						Fully Loaded					
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	3.052s	0.531s	4.918s	499	117	0.052s	17	889 KB	0.234s	18	891 KB	\$5
Repeat View	1.334s	0.395s	1.406s	1431	117	1.304s	1	3 KB	1.304s	1	3 KB	

Fig. 5. Web Page Test results - Loading time

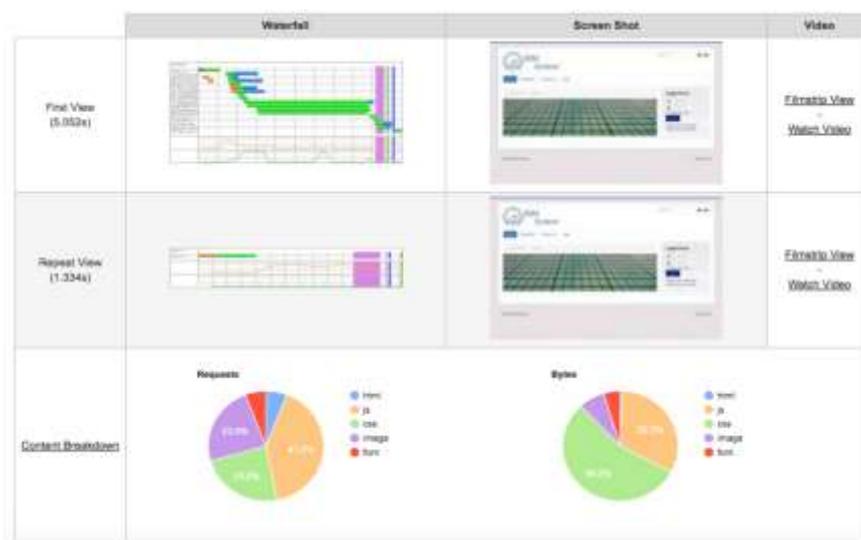


Fig. 6. Web Page Test results - Details about the loading

### 5.5 User Interface

According to ISO/IEC 9126, "Usability refers to the ability of an application to be understood, learned, used and attractive to the user, specific conditions of use" [11]. Usability is a concept that allows us to evaluate

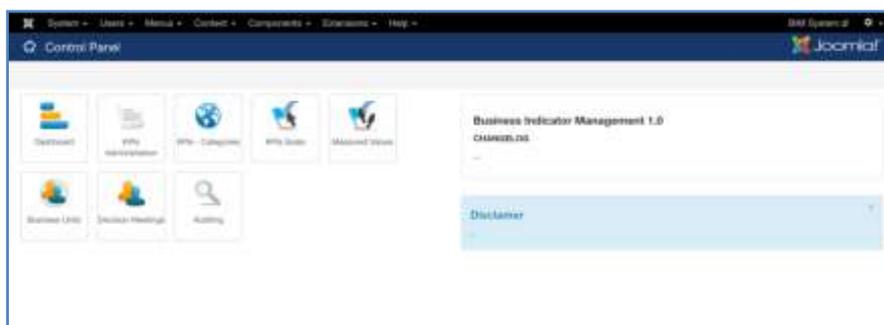
factors that influence the use of the application, for example, Learning facility; Ease of use; Storage facility; Use of safety; User satisfaction.

The interfaces with good usability cause the user to become more productive, as he does not have to stop all the time their work to seek further guidance. The interface provides this guidance itself or, ultimately, the application's help system does this job. But often the user is forced to deviate from their jobs and seek in the external environment with the necessary skills to perform their tasks. This necessity is due to the application usability errors, which is not friendly and confuses the user.

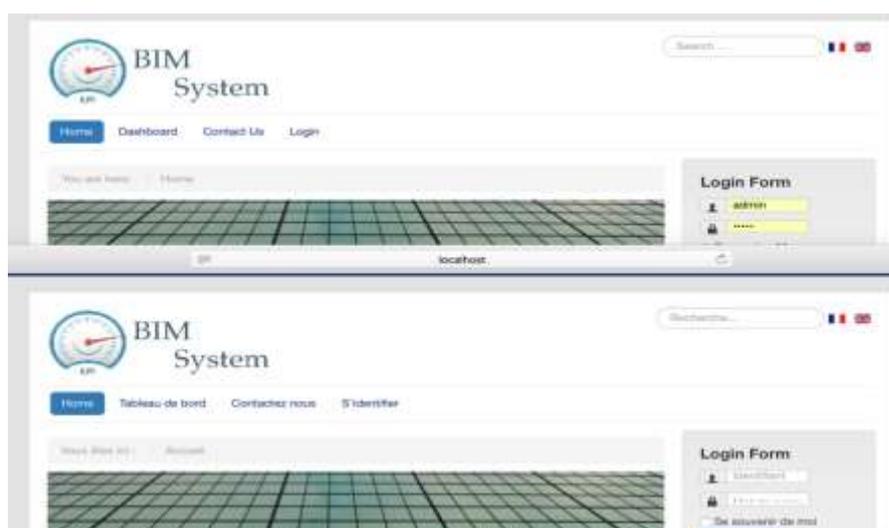
In the rest of this section, we have some screenshot from the system that shows the usability tests. The test is for auto-expiation, browser friendly (same behavior in different browsers), dual language (to improve the number of people that can use the system, it was available in English and French) and mobile friendly (if the system can be used in the mobile device without loss information).



