

# Implementation of a Prototype in Free Software for the Management of Loan of Laboratory Materials at an Institution of Higher Education

Ever Blanco<sup>#1</sup>, Diofanor Acevedo<sup>\*2</sup>, Freddy Méndez<sup>§3</sup>

<sup>#</sup>Faculty of Engineering, Systems Engineering Program, Universidad de Cartagena,  
Av. El Consulado, St. 30 No. 48-152. Cartagena de Indias

<sup>\*</sup>Department of Food Engineering, School of Engineering, Research Group Nutrición Salud y Calidad Alimentaria (NUSCA), Universidad de Cartagena, Av. El Consulado,  
St. 30 No. 48-152. Cartagena de Indias

<sup>§</sup> Systems Engineering Program, Research Group PRISMA,  
Universidad de Autónoma Bucaramanga, Av. 42 #48 - 11, Bucaramanga, Santander  
blancoever@hotmail.com<sup>1</sup>, diofanor3000@gmail.com<sup>2</sup>, fmendez@unab.edu.co<sup>3</sup>

**Abstract**—This paper describes the analysis, design and implementation of a software tool that provides assignments for the management of loan materials in libraries and laboratories of a Higher Education Institution. Within the development time used under the methodology of evolutionary prototype which part of the initial deployment where the user interacts with your comments thus creating a refining process through the various versions until a suitable system are determined. Development outcomes of the project are presented, which include software implementation: based on the ISO/IEC 25010 standard under the criteria of functionality and usability, the last criterion supported under the survey of a sample of 8 users from a population of 450 students. The software has complied with the requirements established in product requirements.

**Keywords:** requirements, requirements engineering, tools, techniques.

## I. INTRODUCTION

The process of assigning classrooms, laboratory materials and multimedia resources requires time and staff at the various universities; most of these institutions employ a simple computer-assisted heuristic approach, consisting of a computer-based assignment based on the order of arrival of the lists of classroom requirements for each session to be taught, which are lacking in the analysis of data to measure compliance with the objectives proposed by the parties involved in the allocation of resources[1,2].

The problem with laboratory scheduling is the arrangement of objects within a pattern in time or space in such a way that certain objectives are achieved and restrictions on how such objects are to be arranged are satisfied[3,4]. In any programming process, there are three fundamental elements: objects, patterns and constraints. Objects refer to the resort to organizing or arranging, which are people, laboratories, classrooms, among others; the pattern is the order that events must follow and is created as part of the scheduling process; and restrictions are defined as tangible or intangible relationships between objects and represent the rules that must be followed for building a pattern. Usually referred to as Timetabling, the allocation of a set of resources to specific objects subject to restrictions, being located in time frames, in such a way that it satisfies as much as possible a set of desirable objectives[2,5,6].

Suárez et al.,[6] proposed the solution to the problem of optimal scheduling of school schedules in a Colombian public school. They considered not only the appropriate allocation of classrooms and teachers, but also the cognitive rhythms that students present as the most important factor in the optimization process. The results show a higher efficiency of the algorithm compared to others applied to the same problem and evaluated in the same way. Scheduled groups showed a reduction in academic mortality levels compared to time schedules that did not take into account students' cognitive rhythms.

The process of allocating resources such as classrooms, laboratory materials and multimedia resources is one of the main activities carried out in the different universities. In many of these institutions, these processes are carried out manually and in some cases are partially systematized, despite the fact that today there are technological tools that could be used to implement an effective solution for such a situation. In the study of the free software systems developed for the management of resource loans, there are not properly developed applications for the management of material loans. Some of the systems are simple applications for schools and not for universities, others are more advanced projects are not very flexible, such as for example, they manage

the assignment in time range, others are oriented to the reservation of rooms for casual activities such as events, in general it is necessary to implement a system that manages the loan of any type of material in institutions of higher education. The objective of this research was to develop a system of materials loan management at prototype level using free software tools to improve internal processes in Higher Education Institutions.

## II. METHODOLOGY

To implement the prototype services for the materials loan management system at the university, the evolutionary prototyping was used as a development methodology. The methodology used is described in the following items.

### A. Determination of global software needs

The overall needs of the software were highlighted by holding meetings between developers, laboratory room managers and the course shows that it was the users of the system who contributed ideas during the development of the system, identified requirements and identified sub-processes where more emphasis was needed.

### B. Design of the prototype

The design of the prototype was carried out taking into account first the relevant aspects of the code, such as query methods, insertion, updating and deletion in the database. The design of the user interfaces such as input/output forms was then continued.

### C. Prototype evaluation

The prototype was evaluated by project developers, laboratory room administrators and the sample course. The methodology for the usability test was done in a similar way to the procedure made by Mercovich [7].

