Customer Relationship Management in University Environment

João Bruno Trindade Albuquerque Ala de Rezende
joao.rezende@tecnico.ulisboa.pt
Instituto Superior Técnico - Taguspark

ABSTRACT

The management of the relationship with the customer plays an important role in all companies, regardless of the business area in which the company operates. When applied to the university environment, the Customer Relationship Management (CRM) focuses on the relationships that students develop with the university, and how this relationship can be managed by the university with the goal of increasing the success of students throughout their academic career. We propose the implementation of a CRM application and parallel systems to allow Professors in the Department of Computer Engineering at Instituto Superior Técnico to develop a closer management of their relationship with students in the courses ministered by them.

This article describes the motivations and what was the problem identified, leading to the creation of a proposed solution and its architecture to solve the refered problem. Subsequently it is described the steps taken to make possible the implementation of the solution that has been previously proposed and an evaluation of the results achieved, always taking into account the objectives outlined and the proposed solution. Finally, conclusions about the entire process are drawn, an improvements/changes are suggested for future project iterations.

Keywords

CRM, Professor-Student Relationship, CRM Applications, Segmentation, Profiling

1. INTRODUCTION

The CRM initiatives and applications have gained an increasing relevance in the development of the companies businesses in recent decades.

With the need to present distinctive features and a place in customers preferences, many companies felt the need to treat their customer relationship with a degree of differencing that made the customer feel special without losing the efficiency of processes implemented in the company. This led to the CRM methodology growth, that advocates the creation of a vision focused on the customer and the idea of creating a one-on-one relationship with the customer started gaining ground to other methods of customer relations management. For this methodology to thrive, the necessity to created applications that support this way of thinking and acting within the companies arose.

The Customer Relationship Management is a system that allows information collected from interactions with customers to be stored, grouped and manipulated so as to create a compilation of the relationship developed with the customers, thus creating a client profile as complete as possible. From the moment that a profile of the customer is created, it becomes possible to perform customer segmentation to create targeted campaigns, it becomes possible to create privileged communication channels, among other features related with the management of the relationship with the customer.

1.1 Motivation

The objective of this article involves understanding how a CRM initiative can assist in the way that Professors at DEI support and interact with the students attending their courses.

The proposed system aims to make it possible to incorporate the personal information of students and other data relevant to the monitoring of their courses throughout the semester, such as their attendance at the classes. Furthermore, it is intended to give the possibility to create sub-sets of students based on the analysis of the collected information, thus creating the possibility of managing communication channels for custom identified targets. In particular, the main contributions of this article are:

- Study on the business processes in place at IST, as well as what the real needs that can be addressed with a CRM initiative.
- A compilation of information about CRM and successful CRM implementation initiatives in the corporate environment, as well as in academic environment.
- Analysis of the adequacy of existing CRM applications in the market to a very specific type of business (education and university environment), that has its own business processes.

- Design and development of an application to perform the collection of attendance in classes of students of DEI, that at the same time may be connected to the CRM application.
- Implementation and adaptation of existing CRM solution on the market to the reality and needs of the academic Department DEI.
- Set of scenarios and tests to demonstrate that the use of the application allows to perform the activities identified more quickly and accurately.

1.2 Problem Statement

IST is a college of Engineering located in Portugal. The organization of the University follows a classical model, and its main division is made by Departments.
Departments offer courses to the various Bachelors/Masters, which are taught by Professors. At the beginning of the semester students have the possibility to enroll in one or more courses, and a Professor can in turn be associated with one or more courses.
During a semester several classes are taught in each course, and there is one or more time points of evaluation of the knowledge gained during the classes. Recently, one of the Department of IST, the DEI, decided it would be beneficial to collect the attendance of students in classes, even though the class attendance is not mandatory by the School regulations.

Based on an empirical observation, which was recently supported by statistics, it is possible to observe that the rate of abandonment of the academic path and the failure rate in courses is significantly high.
Thus, the major problem identified is that there is a dropout and absenteeism rate that is significantly high, and that there are no tools available that allow the various responsible elements of the School to timely identify risk behaviors that may lead to abandonment, and to interact with groups of student in a closer and more personal way.
To tackle this problem it was identified an application that aims to make a management of the relationship with the customer in the business environment, the CRM.

Although used in abundance, it is not simple to find a commonly accepted definition of CRM. The difficulty in finding a broader definition arises from the multiple and different motivations that companies have when implementing a CRM initiative, which leads to distinct ultimate goals. [2, 4, 5]
One of that definitions is presented by Atul Parvatiyar and Jagdish N. Sheth [9], and possibly is the one that better fits the notion of CRM:

Customer Relationship Management is a comprehensive strategy and process of acquiring, retaining, and partnering with selective customers to create superior value for the company and the customer. It involves the integration of marketing, sales, customer service, and the supply-chain functions of the organization to achieve greater efficiencies and effectiveness in delivering customer value.

Thus, it was concluded that with the implementation of a CRM application, and with the aid of an application dedicated to the collection of attendance of students in classes, it would be possible to make the profiling and segmentation of the students of DEI, giving Professors a tool that allowed them to make the identification and closer monitoring of students that attend their courses.

2. PROPOSED SOLUTION

The proposed solution requires the implementation of SugarCRM Community Edition, which is a free and open-source CRM application, and also the implementation of a presence collection application that feeds SugarCRM with data that enables the profiling and segmentation of students. Given this premise, the presentation of the proposed architecture is made individually for each application, being also presented how the two applications are connected and how they interact with FenixEDU API.

2.1 Attendance Collection Application

The main objective of this application is allowing the collection of attendance of the students, and the subsequent query of these presences and of some predefined statistics that were considered relevant for Professors.

Regarding the role of this application, it was identified that it would be necessary a special care for its integration with the existing processes at SugarCRM database level. This means that it was necessary to focus on the transaction management, as well as the implementation of a series of validations to guarantee that the input data is consistent with the structure and constraints implemented at the SugarCRM database level.

