# LEARNING CONTENT MANAGEMENT SOFTWARE PERSONALIZED FOR A UNIVERSITY ENVIRONMENT

## *SOFTWARE DE ADMINISTRACIÓN DE CONTENIDOS DE APRENDIZAJE PERSONALIZADO PARA UN AMBIENTE UNIVERSITARIO*

***Aldonso Becerra Sánchez***
Universidad Autónoma de Zacatecas
*a7donso@uaz.edu.mx*

***Gustavo Zepeda Valles***
Universidad Autónoma de Zacatecas
*gzepeda@uaz.edu.mx*

***Alexandro Iván Pérez García***
Universidad Autónoma de Zacatecas
*alexandro20.0@icloud.com*

***Uriel Ramírez-García Correa***
Universidad Autónoma de Zacatecas
*uramirezgc@uaz.edu.mx*

***Santiago Esparza Guerrero***
Universidad Autónoma de Zacatecas
*chago@uaz.edu.mx*

## Abstract

The use of content management systems (better known as e-learning platforms) among universities is a key element in today's educational approaches. Nevertheless, common content managers found in the web aren't always the best option, given that they do not offer a personalized integration to satisfy the necessities of individual institutions, or they simply lack in fundamental areas, which leaves the necessities of both professors and students unsatisfied. Our proposal focuses on the design and implementation of a content management system integrated into the higher education environment, which takes into account the use of an academic model being utilized within our academic institution (Universidad Autónoma de Zacatecas), making emphasis in the interaction of both the students

and professors with the system. This brings forward uniformity throughout the institution. It is projected that the system will enable students a better academic control, displaying performance statistics throughout the academic year, making time management easier for each of their activities.

**Keywords:** Content manager, e-learning platform, mobile app, schedule manager, school system

## *Resumen*

*El uso de software gestor de contenidos en las universidades es un elemento necesario en la educación actual. Sin embargo, los gestores de contenidos disponibles en la red no siempre son la opción más idónea, dado que no ofrecen una integración personalizada a las necesidades de cada institución, o resultan ser obsoletos en áreas trascendentales, dejando de satisfacer los requerimientos prácticos de los estudiantes/profesores. La presente propuesta está enfocada en el diseño y desarrollo de un software gestor de contenidos integrado al entorno universitario, que tome en cuenta el uso del modelo académico empleado dentro de la institución correspondiente, haciendo hincapié en la interacción del alumno/profesor con el sistema. Este hecho brinda uniformidad a través de la institución. También se tiene contemplado que el software permita al estudiante un mejor control académico, mostrándole su desempeño conforme se avanza, facilitándole la administración del tiempo que da a cada actividad programada.*

***Palabras Claves:*** *Administrador de contenidos, aplicación móvil, agenda escolar, plataforma e-learning, sistema escolar.*

## 1. Introduction

To better understand the developed system is necessary to understand what a learning platform is. A learning platform is a software application that gives teachers, students, parents and administrators access to common resources, communication tools, and information, not just within the school but outside it too [Itslearning AS, 2011], this can be accomplished with the use of web and mobile applications that provide both the student and the professor with the necessary

tools to make academic life simpler, this is accomplished by allowing for the creation and submission of assignments such as homework, projects, test, etc., in an easy and convenient way. As of the writing of this paper, our academic institution makes use of the e-learning platform known as Moodle, having different versions of the platform depending on the Academic Unit. This gives a sensation of unconformity and incongruence for the students who move through different Academic Units, who have to create different accounts for each academic program and their learning platforms. This type of problems only grows in intensity when the platforms are different, where both the student and the professor have to make use of more than one of this platforms in order to carry out his/her everyday activities.

A clear example of this would be a professor who imparts classes in two different Academic Units, each of them making use of their own platform. In this type of cases, the professor would be required to manage two accounts (one for each platform), this not only becomes bothersome but also problematic, given that this type of situation would make it troublesome for the professor to keep an accurate record of each student's progress. Now imagine a student in similar circumstances, taking courses in distinct colleges, having to be conscious of each assignment in different platforms with different graphical interfaces and a completely different User Experience (UX).

