Development of a web based instrument on higher education structures of industrial engineering

*Ioan - Cristian Tarba**

Manufacturing Engineering Department, Politehnica University of Bucharest, Splaiul Independentei 313, Romania

Abstract. The research and development of assisted operational instruments on higher education structures of industrial engineering represent a continuous and complex process. The present paper contributes to the building up of support elements and an assisted operational instrument on higher education structures of industrial engineering, with focus on the specific curricula. The use of tested and validated constructive solutions from other projects, as base for the new design, reduces the design time.

1 Introduction

The quality and the reliability of the computing systems are essentially determined by the software applications. The complexity of the software was and still is a challenge to those who design, implement, maintain, run and develop the systems, even the simple ones. Thus, as a new instrument, the methodology for software development consists in rigorous approach, as a plan, on the work style, technique of code writing, within the teams, between which the coordination and communication become essential [1].

Within methodologies for software development, there are traditional (formal) and agile approaches [1-3]. During the present research, the following methodologies have been applied: cascade methodology, evolutionary methodology, spiral methodology and the agile methodologies - SCRUM [4] and Extreme Programming [5].

The technologies used for the development of software applications are selected based on a various number of criteria, because of a large variety of programming languages [6, 7], database systems and solutions for the user interface [8].

The modular web applications are more and more developed [9, 10].

The algorithms' building is an important part of web applications development [11].

2 Objective and method of research

The main objective of the present work is to configure a modular web application as assisted design instrument to support the operational actions on curricula, especially in the framework of higher education industrial engineering structures, with the main aim to offer a tool for the

* Corresponding author: cristian.tarba@imst.pub.ro

managers of the departments to elaborate the curricula needed for education process and to reduce the time for getting the curricula done. During the research there were discussions about the input data and expected results of the developed application. The actually tools used for elaborating the curricula were spreadsheets and the time for completing them were critically affected by other factors. Thus, the time for elaborating the curricula using spreadsheets method cannot be a reference and it is obvious that the developed tool will reduce this time considerably.

The research activity has been structured with regard to the following reference elements: selection of the programming languages; building an algorithm, the structure and content of the application; conducting a case study in a real higher education institution.

3 Design of the objective modular web application

The objective modular web application is named Integrated Operational Instrument – Web Application, in short IOIAW.01.

The development of the IOIAW.01 application is based on an open source platform provided by a Linux distro, CentOS 7, and the selected solution for the persistence of data which is MongoDB database system. As development technologies, the followings have been applied: Python and JavaScript, as programming languages; Flask framework [12] with its 2 modules: Werkzeug, which is a development web server and powerful debugger, and Jinja2, which is a language for creating templates; Bootstrap framework used for the user interface design.

The IOIAW.01 application provides with the possibility for the user to access the process of adding new curriculum, modifying the existing ones, adding/ modifying/ deleting the disciplines which belong to curriculum. Because the decisions regarding the content of a curriculum are to be taken by the persons in charge within the higher education institution, it was developed a routine for managing the users and the roles assigned to them. Thus, when the user login in he/she will have access only to those collections in the database which they have permission assigned to.

The IOIAW.01 application is developed structured on 4 sections: View Curriculum, Modify Curriculum, Add New Curriculum and Copy New Curriculum, presented in Fig. 1.

Each of the defined section within the application can be accessed in its homepage, as presented in figure 1.

Fig. 1. Homepage of the application.

The scheme block of the software system as well as the work flow of the data through it are presented in figure 2.

Fig. 2. Scheme block of the software system.

3.1 View Curriculum section

The purpose of the View Curriculum section and of the algorithm developed was to return to the user the view of a curriculum of which the identification data was entered in a web form. There were defined the route, /search_curriculum, the view function, search_curriculum()

and the html template, `filter_curricula.html`, which is used to generate a form in order to get the data from the user, as presented in figure 3, and the route `/view_curriculum/<curriculum_id>`, the appropriate view function, `view_curriculum(curriculum_id)`, and template `curriculum.html`, which leads to displaying the curriculum, as presented in figure 4.

Fig. 3. Curriculum search form.