The proposed architecture is presented in image 1.

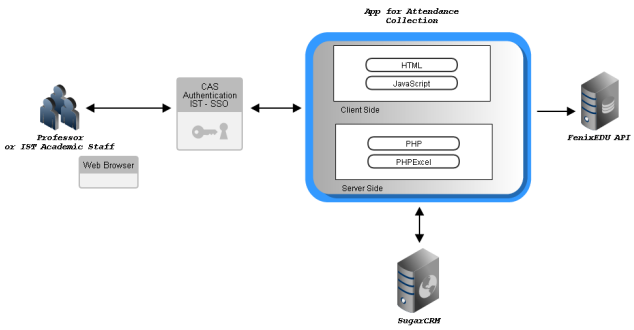

Figure 1: Application for Attendance Collection proposed Architecture

The access to the application by the users is done using a web browser. When a user logs in the IST SSO, it is possible to access the Application for Attendance Collection. The use of IST SSO aims to make more customary and simple the user access to applications. In most of the CRM implementations a major issue identified, related to the difficulty in motivating users to take advantage of the application. The integration with IST SSO aims to build a bridge with the users.

On the other hand, the use of a secure and centralized mean of connection is extremely important, because it makes possible to know what is the role of the user in the IST structure

and it becomes easier the management of their permissions as well as the access to the applications that their profile allows.

Regarding the choices made in the Client Side technologies, the markup language HTML [1] in conjunction with CSS [2] was the obvious choice, since it is the standard in the presentation of Web content. The programming language JavaScript [3] and its library JQuery [4] where also chosen, and in this context its main job is to allow some pre-processing of the data and actions provided by the users of the application during its usage. The choice of Javascript for this purpose is a consequence of the great implementation that this technology has in the Web environment, where it is used in active matter in about 89 % [5] of the websites in the World Wide Web, which makes it a very robust and consensual technology.

Regarding the Server Side technologies, the chosen programming language was PHP [6]. PHP is a reference in the Server Side programming languages, and was further selected the usage of the PHPExcel library, which provides a number of classes that can be used to easily perform the creation and manipulation of Excel spreadsheets. These spreadsheets enabled the entry of data to be done in a simpler manner by the administrative services of DEI, while allowing that some statistics could be exported by Professors. [3, 8]

The other applications with which the Attendance Collection Application is connected are represented in the architecture diagram as black boxes as for understanding of Attendance Collection Application architecture they are irrelevant.

The communication with SugarCRM is bidirectional, as for the presentation of data on the presence already collected and statistics extraction a direct link is required with the SugarCRM database. Conversely, the introduction or modification of attendances already collected in the Attendance Collection Application is made, and these data is stored in SugarCRM.

For the FenixEDU API [7] the communication is unilateral, and of acquisition of information. Through this API and using REST requests it is possible to obtain information about the available courses in the DEI, about the Professors, the list of registered students, the Professors of the courses, and even the time tables of the courses. This information is accessed by the Attendance Collection Application initially for entering this information into SugarCRM database, and for the validations with a predetermined frequency of the data collected.

2.2 SugarCRM

[1]http://www.w3.org/html/ (retrieved at April of 2015)
[2]http://www.w3schools.com/css/ (retrieved at April of 2015)
[3]http://www.w3schools.com/js/ (retrieved at April of 2015)
[4]https://jquery.com/ (retrieved at April of 2015)
[5]http://w3techs.com/technologies/details/cp-javascript/all/all (retrieved at April of 2015)
[6]http://php.net/ (retrieved at April of 2015)
[7]http://fenixedu.org/dev/ (retrieved at April of 2015)

SugarCRM is the application that allows the implementation of the majority of the objectives presented in this article.

By building a personalized access for each of the DEI Professors, it is possible for Professors to access a list of attendance of students in their classes, and they can perceive the student as a whole while also having access to the statistics of their attendance at other courses.

Besides being able to consult an individualized profile of the students, SugarCRM allows the construction of communication campaigns with these students. To this end, the application allows the segmentation of students by variable criteria, and it is possible to see the results of reports generated through the application and to create direct contact campaigns with the students concerned.

The proposed architecture for this component is as follows:

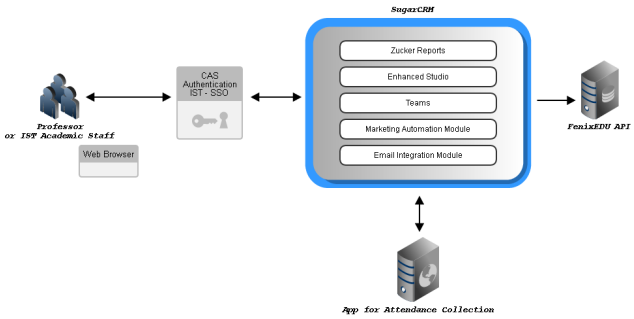

Figure 2: SugarCRM proposed Architecture

As seen in the Attendance Collection Application, the access to SugarCRM is done using the IST SSO, with all the advantages that this represents. Access to the IST SSO is done through the web browser on the user's machine.

Regarding the modules identified as relevant to a solution with the desired characteristics, the Marketing Automation Module is used, as well as the Email Integration Module, the Teams Module, the Enhanced Studio and also the Zucker Reports.

The Marketing Automation Module allows the creation of marketing campaigns focused on a set of specific users. Taking in account the environment in which this implementation of SugarCRM happens, it is intended that through this module it is possible to make the management of the contacts with students, after their segmentation by user-defined criteria occurs.
As the name implies, Email Integration Module allows the configuration of SugarCRM for the use of tools and external mail servers, which in this case is extremely relevant, since the synchronization of e-mail accounts of the Professors with the application greatly facilitates the use of the application.
The Teams Module allows the creation of the concept of *Teams*, that allows a logical division in the application. This logical division aims to ensure that each Professor only has access to information about students who are under its jurisdiction, not allowing the dissemination of information so

that its confidentiality may be in risk.
Enhanced Studio allows the creation of field in SugarCRM that have more complex structures. As some application fields are the result of calculations on relatively large amounts of information in the application, it is necessary to use these module to ensure that this calculations are done correctly and that the values are constantly updated.
Finally the Zucker Reports module is used, which enables the creation and generation of reports on the data in the database, thus being conducted the segmentation of the students.

