

Software Application for Documents Management in Higher Education

Cătălin-Iulian CHIVU

Transilvania University of Brasov, Romania, catalin.c@unitbv.ro

Catrina CHIVU

Transilvania University of Brasov, Romania, catrina.c@unitbv.ro

Abstract

Study program management is a very time-consuming process, especially due to large number of documents to be made. On the market, Romanian or international, there is no commercial software that may help. Software application developed by the authors is designed to automatically generate documents required in quality evaluation of the university study programs (with direct application for distance learning study programs) based on curricula. Using this software, half of the documents are generated 40% faster than are generated before. In addition, a great advantage of it would be the opportunity of using it on an eLearning platform.

Keywords

software, informatics application, higher education, documents management

1. Problem description

In Romania, each study program has a coordinator, a teacher that manages the relation between students and Department, Faculty and University management.

Each study program is annually audited by university and, depending on the characteristics of the program by ARACIS (Romanian Agency for Quality Assurance in Higher Education). Romanian standards stipulate [1] that quality evaluation is carried out in the following situations:

- a) for the temporary functioning authorization of a study program (program authorization) or of a higher education service provider (institutional authorization);
- b) for the accreditation of a study program (program accreditation) or of a higher education institution (institutional accreditation);
- c) for the periodical certification, every five years, of the academic quality of education and research services from an accredited university.

The main problem of these processes is high number of documents, most of them intercorrelated and interconnected. Most of the documents regarding curricula are done by teachers that should know all the interconnection, thus, to correctly complete documents. Unfortunately almost on each evaluation process, there are mistakes in these interconnections and, depending on gravity of these mis-correlations, the program could obtain the non-authorization / non-accreditation.

If the syllabus is a long-distance one, number of documents to be prepared by the teachers is even higher. This is the reason why it is mandatory to be developed an integrated software application for all these documents.

Each university from Romanian have software applications for grades management, curriculum management, payrolls. These applications are developed inside universities, not a national software generally available.

Authors of the present paper developed an integrated software application designed especially for long-distance programs, but applicable for any other kind of study program. The application is easy to be used, with user-friendly interface, and allows the user to introduce data from Microsoft Excel and to export documents in Microsoft Excel or Microsoft Word format [2, 3, 5]. In this way, the documents will correspond to the standards' requirements and also may be used on any computer and even by a less experienced user.

2. Software Design

2.1. Information flow

Software application was developed in C# with user friendly interface. To be able to implement entire application, there should be done some interconnections between it and university software or should be posted on a private platform. Combination C# and .NET is an ideal solution to develop such an integrated application [3, 5].

.NET is a developer platform made up of tools, programming languages, and libraries for building many different types of applications. ASP.NET extends the .NET developer platform with tools and libraries specifically for building web apps.

To be able to create application in Visual Studio there are some conditions that should be satisfied: the project should be an ASP.NET one, written in C# language and the platform for which it is written should be Windows [3, 5].

The application has many modules, interconnected in the stage of output documents, modules oriented on long-distance documents:

- long-distance curriculum generate based on full-time undergraduate program;
- syllabus for each course based on curriculum and teachers' inputs;
- schedules: global, student, teachers, each course.

Logic diagram of the software (Figure 1) contains the block components that define subroutines of the program, the input data in each block, including type of documents that should be uploaded by the users and output documents, including type of these documents. Output documents are, generally, templates in according with either university quality management systems [1, 4], available in current year, or the ARACIS.