The steps the algorithm performs are the following:

1. The route `/search_curriculum` is accessed in the web browser;
2. It is called the view function `search_curriculum()`:
 - a. A connection to the database is created and retrieved the appropriate collections: `curricula`, `study_cycles`, `study_domains`, `study_programs`, `departments`, `faculties`;
 - b. There are created the lists for populating the form elements in the html template, such as: `yearslist`, `cycles_list`, `domains_list`, `programs_list`, `study_years_list`, `departments_list`. Any combination of the 6 elements in these lists could be valid because they are already in the database collections. They are also called the general identification data for a curriculum;
3. The selected data in the web forms will form the id of the curriculum and used to search the database collections for an existing curriculum identified by this calculated id;
4. The database will be searched for the existence of the calculated id:
 - a. If the id was not found in the database, the system will raise an error and will redirect the user to the search form page to modify the general identification data for a curriculum in order to calculate another id;
 - b. If the id was found in the database, the user is redirected to the route `/view_curriculum/<curriculum_id>` and the view function `view_curriculum(curriculum_id)` is called:
 - i. There are retrieved from the database the needed collections: `curricula`, `disciplines`, `departments`;
 - ii. The documents of the collections `curricula` and `disciplines` are filtered used curriculum id as filter.
 - iii. There are created lists (`study_programs`, `program_name`, `study_domains`, `domain_name`, `coordinating_dep`, `dep_name`, `academic_year`, `study_year`, `disciplines`, `semester`, `opt_disciplines`) that are passed to the html template, named `vieplan.html` to appropriate display the data, as presented in figure 4.

Fig. 4. View curriculum.

3.2 Modify Curriculum section

The section *Modify Curriculum* refers to the process of altering the content of a curriculum, but not the general identification data of a curriculum. In fact, modifying the curriculum consists in adding, editing, deleting the disciplines and their characteristics. The application administrator will have the task of deleting the general identification data of a curriculum.

There were defined the route, `/search_edit_curriculum`, the view function, `search_edit_curriculum()` and the html template, `filter_edit_curricula.html`, which is used to generate a form in order to get the data from the user, and the route `/edit_curriculum/<curriculum_id>`, the appropriate view function, `edit_curriculum(curriculum_id)`, and template `edit_curriculum.html`, which leads to displaying the curriculum, as presented in figure. 5.

Fig. 5. Modify a curriculum.

The algorithm for searching and displaying a curriculum for editing is similar to the algorithm presented for View Curriculum section. In addition there were defined the route `/add_discipline/<curriculum_id>`, the view function `add_discipline(curriculum_id)` and the html template `add_discipline.html` which will offer the user possibility to add new disciplines to an existing curriculum identified by `curriculum_id`. Also, for each discipline there were added actions to edit or delete it. Thus, for the edit discipline action were defined: the route

/edit_discipline/<discipline_id>, the view function edit_discipline(discipline_id) and the html template edit_discipline.html. For the delete discipline action were defined: the route /del_discipline/<discipline_id>, the view function del_discipline(discipline_id) and the html template del_discipline.html.

4 Conclusions

The IOI.AW.01 application has been developed and implemented as assisted design instrument to support the operational activity on curricula development associated to higher education industrial engineering structures.

As for the methodology adopted in the research, it was a combination of a traditional one (waterfall) and an agile one (extreme programming). The requirements of the final product were established at the beginning of the process, but later, the whole process of the development of the software system was according with the four basic activities described by extreme programming methodology: coding, testing, listening and designing.

The algorithm was developed with possibilities of future improvements such as validating the curricula against the nationwide regulations, building statistics regarding different elements of a curriculum, implementing the application in more fields than industrial engineering field.

The chosen solution for the data persistence is a NoSql database, which uses rules defined by the developer to deal large amounts of unstructured data

The low working time and high performing quality represent relevant performance indicators of IOI.AW.01 application.

This work is supported by the Sectorial Operational Program for Human Resource Development, financed from the European Social Fund and the Romanian Government under the contract number POSDRU/159/1.5/ S/137070.

References

1. D. E. Avison, G. Fitzgerald, Information Systems Development: Methodologies, Techniques and Tools (1995)
2. J. Highsmith, Agile Software Development Ecosystems (2002)
3. J. Highsmith, Agile Project Management: Creating Innovative Products (2010)
4. K. Schwaber, M. Beedle, Agile Software Development with Scrum (2002)
5. C. Novac, Extreme Programming - a Challenge for Software Developers, Revista Informatica Economică, **31**, no. 3: pp. 80 – 83 (2004)
6. V. Loganathan, PySide GUI Application Development (2013)
7. D. Ascher, A. Martelli, A. Ravenscroft, Python Cookbook (2005)
8. J. Anderson, J. McRee, R. Wilson, Effective UI: The Art of Building Great User Experience in Software (2010)
9. D. Esposito, Modern Web Development: Understanding domains, technologies, and user experience (2016)
10. I. Maia, Building Web Applications with Flask (2014)
11. D. Baldwin, Algorithms and Data Structures: The Science of Computing (2004)
12. M. Grinberg, Flask Web Development: Developing Web Applications with (2014)