3. IMPLEMENTATION

3.1 Attendance Collection Application

As a starting point, it is important to emphasize that the main objective of this application is to allow the SugarCRM database feeding with data from the attendance of students in classes, by the elements of the administrative services of DEI.

Later it was considered that opening the possibility for Professors to be able to enter the presence of their students in their classes instead of these being centralized in the administrative services could bring an interesting dynamic to the CRM initiative. Given this decision, it was considered that it would be interesting through this interface that Professors could easily have access to some simple statistics regarding the attendance of students, in order to stimulate the use of the application.

Given these two objectives, it was considered that it would be more relevant than the application presented a robust and simple operation mode as opposed to an overly complex interface.

3.1.1 Application construction steps

The construction of this application consisted of a well-defined sequence of developments and steps, which allowed the creation of the application.
For each of the features implemented was used an agile development methodology, in that whenever an iteration on the development of a functionality happened it was immediately presented to end users for their feedback. Considering this feedback changes were made to the functioning and graphical interface of the application, to make it possible to build a application that stimulates a constant use by the users.

The first step in building the application was to populate the SugarCRM database with the information necessary to make possible the collection of attendances. To make possible a correct collection of the presences it was necessary to populate the database with the following elements:

- Data regarding the Courses - Data like the name of the course, its website, the campus where the course is given, which Professors minister the course, the number of students, etc., was collected
- Data regarding the Students - Data like the name of the students, its IST number, its course, the personal contacts, and the courses where they are registered was collected
- Data regarding the Schedules - Data like the day when the class took place, the start and finish time, and the type of class

Excluding the data regarding the students (provided by the administrative services of DEI), all the information was obtained through the FenixEDU API.
As stated before, the FenixEDU comprehends a range of solutions, from academic and administrative processes to general purpose software libraries. This application is quite complex and aims to serve as a basis for developments in the academic context, but for the purpose outlined above it was only was necessary to use the API provided by FenixEDU.

Through the supply of various endpoints and using a dynamic construction of web hyperlinks it is possible to access all the information that is required to populate the database.

One example of an endpoint used is the link in the footnote[8], through which it is possible to access the information of the course Complex Analysis and Differential Equations. The structure returned by the page is displayed in image 3, where it is possible to check that the pages have a coded hierarchy structure using the format defined by the notation JSON[9].


```
{
  "acronym" : "ACED585",
  "name" : "Análise Complexa e Equações Diferenciais",
  "academicTerm" : "2 Semestre 2013/2014",
  "evaluationMethod" : "Aulas teóricas de exposição da matéria complementadas com sessões\r\nde resolução de problemas nas aulas práticas, individuais ou
em grupo.\r\nA avaliação combina uma componente de avaliação contínua nas aulas\r\npráticas (opcional) e avaliação escrita dividida por 2 testes.",
  "numberOfAttendingStudents" : 150,
  "summaryLink" : "https://fenix.tecnico.ulisboa.pt/disciplinas/ACED585/2013-2014/2-semestre/rss/summary",
  "announcementLink" : "https://fenix.tecnico.ulisboa.pt/disciplinas/ACED585/2013-2014/2-semestre/rss/announcement",
  "url" : "https://fenix.tecnico.ulisboa.pt/disciplinas/ACED585/2013-2014/2-semestre",
  "competences" : [ {
    "id" : "222908803104б",
    "program" : "Análise Complexa. Plano complexo. Séries numéricas. Séries absolutamente convergentes. Séries de potências. Diferenciabilidade, funções
holomorfas. Complementos sobre séries de funções, funções analíticas. Teorema de Cauchy. Homotopia. Fórmula integral de Cauchy. Singularidades isoladas.
Série de Laurent. Teorema dos resíduos. Integrais impróprios. Teoremas de convergência. Regra de Leibniz.\r\nEquações diferenciais ordinárias. Equações de
primeira ordem. Equações lineares, separáveis, exactas e factores integrantes. Existência e unicidade de solução. Extensão de solução. Resolução de
sistemas de equações ordinárias lineares. Exponencial de uma matriz. Fórmula de variação das constantes. Equações lineares de ordem superior.\r\nSéries de
Fourier. Convergência de séries de Fourier. Equações diferenciais parciais. Método de separação de variáveis. Equação do calor. Equação de Laplace.
Equação das ondas.\r\nTransformada de Laplace. ",
    "bibliographicReferences" : [ {
      "author" : "L. Ahlfors",
      "reference" : "3rd edition, McGraw Hill",
      "title" : "Complex Analysis",
      "year" : "1979",
      "type" : "MAIN",
      "uri" : null
    }, {
```


Figure 3: FenixEDU API endpoint structure example

By creating various PHP scripts it has been possible to make successive iterations in the JSON arrays received from the FenixEDU API, and subsequently transform and store this information into the SugarCRM database.

In total it was transformed and introduced into the database the data for the 5 courses managed by DEI. Thus, data were introduced about 80 courses, 2266 students and 6358 class schedules where students presences could be introduced.

3.1.2 Presences introduction/consultation

This feature aims to serve both user groups identified for this application, ie, the administrative services of DEI and the Professors. With this feature the aim is to give these two groups of users the possibility to make the introduction/update of the attendances in classes to predefined schedules, and the possibility to consult the schedules in which presences were already introduced.

The data introduction can be made by two methods.

