# Selection

The selection component handles the full selection process.

## Campaigns

Each selection process is represented by a selection campaign.

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Campaign.wmf

A campaign can contain several programs, each program corresponding to a *curriculum*. For example, in PNC we have a program SNA/WEP and another program DMO. The criteria of the selection for each program may be different.

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Functionalities_Campaign.wmf

Before to access to any data of a selection, a user must select the campaign on which it wants to access.

Only specific users can manage campaigns (create, rename, remove, edit).

## Partners contact

The first step of the selection is to contact partners (schools, NGO…), who will participate in the selection process (interested in PN scholarship for their students, providing facilities, contact point to reach students…).

The application will use the *Contact* component, so the selection team can manage a list of contacts.

The selection component will automatically create a contact type (*ContactEntityType*) “Selection process partner”, so we can easily identify a contact in the whole contact database.

In addition, the selection component will attach additional information to the contact, which are more specific to the selection, through *SelectionPartner*:

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Selection Contact.wmf

This way, the selection component provides the possibility to:

* Have a database of contacts, for each selection process.
* See the partners over the years (who was partner before…)
* Plan the selection, by managing groups of contacts among the partners (i.e. “to be contacted”)

## Staff roles for selection

All staff in PN are supposed to take part of the selection process. However, each staff may have different roles, depending on its seniority…

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Staff roles.wmf

## Selection zone

The selection is organized by “zone”, corresponding to a geographic zone, and most of the time it corresponds to an exam center.

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Selection Zone.wmf

Information sessions, exam centers, and interview centers are attached to a zone. This is used by selection team to organize the process, and have statistics.

## Information sessions

For information sessions, the main activity is to plan the sessions and assign staff to the sessions. Then, once a session is done, to enter the number of attendees.

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Information Session.wmf

An Information Session is attached to a campaign and a zone. When a schedule is available, a *CalendarEvent* is created. When the exact place is known, it is attached to the *SelectionPartner*. If not yet known the free text can be used.

Staff are assigned through *InformationSessionStaff*. When assigned, a staff is automatically invited to the *CalendarEvent*.

While planning, the expected number of attendees can be entered. Once the information session is done, the observed number can be entered.

## Plan exam centers

Most of the time, exam centers will be places where information sessions were done. Usually, when two places where information sessions are conducted are close, only one will be the exam center. The application will stay open and offers to create new places if needed.

As for the information sessions, an exam center is help on a partner place.

At this step, we do not define the schedules, but just which places will be exam centers.

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Exam Center.wmf

## Application forms

The application forms are imported from Excel files, through the data import component.

As the tables of applicants may become very big after few selection processes, it may decrease performance. And as we want to keep information, in a consistent way, for each selection process new tables will be created, using the “sub-model” feature described in the database chapter.

Also, each project may need to add or remove some specific information easily. Having a dedicated table for those specific information per selection process will make it easier.

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Application Form.wmf

At this step, the applicants are already assign to a specific exam center (even we keep the possibility to change this assignment at any time).

## Organize written exams

The goal of this step is to schedule when each exam will occur, to specify the rooms of the exam center, and assign applicants to rooms/schedule as well as supervisors.

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Exams.wmf

Once applicants are assigned to a room, it will allow to print the list of applicants for each room (to be displayed in front of the room, and allow the supervisors to check the attendance, as well as to give to the applicants their “code number” which is used with automatic correction to identify an applicant).

Note that a supervisor is only a *People*, to stay open enough: indeed on some projects, it must be a staff, while on another it may be also an alumni, or a partner… Staff will be proposed in the screens, but it will stay flexible to be able to assign someone else.

## Specify exam subjects and structure

D:\_PN\student_soft\repositories\lecousin\doc\system\diagrams\Components_Selection_Model_Exam structure.wmf

The difference between *ExamTopic* and *ExamSubject* is made to handle the case like the “Math and Logic” exam: we want it to be in the same subject, but we want to be able to differentiate the grade of Math and the grade of Logic.

In this case, we have one *ExamSubject* which is “Math and Logic”, and we have 2 *ExamTopic*: “Math” and “Logic”. *ExamSubjectPart* allows to identify different set of questions in a subject. If the subject is split into different topics, it will specify on which topic is the set of questions. But it may be useful also even a subject has only one topic, to know where an applicant is weak or strength (i.e. for “English” we may have parts “Grammar”, “Vocabulary”, “Reading Comprehension”).

The *max\_mark* for each topic and each subject is only a consolidation, to avoid calculating the sum each time we need, for performance reason.

## Enter exam attendance and results

After a written exam has been done, we can enter the attendance for each students, as well as if they cheated.



The results will be imported from Excel files (from scanning tool, or clicker tool, or manual…).



If available, the mark for each question is entered in *ApplicantExamResult\_Questions\_[SC]*. If not, the *ApplicantExamPartEsult\_[SC]* is directly entered.

The mark at each level is consolidated: from all questions mark, the mark is consolidated at the “exam part” level. From each part, the mark is consolidated at “subject” and “topic” level. Those consolidations are for performance only, to avoid retrieving a lot of data with calculation but to be able to access a mark at any level quickly.

## Enter eligibility rules for each program



The user can define *ExamRule* with a minimum mark for the addition of a list of topics through *ExamRuleTopic.* Example: Math + Logic + English = 65

Then, the user can define paths to reach the eligibility, using *ExamEligibilityRule*.

Example:

**Rule 3**

Math + Logic + English = 60

**Rule 2**

English = 20

**Rule 1**

Math + Logic = 30

**ELIGIBLE**

**Rule 4**

Math + Logic + Speed = 50

In this example we have 4 *ExamRule*. They are linked through *ExamEligibilityRule*.

The result is an applicant is eligible IF Rule1 AND Rule2 AND (Rule3 OR Rule4).

While the user is defining the eligibility rules, statistics about how much applicant comply to a rule, or all the rules are displayed. Also statistics like minimum grades, average grades… are displayed to help the user define the rules.

When the rules are saved, the eligibility is applied, and all passers can be easily retrieved through *ExamPasser*:



This information does not add any new information, but will ease to know who is eligible or not. After this step, normally only applicants in this list will be considered.

## Organize interviews

The organization of the interviews is quite similar to the organization of exams. For each exam center, we can see the passers, decide if we make the exam center as interview center, create new interview centers, assign passers to interview centers.

Once passers are assign to a center, we can schedule interviews and assign staff.



Note that an interviewer is a people, but the application should only propose staff, as an interview should not be conducted by someone else… but we may have a flexible option.

The way an interview is graded may be quite different from a project to another. To make it flexible, the structure of the grade is specified through the following table:



This way, we may have only few categories, like in PNC and PNV, but we can also have “parent category” containing sub-questions.

## Enter interview results

After interviews, the attendance is filled in *ApplicantInterview* while the grade for each defined category is entered in *ApplicantInterviewResult*:



TODO: additional info from interview, like choice of program…

## Enter eligibility rules for each program

## Organize social investigations

## Enter results

## Final choice of students and waiting list

## Applicants become students

## Changes: pick from waiting list, update list of students in the new batch

TODO:

When a campaign is created, a calendar is also created and associated with the campaign.

## Title 2

Text

### Title 3

Text

#### Title 4

Text

##### Title 5

Text

* Bullet 1
  + Bullet 2
    - Bullet 3