The Learning Content Management Systems (LCMS) are applications designed primarily to support these academic tasks, however, many of them have no particularities and personalization in the environments where they are applied, some examples of these systems are WebCT, eCollege, Desire2Learn, Moodle, ATutor, Dokeos, Claroline, Sakai, LRN.

With the technological evolution in the educational field, new modalities have emerged with the use of information and communication technologies such as [Ramirez, 2011]:

- E-learning: Education and training through the Internet.
- B-learning: A mixed model that requires distance and face-to-face education.
- M-learning: The E-learning educational modality that is supported by mobile devices and wireless transmission.

These types of education require the support of a technological platform to manage virtual learning environments. Learning Management System (LMS) are these web-based software web applications that are used to plan, implement and evaluate learning processes related to online management and training in order to achieve superior academic and work performance.

Learning Content Management System is a software application that allows managing virtual learning environments with respect to educational content, didactic resources and both synchronous and asynchronous activities. Among these activities, the most representative are online evaluation, management, distribute and control users (teachers, students, administrators) in terms of enrollment, de-enrollment, and group management. Allowing users to follow up on their learning and teaching process through the user activity reports generated by the software system. The Learning Content Management Systems can be mainly of two types [Fernández, 2009], [Ramirez, 2011]:

- Proprietary systems. For instance, Blackboard-WebCT, eCollege, Desire2Learn.
- Open-access systems: For instance, Moodle, ATutor, Dokeos, Claroline, Sakai, LRN.

WebCT (Web Course Tools) is a commercial system of online virtual learning, which is used mainly by educational institutions for learning through the Internet. The flexibility of the tools for the design of classes makes this environment very attractive for both beginners and experienced users in the creation of online courses. In this sense, some research works have measured the reception and attitude of students towards the use of the WebCT virtual platform as a complement to face-to-face teaching in the Biotechnology Processes and Products subject of the Biotechnology Degree [Martín, 2010]. From the point of view of the teacher, the virtual platform led to greater interaction not only between teacher-student (through possible virtual tutorials) but also between student-student.

On the other hand, Atutor is an open-source on-line LCMS based on Web, and it has nice adaptability and utility. Tutors can assemble the courses on Web

immediately; they can also obtain and input packed courses easily and manage their on-line courses as well. Students are able to study under an adaptable circumstance. Atutor can easily set up course management platform. With the help of powerful, easily operating Atutor, we can save time, effort and money to set up a good online course management platform. It provides an effective supporting role for general course management staff. On the atutor platform, if the user has the super administrator's or teacher's permission, he can build up his own on-line course according to the actual demands [Jiugen, 2011].

The online learning platform, Moodle (Modular Object Oriented Dynamic Learning Enviroment), is presented as a software that offers the tutor, free of charge and without physically replacing it, some instruments that sustain and enrich their face-to-face classes [Ontoria, 2013]. One of the main features of this LMS is that it offers the possibility of inserting multimedia content: the teacher can upload images (photos, illustrations, graphics); videos (dynamic presentations, advertisements, short and feature films, documentaries, television news, television programs, trailers, video clips); and audio content (music, dialogues, radio programs). Due to the importance of sensory stimuli in teaching, the inclusion of multimedia elements in a Moodle course is essential to encourage motivation.