[8] https://fenix.tecnico.ulisboa.pt/api/fenix/v1/courses/161061294613
[9] http://www.json.org/ (retrieved at January 2014)

The first step of the procedure works the same way for both methods. This first phase involves choosing to which course you want to make the attendance introduction. At the request of users, the courses are grouped by name, which is quite useful for courses that are held for various courses and/or different campus.
After choosing the course it is necessary to choose the class schedule for which it is intended to make the attendance introduction. The identification of the class can be done through a grid where ir is possible to identify the day of the week when the class was held, then the date and start/finish time. At the same time, it is possible to identify the classes by its identifying code, and these codes are the same used on the web pages of the courses in the IST website.
After choosing the class to which it is intended to make the introduction, it is then possible to carry out the introduction by means of the two different methods refered above.

The first involves selecting all students who attended the class, making for such a check on the available checkboxes with the names of the students. Although this method is useful for classes where the number of students present is relatively low, it is very costly when the number of students to introduce is significantly high.

Thus, was created the possibility for users to introduce the data using Excel sheets. As a tool used quite frequently by both groups of users and for allowing the introduction of mass data, Excel presented itself as the ideal tool for bulk introduction of attendances.
For this mean of introducing data it was necessary to create a template file with a well-defined structure, which allowed the validation of the entered data. The selected model is displayed image 4.

SDTF		
Docente Responsável: Prof. Luis Rodrigues		
		Prof. Paolo Romano
IST Number	Name	3.ª feira - 09/04/2014 T- 14h30-16h (F2)
XXXXX	XXXXX	
XXXXX	XXXXX	
XXXXX	XXXXX	1
XXXXX	XXXXX	1

Figure 4: Attendance Collection Excel structure

In this structure the user enters the information for the student (IST identification number and student name), followed by whether they attended the class. The number of students that can be entered is practically unlimited and all of the appearances in of a class may be introduced in one file.

Regarding the technical component, for the graphical interface HTML and CSS were the chosen programming languages. To validate the choices and actions performed by the user JavaScript programming language was used, with its library JQuery. Regarding the processing and validation of input data (either with the manual insertion of each presence, or the Excel validation) was performed using the PHP programming language, together with the PHPExcel library. Upon validation of the actions using JavaScript a significant amount of processing burden is passed on to the client side, allowing the server to have a better overall performance with a higher number of requests.

A careful management of transactions in SugarCRM database is made at this level. Given that it is likely the simultaneous use of the Attendance Collection Application and SugarCRM, and that the applications share the same database, it was necessary that the Attendance Collection Application respected the transaction rules imposed by SugarCRM. Thus, a management was adopted at the transaction level for the Collection Attendance Application which ensures that all data manipulation language activities take place unambiguously and with a policy of all-or-nothing. This was achieved using a series of commands available at the PHP level which allow the management of the transactions in a MySQL database.

3.1.3 Statistical information consultation

This feature is intended to serve only the Professors, since it can be considered that these are conclusions drawn from confidential data. With this feature it is intended to give Professors the ability to access two types of simple statistics regarding the frequency in their classes.

A first statistic lets them know what is the total number of students who were present in each of their classes during the semester, while a second statistic allows them to know the statistics of attendance of each one of his students throughout the semester. These statistics are exported to Excel in order to make it easier the handling by the Professors. The first statistic available is represented in image 5.

	Week 2014-04-14 to 2014-04-19	Week 2014-04-28 to 2014-05-03	Week 2014-05-05 to 2014-05-10	Week 2014-05-12 to 2014-05-17	Week 2014-05-19 to 2014-05-24	Week 2014-05-26 to 2014-05-31		
Classes	9	10	11	12	13	14		Number of enroled students
Attendances	17	30	18	14	13			48

Figure 5: Total Attendances Statistics Excel

In this statistic it is possible to check for each of the classes scheduled for that course what was the number of students that were present. These classes are grouped by week in order to facilitate their viewing.
It is also presented a reference to the number of students who are enrolled in the course.

The second statistic aims to show percentage of presences of each student enrolled in the course, and an example of this statistic is shown in image 6.

Week	1	2	3	4	5	6	7	8	11	12	13	14	15			
Classes	1	2	3	4	5	6	7	8	9	10	11	12	13		Max Classes:	11
XXXXX		B	AB	AB	B	B									7	64%
XXXXX		B	AB	AB	B	B	B	B	B						10	91%
XXXXX															0	

Figure 6: Individual Attendances Statistics Excel

In this type of statistic is presented a list of the students enrolled in the course, and it is possible to identify which classes they attended. Each class is identified by a letter, which identifies in each week what were the classes in which each student was present. Taking into account the maximum number of classes in which the student could be present, it is also indicated the percentage of presences for each student.

The big challenge in the construction of these statistics is

that the courses follow their how organizational model taking in account its specificities, and the type of class that each course can range from four different types: Theoretical, Pratical, Laboratories and Seminars. Thus, it was necessary to create a generic algorithm that allowed to survey the classes that had already been introduced to the selected course, ensuring that each type of classes were grouped in a different tab of the Excel provided to the Professor.
To make it possible to provide this functionality the same programming languages were used as the ones in the previous functionality, with a special emphasis on PHPExcel library.

3.2 SugarCRM

SugarCRM is the centerpiece of the architecture of the proposed solution. Through this application it is intended to make possible the segmentation and profiling of students by the Professors, thus providing an application that allows a closer and personalized monitoring for each student.

SugarCRM is an application that features a very well-defined structure and through its own framework allows it to be possible to do development on its base features. It also allows the incorporation of modules developed by other users, thus enriching the available features.

The application logic architecture is shown in the image 7.

SugarCRM follows a model pattern MVC [6]. Given the changes that had to be implement to accommodate the solution requirements, it was necessary to make changes at all levels.