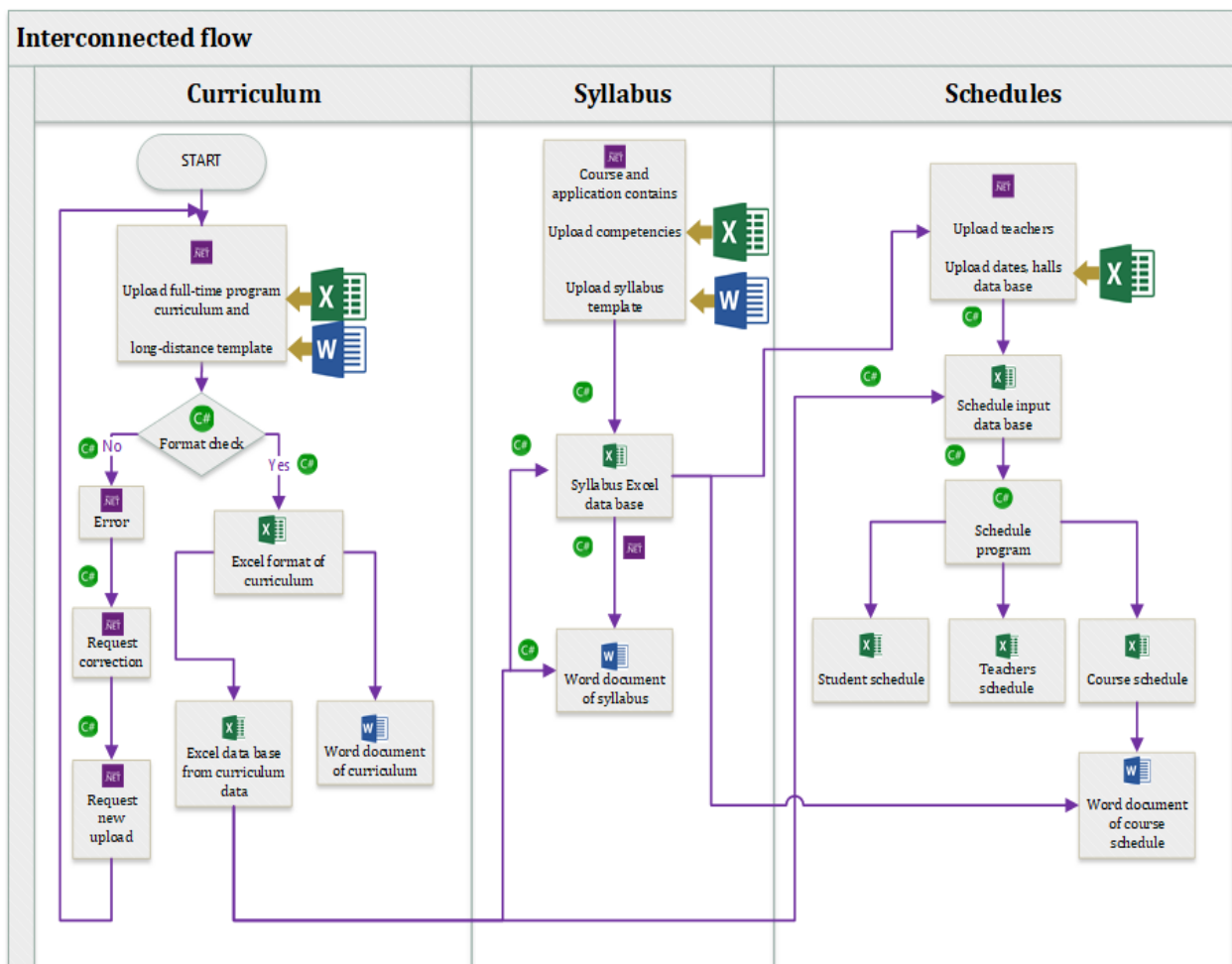


Fig. 1. Interconnected flow of the software

2.1. Operation mode

To be able to understand how software works, in the following there is a brief description of operation mode, which include connections, checking principles, errors, etc.

Curriculum module. The main purpose of this module is to obtain curriculum for a long-distance program, based on full-time undergraduate program. In Romanian, the two curriculums do not look the same, includes the same courses and number of hours, but there are different structured. In Table 1 there is shown the conversion and also the differences between the two curriculums.

Table 1. Differences and similarities between curricula

Characteristics	Full-time program	Long-distance program
Disciplines included	all	all (same number and type)
Hours computing method	per week	per semester
Course (example) activity	1 h: 1C	14 h: 14SI (14 IS - individual study) for 14 weeks semester
Seminar (example) activity	1 h: 1S	14h: 14 h(AT+TC) (TA - tutorial activities + H - homework) national usual conversion is: $14h = 4h(AT) + 10h(TC)$, but is not mandatory, standards require a minimum 4 h of AT (TA) for 14 weeks semester
Laboratory and project (example) activity	1 h: 1L/ 1P	14 h: 14h AA (assisted activities) if a course has both laboratory and project, the number of AA is the sum of those

The inputs document of this module are:

- full-time program curriculum in Excel format (save either .xlsx, or .csv);
- long-distance program curriculum template in Word document.