In terms of LMS designed with more particularities and focused on university principles, VirtualMed [Dominguez, 2007] is a open-use application encoded in PHP that uses MySQL as Database Manager. It has all the basic tools to be considered a content manager, such as: communication tools, performance, administration and productivity; as well as a friendly Web interface that supports the development of various forms of teaching organization. It can be installed in an educational intranet, with the aim of ensuring, maintaining and improving the quality of the educational teaching process, guaranteeing the conservation, transfer and reuse of knowledge; allows thus to take advantage of the advantages offered by technological development to all vital processes of scientific-technical development. The developed system presented in this paper solves most of this problems in a more useful way for both the student and the professor and even the academic institution, this by becoming more attached to the daily routine of the

student, providing him/her with the necessary instruments in order to stay aware of assignments and making communication with the professor something natural. In a similar way, it facilitates the professor the ability to manage and control all of the courses designated to him/her, even if they are under different colleges. Although undoubtedly one of the biggest incrustations of the proposed work is the version of the software for mobile devices with characteristics of alerts to the students of their activities, tasks, jobs, suspensions of classes and changes in schedules. In this scheme the student will have the ability to detect by himself the weak areas that she/he has and receive a personalized and autonomous support through intelligent models. The software project is still in the development stage in some modules and in the testing stage in another, so efficiency validation mechanisms at the user level are not yet available. It is hoped that in the medium term this information will be obtained to validate the operation and effectiveness of the application.

## 2. Methods

### Development

For the present project, the following requirements were taken into account:

- Must be a unique system within the university.
- Must be easy to use.
- Must provide a uniform UX experience throughout all the different Academic Units within the institution.
- Must provide a schedule/agenda linked directly to the student's courses and activities.
- Must be able to provide the student's record and progress to both the student and the professor.
- When there is an assignment such as a homework or project the student must be able to submit files to it, marking the assignment as complete.
- The system must be able to provide with mobile interfaces (apps) for iOS and Android as well as a web interface.
- The server must be able to escalate given the size of most academic institutions.

With these necessities, we decided to make use of the following technologies:

- Django & Django REST Framework [Django, 2018]; [Django REST, 2018]; [Geary & Horstmann, 2010].
- Python [Oliphant, 2007].
- Angular 5 [Angular, 2018].
- TypeScript [TypeScrypt, 2018].
- Ionic Framework [Ionic, 2018].
- Nginx [Nginx, 2014].

Each one of these technologies has their own advantages as well as their own jeopardies, nevertheless, we decided for them given the vast improvements they bring to the development of the system.

**Development methodology**

An agile methodology commonly known as Prototyping was chosen for the development of this system, this methodology is defined as a System Development Method (SDM) in which a prototype (an early approximation of the final system or product) is built, tested and then re-worked depending on how necessary it may be until we reach an acceptable prototype, on which a complete system or product can now be developed on top of [Rouse, 2005]. This type of model allows us for a feedback from some of the interested parties in early stages, these results on system features being discarded or suppressed while new functionalities and necessities were added as they were needed. For this specific project, we define a prototype as a system/software being used with demonstration purposes on a specific topic, testing its defined functionality in design so that we can have a better understanding of the problem to resolve. The prototype concept is defined as a preliminary version, intentionally incomplete of reduced of a system [Weitzenfeld, 2007]. Under this premises, the use of prototypes is a useful tool to apply to almost all activities of software design and creation.

For a more technical approach to our system, our prototypes are based on the design and requirement specifications under the following assumptions:

- Requirement gathering: is generated based on information provided by the user that needs or requires aid from a software tool.
- Requirement validation: The developed prototype is required to show and prove (if necessary) missing functionality according to the requirements obtained in stage 1.

Our prototyping focus follows the process indicated by figure 1. This procedure is repeated on each iteration [Sommerville, 2001]:

1. Establish prototype objectives.
2. Define prototype functionality.
3. Develop prototype.
4. Evaluate prototype.

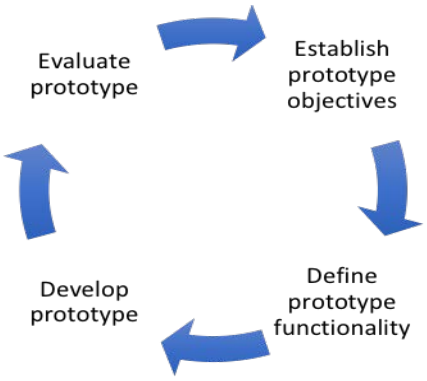


Figure 1 Prototype life cycle.