At the database level the technology chosen was MySQL. This choice is a consequence of the fact that this technology is free, coupled with the fact that MySQL is recognized as being a very consistent technology. Regarding the Data Layer the PHP DBManager class was used, which allowed to perform a series of operations from the creation of users to the creation and manipulation of tables in the database. Regarding the Business Layer Sugar Beans (a Object Relational Mapper)was frequently used, allowing the creation and management of objects created from the database and used by several modules created in SugarCRM.
Regarding the Application Layer, SugarCRM as shows little flexibility and openness to changes in its interface. Thus, this layer turned out to be not handled directly at the code level, but mainly through the graphical manipulation tools offered by SugarCRM [1, 10]. Note that this implementation was accomplished using SugarCRM release 6.5.15, with Community Edition, which is the free version of SugarCRM.

3.2.1 Database structure

SugarCRM database structure is very complex, and as such it will not be analysed in the present document. However, we will describe the structures that was created to store the data on elements that represent the IST business.

The created elements and the relationship between them is shown in image 8.

By implementing these tables it was possible to map much

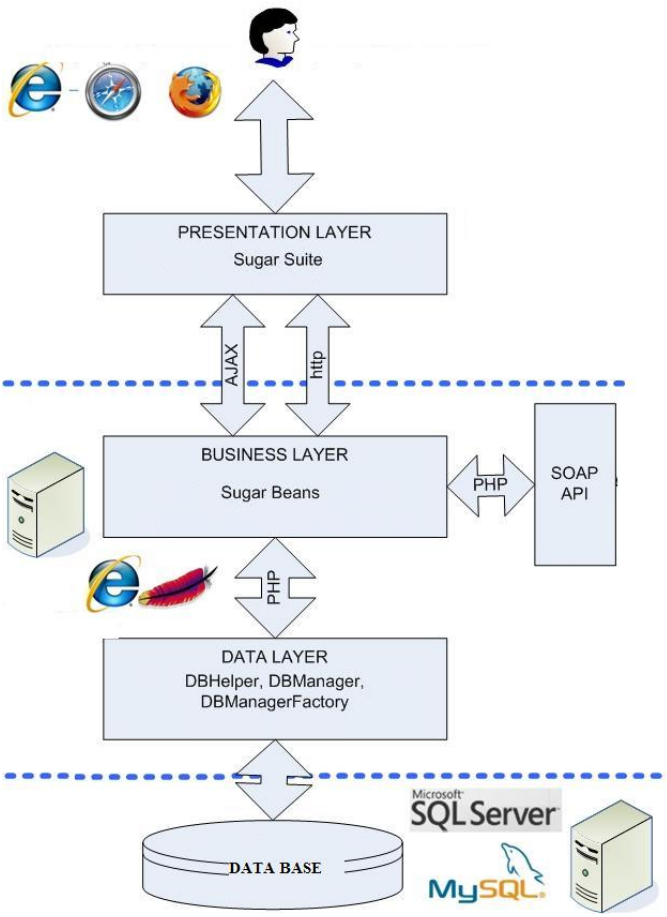

Figure 7: SugarCRM Architecture

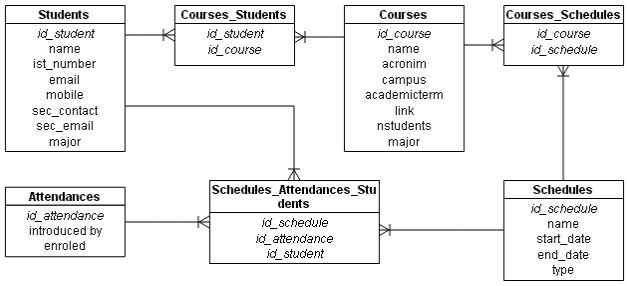

Figure 8: Implemented tables architecture

of the business that this document addresses.

The tables implemented have the following utility:

- Students - Here is stored all the relevant information about the students, including their contact details.
- Courses - In this table is stored data on courses that are teached in DEI. The academicterm variable allows to identify the semester in which the course was given, since it is impossible to uniquely identify the chairs just

by its name. The nstudents field indicates the number of officially registered students in each courses.

- Schedules - This table contains information regarding schedules where it is possible to record attendances. In the field type it is possible to identify the type of class (if it's Theoretical, Practice, Laboratory or Seminary).

- Attendances - In this table the presence of students in the classes are recorded. The introduced by field allows to identify who was responsible for the introduction of that presence, and through field enrolled it is possible to identify if the student is enrolled in the course.

Note that the information on the Professors is not described in this structure. This is because for this was used one of the base tables of SugarCRM, the Users table. All tables represented in this structure are also linked to a number of other native tables of the SugarCRM structure, but since that relations have little importance in the implementation of the IST business rules, they are not represented in this architecture.

3.2.2 Teams module

One of the obvious concerns when implementing this application relates to the safety and limitation of access to data.

Regarding the connection restriction it is possible to achieve this objective using the integration with the IST SSO, but the limitation of access to the data within the application has to be reached through other means.

Because we are dealing with sensitive data from students, it is necessary to ensure that even the users who can access the application only access data which concerns them directly.
An example of data visualization limitations is easily identified: A Professor should only have access to information relating to students who are enrolled or attend their courses. This means that they should not have access to data from that students with whom there is the need to obtain information in order to create a one-to-one academic monitoring.

Because we used the Community Edition version of SugarCRM, it is assumed by the application that any user who has access to the application can access all the data stored in it.
Thus, it was necessary to use, configure and change a module provided by SugarCRM community, the Teams module. Teams module applies the concept of teams, which is very common in the business environment. This concept means that users are grouped into teams, and for each team a perimeter of access to data is set.
Despite using this module provided by the SugarCRM community, the specificity of this project required the manipulation of the module source code to accomplish its mission. Since it is intended that each Professor only accesses information of their courses, classes and students, it can be considered that each Professor is a Team. So, after understanding the inner workings of this module, multiple SQL scripts were developed to carry out the creation and massive population of SugarCRM tables for creating individual Professor Teams.