## III. RESULTS

### A. Requirements analysis

One of the elements taken into account for the requirements analysis is the functional requirement. For this, it was determined the possible transformations that would occur in a system due to inputs and how their effect would be on their possible outputs, for the study of the prototype in this dimension, a comparative analysis was carried out with five applications, software related first under free software licensing and second under the concept of loan management or resource allocation, framework in the technique used in requirements engineering where it was considered that for the development of the prototype, the requirements that were known were identified, to then indicate areas in which possible implementations were necessary [8], the criteria used were: The programs focused on universities, assigning specific hours, individual assignment of the resource, given a resource allows to consult its availability, allows to have various types of user and use through the web, the results can be seen in table 1.

Table 1. Requirements analysis.

Requirement	Software				
	A	B	C	D	E
The programs focused on universities	NO	NO	YES	YES	YES
Assigning specific hours	NO	NO	YES	NO	NO
Individual assignment of the resource	NO	NO	NO	NO	NO
Given a resource allows to consult its availability	NO	NO	NO	NO	NO
Allows to have various types of user	NO	NO	NO	NO	NO
Use through web	NO	NO	NO	YES	YES

The first characteristic refers to whether the application is focused on school or university as required, three of them fulfil it. The second characteristic refers to whether in the application the resource allocation is done in pre-established time ranges or it can be done in any given hour as required, only one of them fulfils it. The third characteristic refers to whether in the application the assignment of the resource is made to previously formed groups or it can be made to an individual person as required, none of them fulfil it. The fourth characteristic refers to whether in the given application a resource can be consulted its availability as it is required, none of them fulfills it and the fifth characteristic refers if the application allows several types of users as required, none of them meet it. The sixth characteristic refers to whether the application can be used via web as required, two of them comply.

**B. Functional requirements**

The final requirements are set out below:

- Add, modify and remove laboratory elements in the system.
- Add, modify and remove students from the careers that use these laboratories.
- Add, modify and remove the auxiliaries that will be using the system in the laboratories.
- Keep a record of the maintenance performed on each element of the laboratory.
- Control student lending of laboratory items with a restricted penalty for later loans for failure to deliver on time.
- Allow students to reserve laboratory items.
- Show statistics of usage of the elements.
- Monitor student damage to items.
- Assign 3 user types, General Administrator, Local Administrator and Student.
- Assign to the general administrator profile the management of inventories of all engineering laboratories.
- Assign to the Local administrator profile the management of loans and inventories of the laboratory elements in charge.
- The system must differentiate between materials that are lent by days, weeks or months.
- The system must answer queries about the number of laboratory elements, type of elements, etc., which are in the laboratory or on loan.
- The system must provide up-to-date information about materials that are not available for your loan.

**C. Usage cases**

In the table 2 shows the possible uses for each of the three types of users required:

Table 2. Usage cases of the three types of users

USER TYPE	Usage cases
Common user	View application status
	Make requests
Administrator	Make comments on applications
	Perform physical inventory
	Deliver and receive items
	Approve and deny applications
	View reports
Super User	Adding elements to labs
	Maintain elements
	Unsubscribe items
	Assign Permissions
	Creating Users

**D. Commissioning**

The solution was hosted on a high-performance VPS (Virtual Private Server) server, which is active 24 hours a day, all year round, avoiding problems that could cause intermittent service.

**E. System testing**

The tests applied to the system are presented below to ensure that it was developed correctly, without design or programming errors.

*1) User: Super User*

Case of use: Add elements to the laboratory.

Click on the new button in the Elements window

In the Add Element window, type the Test Description, from the drop-down lists select the Laboratory and select the Element Type, click on the Save button

2) *User: administrator*

Case of use: Approve applications.

In the main window, click on the edit button corresponding to request number made by the student user. In the Change Status window, select the Approved option from the Define Request Status list. In the Change Status window, click the Save Request Status button.

In the main window, you can see how the 26th request made by the student user passed from the Requested state to the Approved state.

3) *User: Common user*

Case of use: Make requests.

In the main window click on the menu option New request. In the Create New Request window, click on the from text box and select the start date. In the Create New Request window, click on the to text box and select the end date. In the Create New Request window, click on the Search button in order to view the items available for removal. In the Create New Request window, separate the Video beam element by typing one in the Requested box. In the Create New Request window, click on the Save Request button. In the administrator user view, it can be seen that the request of a videobeam made in the previous point was successfully registered.

F. *Evaluation of prototype quality*

The quality of the prototype was evaluated according to ISO/IEC 25010-Quality Model (2005) in its functional characteristics, and usability in each of its four sub-characteristics: Appropriateness, ease of learning, operability and attractiveness. The functionality feature was evaluated each time a part of the system was developed with its respective coding, it was tested by means of unit tests to ensure that the requirements were met.