During the establishment of the prototype objectives, our main goal was to select one of the expected functionality to develop and determine how far the prototype would reach in terms of functionality. This task required from us a careful examination of the features that were required to be evaluated by the end user for their approval or dismissal. Each one of these steps allows us to make adjustments to the project even before a final product is shipped or built, nevertheless, this brings to the table its own risks and limitations. A risk that we had to face was the continuous change on requirements, given that none of the involved parties had a clear idea of how some of the functionality in the system should behave and there forward how it should be incorporated to the academic environment.

With the purpose of introducing the proposed application, figure 2 shows the functional architectural of the system Astrum. We choose the codename Astrum for the project given the anecdotal star some student gets when they do a nice job. At the moment both students and professors use the web client in order to interact with the system. This client was developed using Angular/Ionic in its most recent version. Aside from a few other common technologies (such as JQuery, bootstrap and Font Awesome, etc.) no special toolkit was used.

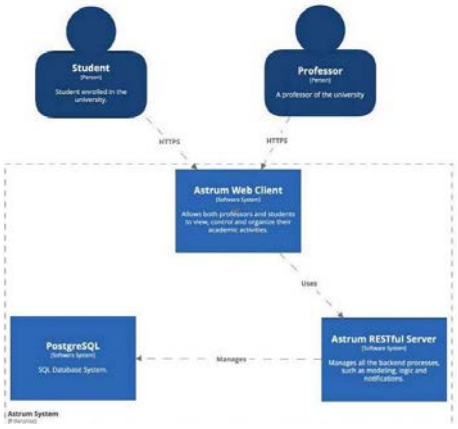


Figure 2 Functional architecture of astrum system.

This client, being only a shell, is not able to perform any operations by itself, it relies on the RESTful Server to provide all the information it needs, this way, when the mobile application is finished the same REST API can be reused.

The RESTful Server was build using Django. Its main functionality is providing all the logic of the system through a simple and consist REST API. In order to make development easier another library/framework (derived from Django) was used, this is the Django rest framework. This framework provides an easy way of creating REST APIs and parsing data. The RESTful Server is stateless, so all the data is stored in a database server with PostgreSQL serving as the Database Manager.

## 3. Results

The Astrum system modules contemplated so far cover those shown in figure 3. The three main areas of the system cover administration (performed by the site

administrator), teaching (dedicated for use by instructors) and learning (focused for students). The modules with dark background mean elements in test phase (already developed modules). The modules with white background involve elements that are still in the development stage; while the modules with white background but with a red line represent modules that are characteristic of this system, since they focus on activities that are not integrated into the LCMS systems of related works. These last models are what distinguish our system. The idea is to support students understand the topics of a subject they do not understand well and help them in a personalized way through specific didactic tools and with case-based reasoning approach and autonomous agents (intelligent multi-agent system).

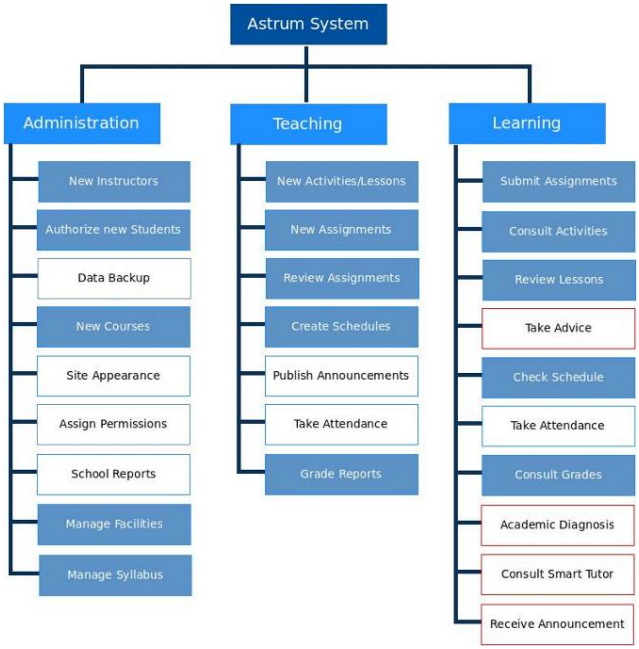


Figure 3 Astrum system modules.