3.2.3 Dashlet building

As an initial image of the application, the main screen dashlets are a way to present the most relevant information to the user, and thus captivate him to use the application frequently.
These dashlets are presented on the homepage of SugarCRM, and teh information presented and its order is configurable.

For the homepage of each Professor three dashlets were created. One of the dashlets presents information on students attending courses that the Professor minister, another that displays the reports created and consulted lately, and finally a dashlet that displays information about the courses that are taught by the Professor. The latter dashlet can be seen in image 9.

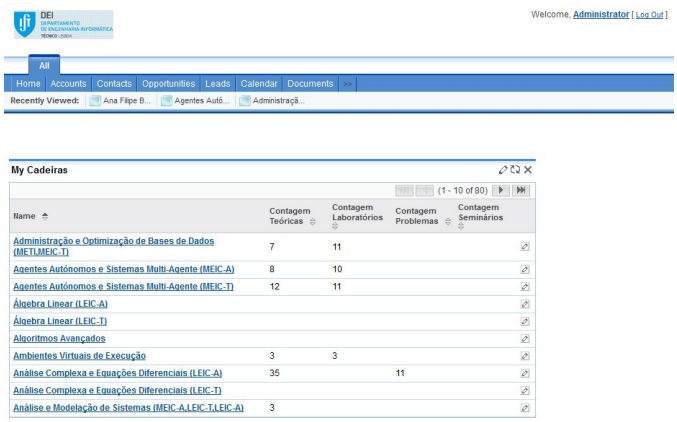

Figure 9: Global attendances introduction dashlet

This dashlets presents another challenge faced during the implementation. The fields with the count of the number of classes introduced to each of the courses according to the type of classes is considered a complex type field in SugarCRM. In practice, a complex field is a field whose value must be calculated and updated with some frequency, depending on the changes in the database.

For it to be possible to create such fields it was necessary to resorted to a SugarCRM module called Enhanced Studio. Through this module it is possible to create fields whose value can be updated depending on the implementation of SQL code to perform queries on the database and return values that are updated and displayed to the user.
To apply the changes to the source code of this module it was used PHP programming language.

3.2.4 Segmentation and communication

This feature represents a significant portion of the objectives that were proposed for this implementation.
It is intended that with this feature it is possible for a Professor to create reports based on the collected data, allowing him to identify groups of students for which he considers that closer monitoring is beneficial. It also offers the opportunity to create a communication channel with the identified students using the email functionality.

In the CRM methodology this feature is the embodiment of

a segmentation activity, in which the segmentation of customers based on a characteristic identified as relevant occurs. Taking in account that specific and relevant information waspreviously was collected on these same customers (in this case, the attendances in classes), we are facing an example of profiling.

By using the Zucker Reports module it is possible to create on-demand reports, by filling some input parameters. The reports are based on SQL queries that are applied to the database, then returning a result set. The returned result set is also configurable.
After the report runs it is then possible to identify how the data may be view, and whether to take any action in based on the new information gathered.
One possible action is to create a Targets List. As previously explained, in the CRM methodology a Target is an existing customer that after performing a segmentation action is considered as a possible target for a particular campaign/contact.
By adding the students returned by the report to a Targets List it becomes possible to use the Marketing Module for the creation of a targeted marketing campaign, in particular for this group of students.

The Marketing module allows tracking of a marketing campaign from its inception to its completion. In the case of this implementation it is intended to assume a marketing campaign of somewhat different contours to a common marketing campaign in a business environment. In this implementation it is intended that a marketing campaign represents a set of logical actions taken by a Professor to a Targets List identified as a consequence of the segmentation made using the reports.
To achieve this goal it was necessary to make several changes to the Marketing module base structure [7]. One of the most relevant changes was the withdraw the obligatoriness that this module had regarding the number of sequential steps to building the campaign and setting a deadline for completing these steps.
This change required a rather large redefining of the framework of this module, and these changes have been made using the SugarBeans SugarCRM model.
After making these changes it became possible to perform an unlimited number of actions in each of the campaigns. These actions are as varied as setting alerts for changes in the Targets List, the management of the elements that make up the Targets List, or the management and use of a communication channel.

In this implementation the communication channel was created and configured through the email. To do this it was necessary to configure the SugarCRM Emails module.
The SugarCRM Emails module offers the possibility to use and configure e-mail synchronization with the application. By using this component it is possible for the user to view its emails in SugarCRM, make the association between emails and students in the application, or be able to make email communications directly from the application and using its email address.
In the context of this implementation, the setup of the user email account in the application is of great importance, as it allows the use of the application has a continuity from the Professor, as often allows to be reminded of the necessity of perform a relationship management of their students.
For this implementation it was considered that as all Professors have a IST e-mail address, it would be interesting to test and configure the use of the IST mail server in the application.

To do this, the settings presented in the image 10 were tested, where for the test user it was possible to make the loading of the emails present in its account, and to make the sending of emails through that same account.

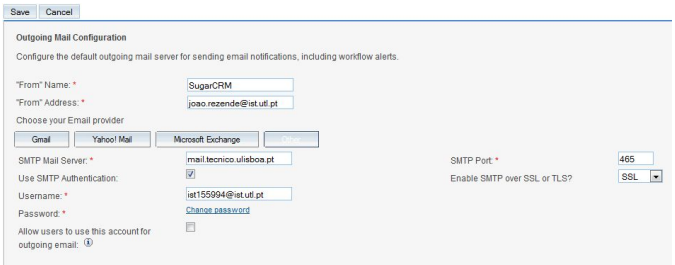

Figure 10: Emails module configuration

4. EVALUATION

The evaluation is made by checking if through the implementation of the proposed solution it was possible to achieve the objectives described in the previous sections.
To this end, it is recalled the objectives that had been proposed:

1. Implement a CRM application for aggregating relevant information on students
2. Make the introduction of the attendance at classes in order to enrich the information on students and allow a personalized support
3. Getting to profile and segment students, by applying filters and analyses to the data collected over time
4. Allow the performing of mass communications based on the profiling and segmentation of the students

So, it will be present how these objectives have been achieved, using a proof of concept performed using real data collected by the administrative services of DEI and that allowed to prove that the solution implemented meets the necessary requirements.