**Fig. 7.**Main Administrator Screen



**Fig. 8.**Main BIM System Screen



**Fig. 9.**BIM System's home page in English and French

## VI. CONCLUSION

To a company have success in a globalized world is necessary to define the actual business picture and the plan for the future. To check if the company is on the correct path to achieve the goals is a need to check and update every time the actual status of the business. They do it using the KPIs. With the rise of technology and its intensive use, brought agility to the processes of all size companies, give better possibilities to get information

when it is necessary to security, privacy, and trust. Typically, in the companies, we have a different system with different information about the business performance and one system to consolidate this information is necessary.

To develop this system quickly and efficiently will be needed to specify all the system requirements correctly using the use cases and the design patterns. The goal of this paper was to provide enterprises with objective elements to conclude that is possible improve the process of decision-making and KPI evaluation. We reached this aim after our analysis and design of one functional application that makes it possible. Another point achieved with this paper is that now the organization has a general view, general requirements and one general how-to for implementing their system. Also, the developed system here presented, is a generic system that can be used for any organization and because of the MVC and the design patterns, the system is easily extensible.

One feature notable for the system is that because of the use of MVC, you can easily change the part for the KPI measured. Today, the user needs to input this information, but you also can modify the model to get this information from the internal Big Data already exists in the company or to consume web-services from another system and get the actual information. With this functionality all the KPIs from the business can be centralized in just one system, thus giving more agility to board meetings.

### **ACKNOWLEDGEMENTS**

The CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico - "National Counsel of Technological and Scientific Development") supports this work.

### **REFERENCES**

- [1]. H. Hao, K. Zhang, S. Ding, Z. Chen, Y. Lei and Z. Hu, "A KPI-related multiplicative fault diagnosis scheme for industrial processes," in 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, 2013.
- [2]. Institute of Management Accountants, "Tools and Techniques for Implementing Integrated Performance Management Systems," Statements on Management Accounting, 1998.
- [3]. D. Parmenter, *Key Performance Indicators: Developing, Implementing, and Using Winning KPIs*, Hoboken, NJ: Wiley, 2015.
- [4]. CallCopy, "Performance Management KPI Dashboards: Using Big Data to Align Enterprise Objectives," 2013. [Online]. Available: <http://www.frostftp.com/JE/Events/CC/Callcopy/3w.pdf>. [Accessed 27 Jan 2015].
- [5]. B. BRUEGGE and A. H. DUTOIT, *Object-Oriented Software Engineering: Using UML, Patterns, and Java*, 2nd Edition ed., Upper Saddle River, NJ: Pearson, 2004.
- [6]. MKLab, "StarUML," 18 Jan 2016. [Online]. Available: <http://staruml.io/>. [Accessed 04 Feb 2016].
- [7]. R. S. Pressman, *Software Engineering - A Practitioner's Approach*, 4th Edition ed., McGraw-Hill, 1997.
- [8]. G. J. Myers, *The Art of Software Testing*, New York: Wiley, 1979.
- [9]. B. Boehm and V. R. Basili, "Software Defect Reduction Top 10 List," *IEEE Computer* 34, pp. 135-137, January 1991.
- [10]. WEB PAGE TEST, "Web Page Test," 2016. [Online]. Available: <http://www.webpagetest.org/r>. [Accessed 14 March 2016].
- [11]. ISO, "ISO/IEC 9126-1:2001," 13 September 2012. [Online]. Available: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=22749](http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749). [Accessed 14 March 2016].
- [12]. M. Seify, "Importance of KPI in BI System, Case Study: Iranian Industries," in 7th International Conference on Information Technology: New Generations (ITNG, Las Vegas, NV, 2010).
- [13]. P. R. M. Andrade, F. A. Silva, A. B. Albuquerque and W. D. Teofilo, "Change Management: Implementation and Benefits of the Change Control in The Information Technology Environment," *International Journal of Computer Science and Information Technology*, vol. 6, no. 1, pp. 23-33.
- [14]. P. R. M. Andrade and A. B. Albuquerque, "Project office: Features, advantages and planning their implementation in the public sector," *RBGP: Revista Brasileira de Gestão de Projetos*, vol. 11, pp. 2126-2134, 2014.
- [15]. P. R. M. Andrade, A. B. Albuquerque, O. F. Frota and J. F. Silva Filho, "PM5: One approach to the management of IT projects applied in the Brazilian public sector," in *Proceedings of 13th International Conference on Software Engineering Research and Practice - SERP*, San Diego, 2015.
- [16]. P. R. M. Andrade, A. B. Albuquerque, R. G. Araujo, J. Cronemburgue Filho, T. R. Pereira and N. C. Mendonça, "Improving business by migrating applications to the cloud using cloudstep," in *Proceeding for the 29th International Conference on Advanced Information Networking and Applications Workshops*, Gwangju, 2015.
- [17]. A. Herden, P. R. M. Andrade, A. B. Albuquerque and P. P. M. Farias, "Agile PDD - One approach to software development using BPMN," in 11th INTERNATIONAL CONFERENCE APPLIED COMPUTING 2014, Lisbon, 2014.
- [18]. P. R. M. Andrade, O. F. Frota, J. F. Silva Filho and M. A. N. Morais, "Using a mobile app to improve performance of management of services orders and teams processes," in *ACM International Conference Proceeding Series*, New York, 2015.