It was decided that the functionality of the Web App would be reduced in comparison to the functionality found in the mobile application (which is still under development). Therefore, the web app would only be focused on administrative tasks, figure 4 shows one of the CRUDs (Create, Read, Update and Delete) of the system, likewise, it is possible to get a glimpse of the school agenda for the student (figure 5) which both a professor and a student can have a look upon.

In addition, figure 6 (in complement with figure 7) shows a part of the menu observed in the web application, making reference to the options institutions have over the system, such as managing Academic Units (colleges), Classes, Courses, Study Plans and Classrooms found within itself.

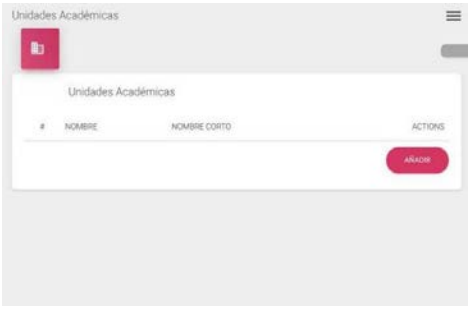


Figure 4 CRUD for academic units (colleges).

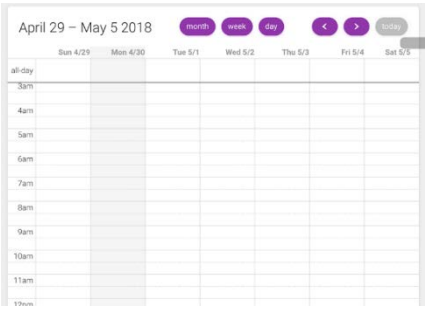


Figure 5 Agenda view on the Web App.


Figure 6 Menu for Institutions on the Web App.

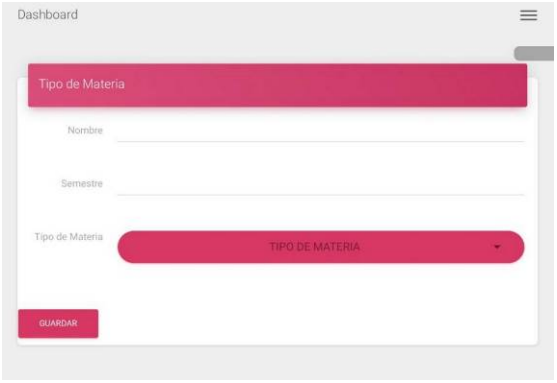


Figure 7 Course creation view.

Currently, the system has a database under the scheme shown in figure 8. This database has suffered some major changes over the course of the development of this system, differing greatly from our original conception. We expect the database to continue growing and getting more complex given the essence of the project. This database accounts for data from system users, permissions, professors, schools (colleges), courses, classrooms, study plans, classes, assistance, score tracking and assignment of homework.