Section *User Acceptance Tests* intends to present tests that confirm that using the functionalities of the application it is possible to perform the activities described in the objectives.

In the section *Simulation Tests* intends to prove that by using the implemented solution it is possible to perform the tasks described more efficiently.

4.1 User Acceptance Tests

This section is intended to explain what tests where performed to prove that through the implementation carried out it is possible to achieve the objectives set during the analysis of the problem.

4.1.1 Test Specification

In order to prove that with this CRM implementation it is possible to achieve the proposed objectives it was necessary to create a test that represented a proof of concept.

The first step was to use the Attendance Collection Application to conduct the introduction of attendances of some students for a number of classes of one of the available courses. Then it was necessary to use the implementation of SugarCRM in order to identify which students attended less than a given number of classes. For this it was necessary to create a report that would allow to proceed with this research.
Using Zucker Reports three input parameters were created for the report: The minimum number of classes attended, what type of class we wanted to analyse, and to which course we wanted to run the analysis.
After the creation of these input parameters it was necessary to create the report. For this it was necessary to define the SQL code that allowed to search the SugarCRM database. For this report in particular it was used the code shown in the image 11.

```
SELECT alu_alunos.id
FROM alu_alunos INNER JOIN
(select pres_presenca.id_aluno, COUNT(pres_presenca.ID) AS CONTAGEM_PRESENCAS
from pres_presenca inner join hora_horarios ON pres_presenca.id_horario = hora_horarios.id
WHERE hora_horarios.id_cadeira like '$nome_cadeira' and hora_horarios.tipo like '$tipo_aula'
GROUP BY pres_presenca.id_aluno
HAVING COUNT(pres_presenca.ID) < $aulas_assistidas) AS B ON alu_alunos.numero = B.id_aluno
```

Figure 11: SQL code to run report

Later it was possible to run the report by filling the necessary input parameters. The report return was a list of students, which was stored as a Target List.
From there it was necessary to create a new marketing campaign, adding the Target List created as the final recipients of the campaign. From that moment it was then possible to get in touch with the students concerned, using for such the previously configured email, and getting this communication associated with the profile of students.

4.1.2 Conclusions

Through the previously presented case study it was concluded that all objectives can be achieved through the implemented solution.

Using the Attendance Collection Application the objective of *Make the introduction of the attendance at classes in order to enrich the information on students and allow the personalized support* is achieved, and this application also contributes decisively to the *profiling of the students*.
Through the implementation of SugarCRM, and as was seen with the use of several built and manipulated modules it is possible to meet the objectives of *Implement a CRM application for aggregating relevant information on students*, the *Getting to profile and segment students , by applying filters and analyses to the data collected over time and still* objective, and also the *Allow the performing of mass communications based on the profiling and segmentation of the students* objective.

Through this test it is possible to prove that from a functional point of view the implemented solution provides all the functionalities needed to achieve the described objectives.

4.2 Simulation Tests

This section is intended to compare how efficient are in performing the activities described in the objectives when using the CRM implementation and the traditional methods currently available. To this end, it was asked users to performed the sequence of actions that were described in the previous section, and cover all the objectives for this implementation.

So that the results were more conclusive, the simulations were divided into two components. In a first test users were given only a few references regarding the application they were about to use, not being given any indication as to the features that each application had. In a second test it was given users a more detailed insight into the existing features of each application.
All tests were repeated twice in order to be able to draw conclusions about the real gains with regard to continued use of the application, as opposed to the conclusions drawn by the tests described above, where it is possible to draw conclusions about the learning curve for the applications.

4.2.1 Traditional methods

This method aims to understand how is it possible for users to be able to meet the actions described in the test mentioned in the previous section with the tools and applications they currently have.
In image 12 are the mean values collected for the set of tests that were conducted.

	No explanation of the features	With the features explanation
First time	12m05s	11m35s
Second and third time	8m43s	8m32s

Figure 12: Values collected with traditional methods

As stated above, the collected values of the time needed to perform the activity indicated for the two scenarios described above are presented.
It should be noted that there are no major differences with regard to the time that users need to accomplish the task when we proceed to the explanation of the features and applications that can be use to achieve the objective.
This is easily explained by the fact that users are already familiar with most of the tools used to achieve the proposed objectives. When this is so, an explanation of the existing features and applications does not have a impact on how quickly users can perform the proposed tasks.

4.2.2 Using CRM Implementation

Using this method it is expected that users perform all actions previously described using only the applications implemented in the proposed solution. Note that prior actions, such as creating the user in the SSO and applications is deemed to have been made earlier, as well as the user email configuration in SugarCRM.

In image 13 are the mean values collected for the set of tests that were conducted.

As stated above, the time needed to perform the activity indicated for the two scenarios is described above.
In this case it is relevant to note the difference in the values

	No explanation of the features	With the features explanation
First time	12m42s	9m54s
Second and third time	4m04s	4m13s

Figure 13: Values collected with new methods

collected when the user does not have any formal presentation to the features and applications available from when the user is formally introduced to the applications and features. The values obtained will be subject to reflection in the following subsection.

4.2.3 Conclusions

To draw conclusions of the tests the graphics in the images 14 and 15 were used, which were obtained as based on the data presented in the preceding subsections tables.

In the graphic of image 14 it is showed the comparison between the collected time when there is no explanation of the type of functionality that make up the final solution, for both methods.