The methodology used for the usability test is presented in the following sections:

- 1) Planning
- 2) Test
- 3) Analysis of the data

1) *Planning*

At this stage, by means of non-probabilistic sampling, three representative groups were selected for the three types of users: one as a super user of the system made up of two people, another as a system administrator made up of two people and a third group as a common user made up of eight people.

The tasks to be performed by each group of participants were designed according to their user type, such as making a request for the common user, approving and denying requests for the administrator user and assigning permissions for the super user.

It was defined that the site of application of the test would be in the computer laboratories of the University of Cartagena previously making the reservation of the room.

2) *Test*

The following types of questions were applied in the questionnaire:

Affirmation with degree of acceptance from 1 to 5, meaning that 1 is in complete disagreement with the affirmation of the question and 5 is in complete agreement with the affirmation of the question.

From YES or NO; meaning with YES you are in complete agreement with the affirmation of the question and NO you are in complete disagreement with the affirmation of the question.

3) *Analysis of the data*

Functionality tests were carried out for each of the cases of use, and were satisfactorily carried out for all cases of use. The usability test applied to the three types of users for the sub-characteristic of Appropriateness yielded the results shown in Fig. 1.

Based on the results shown above, it is observed that the super users placed their degree of acceptance on level 4, the administrators placed their degree of acceptance on levels 4 and 5, from the common users 5 placed their degree of acceptance on level 5, and 3 placed their degree of acceptance on level 4; it can be observed a high degree of appropriateness in the system on the part of the users surveyed, since their responses were as follows.

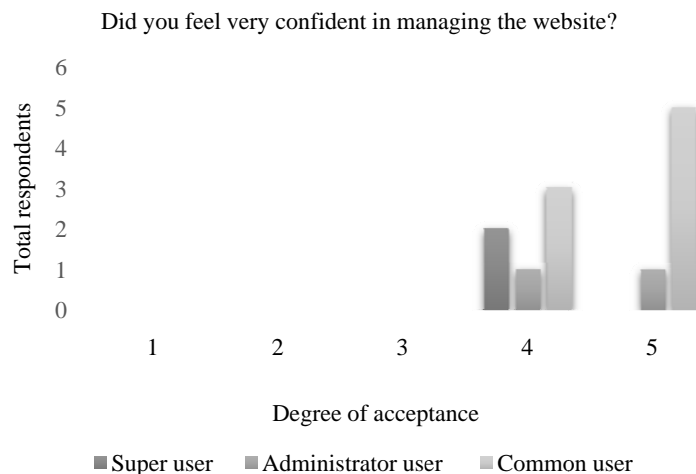


Fig. 1. Results for the sub-characteristic Appropriateness

The usability test applied (Fig. 2) to the three types of users for the Ease of Learning sub-characteristic yielded the following results:

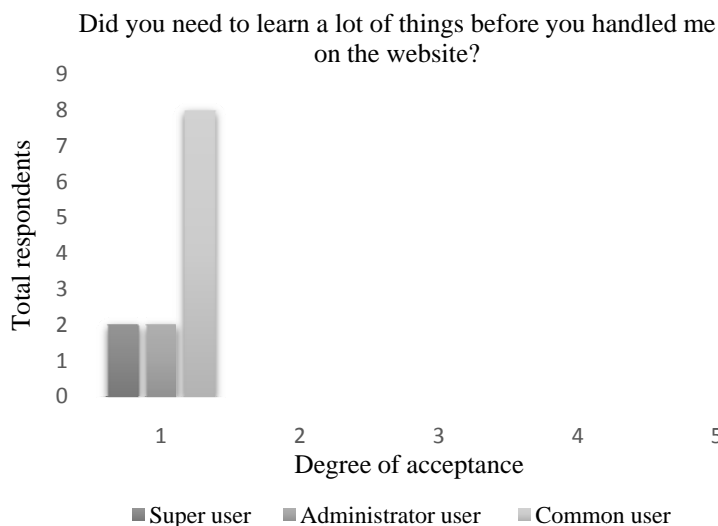


Fig. 2. Results for the Ease of Learning sub-characteristic

This figure shows how the super users consider that they do not need to learn many things to manage the system, as well as the administrators and common users, placing their responses at level 1 which is the lowest degree of acceptance of the affirmation that involves having to learn many things to manage the website.

The usability test applied to the three types of users for the Operability subcharacteristic yielded the following results (Fig. 3).