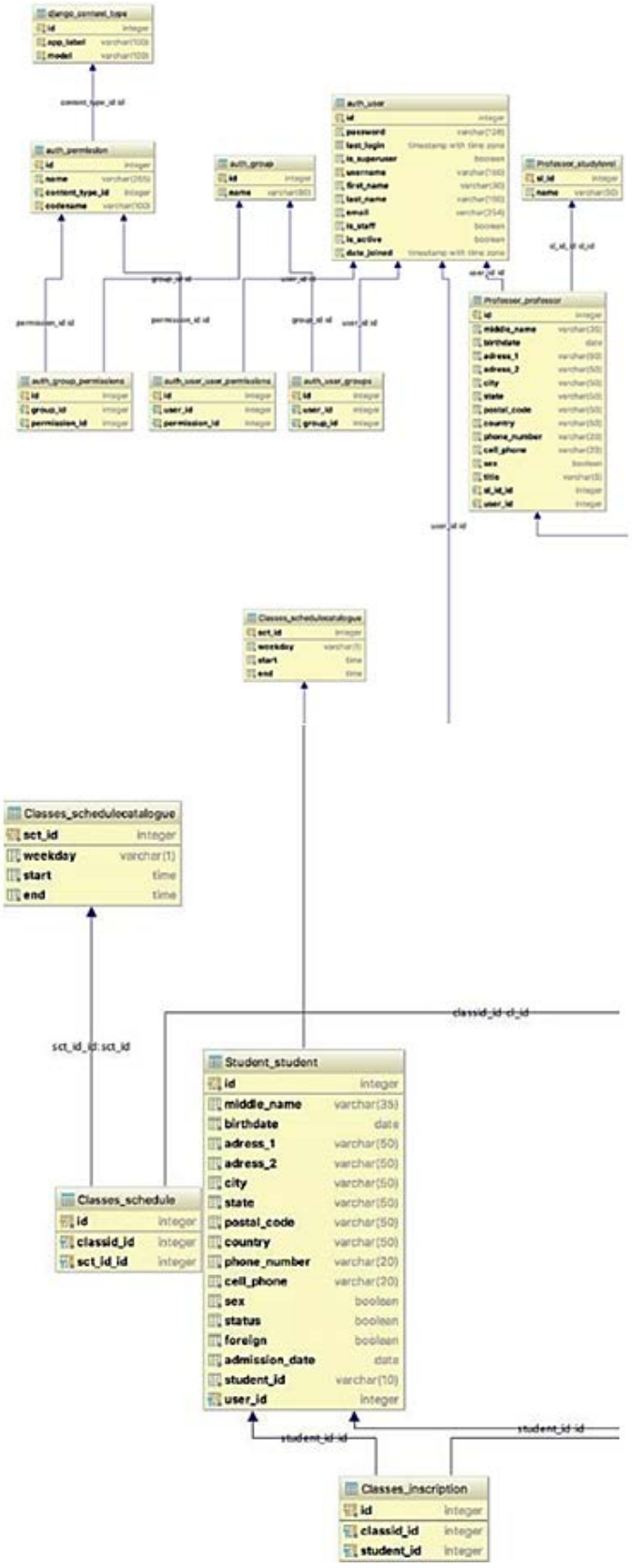


Figure 8 Database scheme.