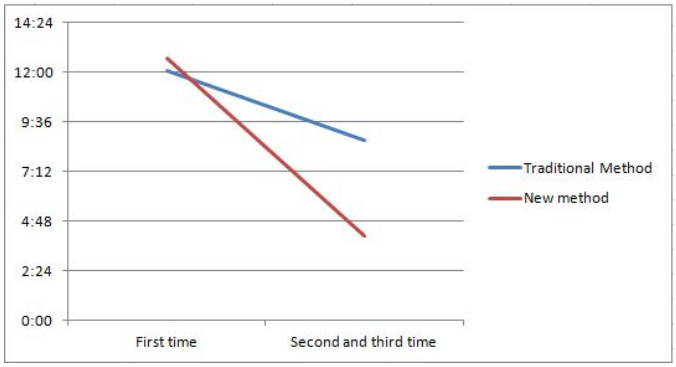

Figure 14: Time consumed to perform actions without explanation of functionalities

In the graphic of image 15 it is showed the values obtained from the collection of time required to complete the action when an introduction of the applications available and their functionality is made.

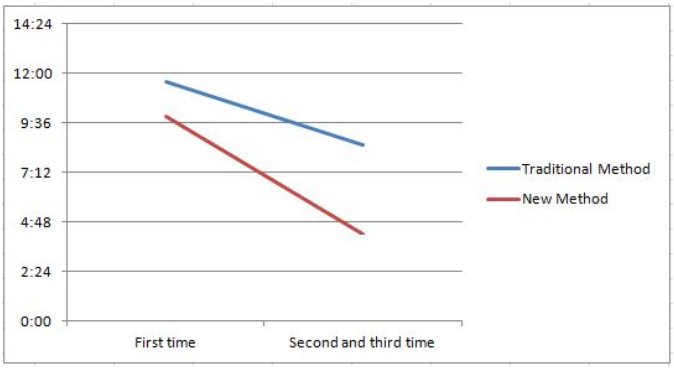

Figure 15: Time consumed to perform actions with explanation of functionalities

In both cases it is contemplated the time necessary to successfully carry out the first trial and the average of the two further attempts.

Using the values of both graphs it is possible to draw some conclusions.
The first is that when users are unaware of Attendance Collection Application and SugarCRM and are faced with their use have some difficulties in completing the proposed tasks. This may mean that the application operation mode is clearly not intuitive, since the common user finds it difficult to complete the task in less time than using the tools that he is used to.
However, when we look for continuous use, the solution presented shows encouraging results. We realize that ceases to be relevant that the user had a formal presentation to the application functionality, as with repeated action it becomes intuitive. We can verify that the frequent use of the implemented solution leads to a reduction of the activity of the proposed actions in over 50% of the time, which represents a very interesting gain.

As a broad analysis, it can be concluded that this implementation allows to perform the activities for which it is proposed, and that the necessary training in their use can dramatically reduce the time spent in profiling and segmentation of the students to provide them with better monitoring.

5. CONCLUSIONS

5.1 Achievements and conclusions

The high-level purpose for this article was to ***understand of the benefits of implementing a CRM initiative to monitor the performance of students during a semester, by the Professors of the courses of DEI.***

Given the objectives of lower level of granularity which were described in the *Evaluation* section have been achieved, it can be concluded that there are clear benefits to implementing a CRM initiative to monitor the performance of students during a semester. By providing to Professors a set of tools that allows them to easily understand which is the current situation of their students opens up the prospect to Professors to act more effectively in preventing cases of abandonment and failure in academic courses that they minister.

However, what may be debatable is the adequacy of a CRM application of type Generalist to an environment with the specifics of the academic environment.
With this article it is confirmed that such is possible and reachable, but however there are some considerations that should be taken into account.
Taking in account the tests made, it is possible to see that when the user is not aware of the functioning of the applications developed some difficulties arise in performing the task proposed since it may require a prior knowledge of the logic associated with a CRM initiative. The particular case of the use of SugarCRM may be a issue, as there is a clear difficulty in manipulating the GUI.

With this we conclude that using a solution of the Generalist type it is possible to implement a CRM initiative in the academic environment, and it would probably be possible to obtain even better results if Go Vertical CRM application was used, since this applications are already built taking

in account the specificities of a particular business environment.

5.2 Future Work

As possible iterations in this solution there are two major areas that can be developed.

The first area regards the increase of the amount of information that allows the profiling of students. By including more criteria, such as information relating to intermediate evaluations, it becomes possible the construction of more complete student profiles, resulting in a more accurate segmentation. A second area involves the expansion of this concept to other components of academic life. An example of the application of this methodology to other areas may be the monitoring of students during the period of the Theses. Because it is a very particular phase of the academic path, it would be interesting to be the target of a CRM initiative focused on that stage.

6. REFERENCES

[1] M. A. Bain. *Sugarcrm Developer's Manual : Customize and Extend SugarCRM*. Packt Publishing Ltd, 2007.

[2] B. H. W. Dale L. Goodhue and H. J. Watson. Realizing business benefits through crm: Hitting the right target in the right way. *MIS Quarterly Executive*, 1(2):79–94, 2002.

[3] P. DuBois. *MySQL (5th Edition)*. Developer's Library, 2013.

[4] B. A. Kubi and A. K. Doku. Towards a successful customer relationship management: A conceptual framework. *African Journal of Marketing Management*, 2(3):037–043, March 2010.

[5] W. Leenslag. Influential aspects of relevance for a crm system in a b2b environment. 3rd Twente Student Conference on IT, 2005.

[6] J. Mertic. *The Definitive Guide to SugarCRM : Better Business Applications*. Apress, 2009.

[7] J. Mertic. *Building on SugarCRM*. O'Reilly Media, Inc., 2011.

[8] R. Nixon. *Learning PHP, MySQL, and JavaScript*. O'Reilly Media, 2009.

[9] A. Parvatiyar and J. N. Sheth. Customer relationship management: Emerging practice, process, and discipline. *Journal of Economic and Social Research*, 3(2):01–34, 2002.

[10] M. J. R. Whitehead. *Implementing SugarCRM*. Packt Publishing Ltd, 2006.