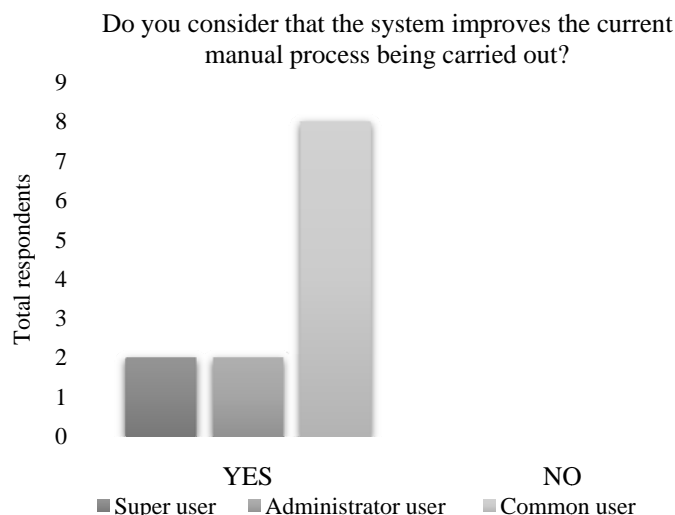


Fig. 3. Results for the Operability sub-characteristic

The responses of the three types of users were consistent with the assertion that the system implemented improves the manual process that was being carried out so far.

The usability test applied to the three types of users for the Attractibility sub-characteristic (Fig. 4) yielded the following results:

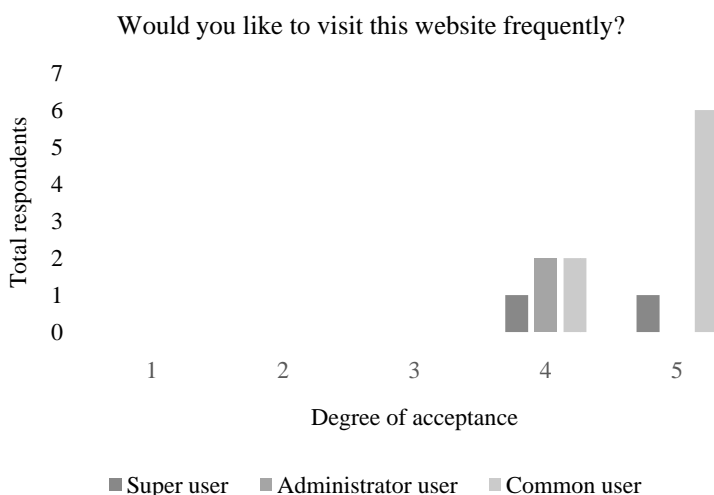


Fig. 4. Results for the Attractibility sub-characteristic

The test showed a high index in the user's taste for interaction, where it can be seen that the super users placed their degree of acceptance on level 4 and 5, the administrators placed their degree of acceptance on level 5, 2 of the common users placed their degree of acceptance on level 4, and 6 placed their degree of acceptance on level 5; it can be observed a high degree of attractiveness in the system per part.

#### IV. CONCLUSION

The use of this type of tools helps the university to have the services of laboratories and libraries systematized and furthermore not many hours of training are needed for those who were using this software, which would be employees of the institution of higher education and students of the same.

## V. REFERENCES

- [1] P. Avella, B.D'Auria, S. Salerno and I.Vasil'ev, "A computational study of local search algorithms for Italian high school timetabling," *Journal of Heuristics*, vol.13, no. 6, pp.543-556, 2007.
- [2] M. Carter and G. Laporte, "Examination timetabling: Algorithmic strategies and applications," *Journal of the Operational Research Society*, vol.47, no. 3, pp. 373-383, 1996.
- [3] P. Flórez, "Revisión de algoritmos genéticos aplicados al problema de la programación de cursos universitarios," *Programación Matemática y Software*, vol. 3, no. 1, pp. 49-65, 2011.
- [4] E. Burke and S. Petrovic "Recent research directions in automated timetabling," *European Journal of Operational Research*, vol. 140, no. 2, pp. 266-280, 2002.
- [5] V.F. Suárez, A. Guerrero and O.D. Castrillón, "Programación de horarios escolares basados en ritmos cognitivos usando un algoritmo genético de clasificación no-dominada, NSGA-II," *Información Tecnológica*, vol. 24, no. 1, pp. 103-114, 2013.
- [6] I.X. Tassopoulos and G. Beligiannis, "Using particle swarm optimization to solve effectively the school timetabling problem," *Soft Computing*, vol. 16, no. 7, pp. 1229-1252, 2012.
- [7] E. Mercovich, "Workshop: cómo hacer un test de usabilidad de un sitio," Available in: <http://planeta.gaiasur.com.ar/infoteca/test-de-usabilidad-de-un-sitio.html>, 2014.
- [8] M. Arias, "La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software," *InterSedes: Revista de las Sedes Regionales*, vol. 6, no 10, pp.1-13, 2005.