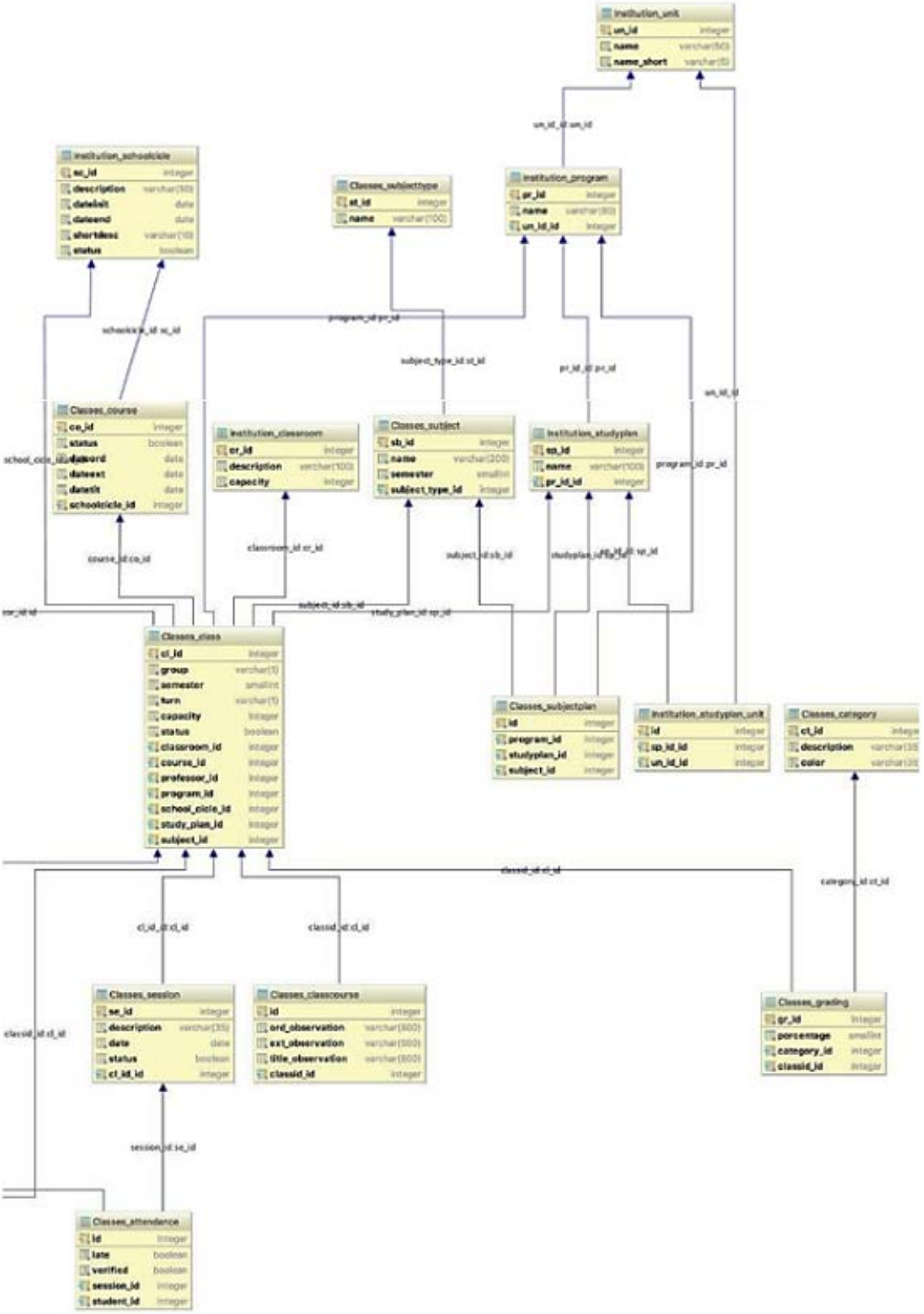


Figure 8 Database scheme (continued).

## 4. Discussion

At the moment, the system is in the testing phase with the modules currently developed. The idea is to continue carrying out feedback work as this beta version is being used. With the time it has in operation, it is not yet possible to determine a set of data that provides information regarding its effectiveness or radical benefits of the system. The one-year projection of the system will allow us to obtain a reliable set of data to analyze and visualize the real areas of benefit and the weaknesses to be corrected.

Under the working framework of the functionality of prototypes presented in this section, we must remember and take into account the portability and considerations of future changes in every software design and development [Weitzenfeld, 2007], which is represented in table 1. The "interfaces" represent the graphical elements, the "functionality" hold the business rules (user requirements), the "data" and "functions" are the equivalent to internal behaviors used to describe "objects" (corresponding to the basic data structures of object-oriented programming), meanwhile the "information" outlines the problem domain of the application. It is important to highlight that the system's architecture must distinguish between elements with higher and lower probability of change. As well, the development of the software system must contemplate a model of processes in which the components with greater change probability do not "drag" more stable elements. Thus, the functionality catalog and tools provided by the shown prototype(s) produce a constant change in their design to better conform to the changing moments in the in-person and virtual (online) approach.