Excel file that includes full-time program should have row header made of: name of course; semester; type of course; mandatory / optional course; number of hours per course, seminar, laboratory, project; ECTS; type of final evaluation; abbreviation of competencies for each course.

Interface of the module has:

- upload zones: for full-time program curriculum (Excel file) and long-distance program curriculum (Word file);
- zones for: number of semesters and number of weeks for each semester;
- zone for *seminar* conversion rule applicability: for all courses or each discipline has its own rule;
- zone for *seminar* conversion formula: *usual* ($1hS = 4hAT+10hTC$); *particular* (in this case the conversion formula / formulas should be given – if there are many formulas the program will number them and make a data base with them).

The *format check* step of this module verifies if:

- on the interface there are given the number semesters and number of weeks on each semester;
- on the interface, in the case of *seminar* activity, is selected if conversion rule is the same for all courses; if not, when the conversion is made by program, it will be requested, on each course to choose the conversion formula;
- on the interface is selected conversion type of seminar activity: *usual* or *particular*; if *particular* is checked, program verifies if the specific formula / formulas is / are given;
- in the Excel file with full-time program curriculum: there is no blank line; each course has hours at least one activity; all course have type according to standards and mandatory / optional status mentioned; all courses have ECTS and final evaluation specified; all courses have mentioned at least one competence abbreviation;
- in Word file there are defined template areas.

Error and *request correction* steps are made of messages windows of the application that will show the errors that appear (there is a message related to each check done in *format check* step). Thus, user is requested either to go back to the interface to check or give information, or to upload files (Excel or Word) according to requirements.

Outputs of the module:

- data base with courses, hours, characteristics;
- according to Romanian quality standards, at the end of curriculum there are so called summary of hours based on different criteria; standards impose some values for these totals; software application is doing these totals and checks compliance with standards;
- Word document that contains curriculum.

Syllabus module has as one input data base from **curriculum** module. The main purpose of this module is to obtain a correct syllabus for each course, according to quality standards of universities and ARACIS.

The inputs document of this module are:

- competencies of program, according to national standards: Excel file with row head made of: competences' abbreviation and detailed description of each competence (abbreviation);
- data base from curriculum module: Excel file that contains name of courses; semester; type of course; mandatory / optional course; number of hours per course, seminar, laboratory, project; ECTS; type of final evaluation; abbreviation of competencies for each course;
- template of syllabus (according to university quality standards and ARACIS): Word document as template document.

Interface of the module has:

- upload zones: for competences file and for syllabus template;
- selection area: based on curriculum module, on the interface of this module there is automatically generated a multi-selection list that contains courses and type of activities for each course;
- input zones for course, seminar, laboratory and project description (for each input is required number of hours) based on selection area;
- input zone for evaluation method, requirements and description;
- input zone for staff: name and status of coordinator and name and status of tutor / tutors.

Before is generate *Syllabus data base*, program checks:

- if there is perfect correspondence between abbreviation for competencies from curriculum module and file uploaded in the current step;
- if the total number of hours introduced for each activity corresponds to curriculum;
- there is introduced an evaluation method and description;
- there are introduced the name and status of coordinator and tutor / tutors, according to standards and university image policies.

Outputs of this module are:

- word document for syllabus of each course;
- data base (Excel) with the following information for each course: data regarding coordinator and tutor / tutors; description of each course and hours for each activity.

Schedules module is designed especially as an instrument to elaborate students' schedule and based on this to obtain the other standards schedules. For long-distance programs, ARACIS standards require, as mandatory document, a schedule for student in a special template. In fact, this schedule is a combination between a classical one (which include days, hours, teacher, hall) and syllabus. This particular template of the schedule is a very important document in management of long-distance program and is one of those documents that most often contains mistakes.

The inputs document of this module are:

- data base from curriculum and syllabus modules (as Excel files);
- Excel file that contains: halls, dates, students groups;
- Word template for student's schedule.

Interface of the module has:

- initial interface: upload zone for Word template for student's schedule and launch schedule

subprogram;

- schedule design interface: allows to design schedule of the student and create outputs as Excel files with schedules of student, teachers (Figure 2). This interface allows user to upload Excel data base from schedule module, and Excel file with halls, dates, etc. Besides these upload options, there are ListBoxes that allow, based on uploaded files to select students groups, courses, type of activity, teacher, hall.

