



The Open Student Information System



Performing Academic Processes

Hildeberto Mendonça Filho

OSIS - Performing Academic Processes

Hildeberto Mendonça

© 2016 Hildeberto Mendonça

Contents

1. Introduction	1
2. Admission and Registration Process	3
2.1 The Application Form	3
2.2 Reviewing Applications	4
2.3 Becoming a Student	4
3. Student's Path	5
3.1 Assessments	5

1. Introduction

OSIS (Open Student Information System) is a lightweight and yet highly scalable system to manage the core business of higher education institutions. It is free and open source under the terms of GPL v3 public license, aligned with the strong spirit of knowledge sharing and collaboration of the community. The project is funded by Universit  catholique de Louvain in the context of its long term goal of modernizing its information systems towards the next generation of students, devices, practices and technologies.

By using OSIS, a higher education institution is able to maintain, control and monitor the entire student s path, from the admission to the diploma and all information orbiting this fundamental process, such as planning the academic year, managing partnerships with other institutions for exchange and mobility programs, assigning teachers to courses and courses to students, etc. At the current stage, OSIS is designed to fulfill UCL s needs, but it can be properly adapted to fulfill more needs coming from other institutions interested in joining the project.

Offering a reusable system is not the only advantage of being open source. It also means:

- fully transparent operations for the entire academic community;
- inspiration for entrepreneurs to build integrated services that may be, at some point, incorporated into OSIS;
- empowering departments, endowed with the necessary expertise, to develop extensions to fulfill their particular needs;
- opportunity for students looking for a job, but with no previous experience, to participate in a real world project;

- reduced maintenance cost for the University thanks to the range of reliable open source products required by the system, such as the web server Apache, the database PostgreSQL and the web framework Django.

2. Admission and Registration Process

In this chapter we discussed the admission and the registration processes, targeting those people who want to become students of the institution.

2.1 The Application Form

Uploading Documents

While filling up the application form, the applicant is required to upload digital copies of several documents. The following formats are accepted: GIF, JPG, PNG and PDF. For each type of document, the applicant is able to upload one or more files, depending on how it was digitalized.

In most cases, a document is fully contained in a single file. The applicant uploads it and the document becomes part of the application. But there are two special cases to consider:

1. the content of the required document is spread in multiple files. It happens when the applicant scans each page in a separate file and did not manage to easily merge those files in a single one.
2. the content of several documents is contained in a single file. It happens when the applicant scans several documents in a sequence and acumulates all pages in a single file.

In the first case, the applicant may upload multiple files, relating them to a type of document, and the application merges all of them

in a single PDF file. The applicant should carefully upload the files in the same sequence of the pages in the originals so the application can produce a consistent merged result.

In the second case, the applicant may upload the same file multiple times, once for each type of document it contains, and specify the pages where the document starts and ends in that file. It is valid for PDF files only because they are paged and image files aren't.

When the applicant downloads what (s)he has uploaded, (s)he gets what was uploaded, except when multiple files are uploaded to a single document. In this case, only the merged version is downloadable.

2.2 Reviewing Applications

For the application reviewer, dealing with dozen files for each application review is a tough work to do. To simplify the document reviewing task, all documents attached to an application are merged into a single PDF file with a table of content, for easy access and navigation.

2.3 Becoming a Student

3. Student's Path

In the previous chapter, we discussed the admission and the registration processes, targeting those people who want to become students of the institution. In this chapter, we start from the moment those people effectively become students and begin their path in the institution. We explain how OSIS manages the students' path until obtaining their degree.

3.1 Assessments

Score Encoding

The score encoding application allows **tutors** (teachers, assistants, coordinators) and **program managers** to encode the scores of the students after the exams. It is a critical process because the correctness of the scores determines the success or failure of the students, a career path changing.

There are three roles in the encoding process:

1. *Teacher or assistant*: tutors with direct contact with students and with the duty of conducting the exams, can encode and double encode the scores, but cannot submit them to the program manager.
2. *Coordinator*: tutor with the right to encode, double encode and submit the scores to the program manager.
3. *Program Manager*: can encode and double encode scores with the additional privilege of modifying the final scores always when necessary. The program manager is in a faculty level, keeping track of the encoding progress and making sure

the process is correctly performed. Because of his(er) special rights, a program manager cannot accumulate the roles of coordinator, teacher or assistant. If it happens, the application will always consider the role of program manager and ignore the others.

This process happens a few times a year from the beginning of the exams to a day before the deliberations. The students that are enrolled in the programs, enrolled in the learning units, and enrolled in the exams in the current academic year are impacted by the process. Teachers, assistants, and coordinators that are assigned to the courses as well as managers assigned to programs are also impacted by the process.

When the exam happens and the tutors finish assessing the students, then they can start encoding the scores into OSIS. There are three methods of encoding scores:

1. *on-line encoding*: the tutor fills in an on-line form for each one of his courses, encoding the scores directly in the application and making them immediately available for the faculty.
2. *spreadsheet encoding*: the tutor downloads a spreadsheet file for each one of his courses. The spreadsheet contains a list the enrolled students. Then (s)he fills in the spreadsheets with the scores and uploads the files back into OSIS to register the scores.
3. *paper encoding*: the application generates an A4 printable PDF file for each course and the tutor fills in the scores and send the paper form to the faculty. Then the program manager get busy encoding scores from paper to the on-line encoding method.

The on-line encoding is definitely the one that allows the process to move faster because the validation and communication are fully

automated by OSIS and there are less complexity and bureaucracy to deal with. But to use this method it is necessary to be connected to the internet without down-time. Optionally, tutors may use spreadsheets to work offline and to perform further calculations, enriched ordering and filtering. But it requires more attention to avoid modifying fixed data used by the import mechanism. Undiscriminated changes in the spreadsheet may cause excessive import errors, wasting valuable time with fixes that have nothing to do with the encoding in itself. At last, the paper form is available anyway to cover unforeseen situations, but it is not recommended because it makes everybody work more than necessary.

The range of a score is from **0 to 20**. In most cases, the score is an integer number, but for some learning units, it can accept up to 2 decimal numbers. There are some special cases where a score does not apply:

1. **unjustified absence:** the student did not show up to the exam and the tutor registered his(er) absence.
2. **justified absence:** the student was absent to the exams, but presented a medical certificate or any other legal document to justify his(er) absence. This option is only available for the program manager, who should be contacted by the student directly in the faculty office.
3. **cheating:** the tutor, who was conducting the exam, is convinced that the student was cheating during the exam. The tutor reports the student's fault selecting this option.
4. **missing score:** For innumerable other causes, the tutor doesn't have the student's score to encode and (s)he does not have the time to find before the deadline.

For any of these cases, the numeric score does not apply.

Double Encoding

To help minimizing encoding errors, OSIS also offers the possibility of re-encoding the scores to detect differences from initial attempts. It seems odd to re-encode everything again, but the probability of making the same mistake twice in the same place is considerably low, helping to expose the mistakes of the initial encoding.

The user interface of the double encoding is identical to the score encoding. It opens always empty, so the initial encoding does not influence the double encoding. As the user encodes the scores or the justifications the application instantly compares the encodings. When the scores or the justifications are different from the initial ones, then the impacted fields are highlighted in red, calling the user's attention to the differences.

The next step is to visually compare the two encodings. The comparison puts the encodings side-by-side and asks the user to chose of the the encodings when there are differences. The chosen ones are then saved for posterior submission.

The double encoding is optional but it certainly increases data reliability, dispite consuming twice the time of a simple encoding.