Table 1 Probability of further changes in the software according to the type of design element.

| Element | Change probability |
|---|---|
| Interfaces | High |
| Functionality | High |
| Data | Medium |
| Functions | Medium |
| Objects | Low |
| Information | Low |

## 5. Conclusions

Although a challenging project at its early stages given the low understanding and knowledge of the team in certain fields and technologies, the benefit far exceeds the challenges encountered, both for the system and the development team. The system is expected to have extensive integration within our alma mater, given that it's based mostly on our alma mater's structure and its daily routines.

Although there is still much left to do, the general idea of the system is more clear now compared to what was expected at the beginning, and it's our hope to be able to add more functionality in the following iterations. We are expecting a progressive deploy within the university. This will allow us to make adjustments to the system depending on the necessities or unconformities that arise.

The system's growth will be based in the requirements of the academic staff of the institution, but more importantly, it will be focused on the needs brought forward by the students, who are the beneficiary of most of the system's qualities.

## 6. Bibliography and References

[1] Angular. What is Angular? Consultado el día 29 de abril de 2018. Dirección web: https://angular.io/docs. 2018.

[2] Django. Django, the Web framework for perfectionists with deadlines: https://www.djangoproject.com/, 2018.

[3] Django REST. Django REST Framework: http://www.django-rest-framework.org/m, 2018.

[4] Dominguez J., y Fresno C. VirtualMEd: un sistema gestor de contenidos de aprendizaje para la universalización de la educación superior. ACIMED, No. 15, Vol. 1, 2007.

[5] Fernández-Pampillón, A. M. Las plataformas e-learning para la enseñanza y el aprendizaje universitario en Internet. In Las plataformas de aprendizaje. Del mito a la realidad. Biblioteca Nueva, Madrid, 2009.

[6] Itslearning AS. The 12 key benefits of learning platforms. How learning platforms support the business of teaching and learning: https://www.ttu.ee /public/h/haridustehnoloogiakeskus/img/12_reasons_web_final-3.pdf, 2011.

[7] Geary, D., & Horstmann, C. S. Core JavaServer Faces. Prentice-Hall. New Jersey, EUA. 2010.

[8] Ionic. The top open source framework for building amazing mobile apps: https://ionicframework.com/framework, 2018.

[9] Jiugen Y., Ruonan, X. y Luyao, Y. The Application of the Atutor Learning Content Management System in Teaching. 5th International Conference on Distance Learning and Education, 2011.

[10] Martín, M. A. La plataforma virtual webct como complemento de la docencia presencial en la asignatura procesos y productos biotecnológicos. Jornadas sobre Innovación Docente y Adaptación al EEES en las Titulaciones Técnicas. 171-174, 2010.

[11] Nginx. Modules reference: https://www.nginx.com/wp-content/uploads /2014 /03/nginx-modules-reference-r3.pdf, 2014.

[12] Oliphant, T. Python for Scientific Computing. Computing in Science & Engineering. No. 9, 10-20, 2007

[13] Ontoria, M. La plataforma Moodle: características y utilización en ELE. XXIV Congreso Internacional de la ASELE, 2013.

[14] Ramírez, M. Gestión de los Sistemas de Gestión de Aprendizaje: https://milagrosrp.wordpress.com/2011/10/24/gestion-sistemas-gestion-aprendizaje/#comments. 2011.

[15] Rouse, M. What is Prototyping Model?: https://searchcio.techtarget.com /definition/Prototyping-Model, 2005.

[16] Sommerville, I. Software engineering. Addison-Wesley. Boston, USA. 2001.

[17] TypeScript. TypeScrypt: Notes for Professionals: https://goalkicker.com /TypeScriptBook2/, 2018.

[18] Weitzenfeld, A., & Guardati, S. Capítulo 12: Ingeniería de software: el proceso para el desarrollo de software. Libro: Introducción a la Computación. CENGAGE Learning. México, D.F. 2007.