The output documents are schedules both classical and those required by internal quality system or national standards.

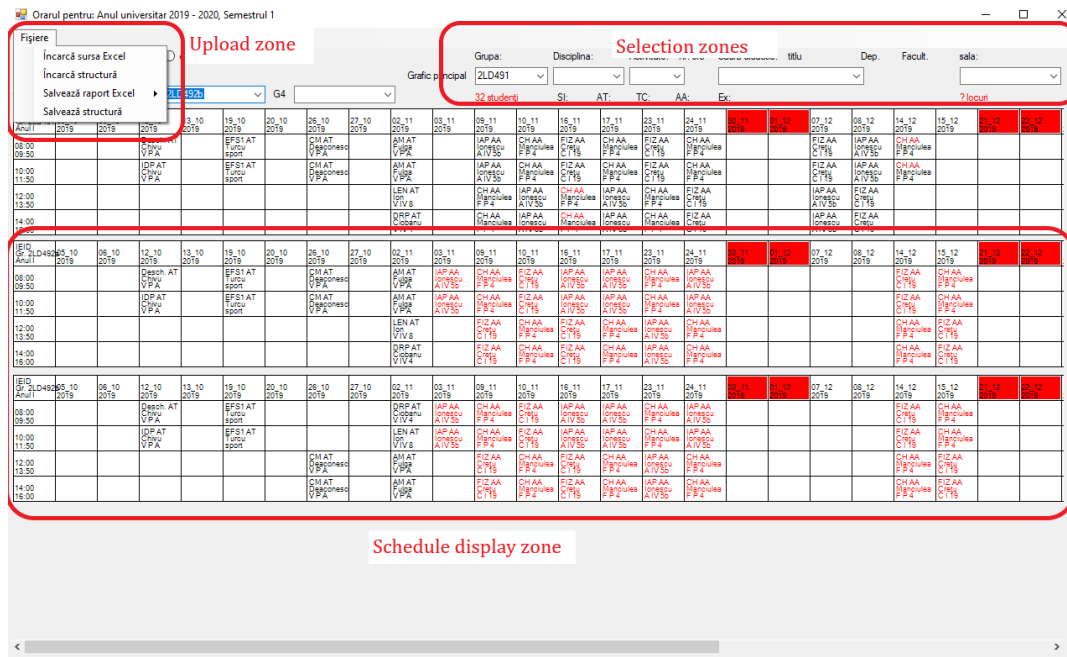


Fig. 2. Schedule design interface

Schedule subprogram has a lot of advantages:

- colours in real time if there is any type of unallowed overlay (teacher has already classes in the same position, or hall is already occupied, or course has no more hours to schedule);
- allow user to see all the groups of the same year;
- red colour the weeks that are not available, being legal free days or holidays;
- yellow colour evaluation period.

4. Conclusions

During evaluation process the time spent to complete the documents regarding curriculum, syllabus and schedules is, based on the study program coordinator experience around one and a half month. This period is so large because the involvement of the teacher that should complete the syllabus, then this is corrected by the coordinator and these actions repeats at least twice. Besides, curriculum and syllabus should be completed annually. Using software application, the process is reduced to one and a half week.

References

1. <http://www.aracis.ro/en/procedures/>. Accessed: 2019-10-21
2. Čevere R., Sproge S. (2010): *Application of software quality models in evaluation of study quality*. Problems of Education in the 21st Century, ISSN 1822-7864, Vol. 21, p. 37-46, <http://www.scientiasocialis.lt/pec/files/pdf/vol21/37-46.Cevere.Vol.21.pdf>
3. Kendal S. (2011): *Object oriented programming using C#*. Bookboon Publisher, ISBN 978-87-403-2140-1
4. Sadler D.R. (2017): *Academic achievement standards and quality assurance*. Quality in Higher Education, ISSN 1470-1081, Vol. 23, is. 2, p. 81-99, <https://doi.org/10.1080/13538322.2017.1356614>
5. Spaanjaars I. (2014): *Beginning ASP.NET 4.5.1: in C# and VB*. Wiley Publishing House, ISBN 978-1-118-84677-3