Eplouribousse

Collaborative webapp for managing serials deduplication in libraries

Whom this guide is for and what does it contain?

This guide is for library users concerned with the duplication of serials. It includes a general presentation of the method, the principles underlying its implementation and finally the app features and how to use them.

What does not this guide contain and where to find the corresponding information?

This guide does not include information about application deployment and contextual settings. For this, see:

https://seafile.unistra.fr/e/a998b238a22b4c13baf5/

1. General purpose

Without waiting to face with the saturation of storage spaces, the librarians have to worry about the good management of these. The weeding of collections is part of this good management, in the first place the weeding of duplicates. In this respect, librarians know that weeding journals is much more "effective" than weeding other types of printed material.

The *eplouribousse* app helps librarians to deal with serials deduplication in a library set.

eplouribousse is the Froggies form for the Latin E PLVRIBVS VNVM whose literal translation would be "Of many, one". *Eplouribousse* helps you to achieve making a good collection, i.e. as complete and bounded as possible, from scattered elements available in a set of libraries.

Note that *eplouribousse* has been initially developed in a french academic library context, so you may meet some terms related to this context. It should not be a fatal issue.

2. Principles (library policy)

We start from the following library policy hypothesis: **Participating libraries reconstruct and maintain a single collection, called the resulting collection or simply resultant, based on the largest collection held by one of the libraries; i.e. the mother collection. The resultant will be as complete and as bounded as possible (but without comparison of**

the quality of the bindings between them (Binding means normally complete volumes, and most of time, the mother collection is generally the best bounded one). However, it may happen that volumes are poorly bounded and that unbounded issues covering the equivalent period may advantageously replace them. *eplouribousse* take these cases into account.

Eplouribousse does not spare us the examination of collections; on the other hand, it allows us to prepare the resulting collection from this timely examination in the libraries involved.

We recognize two consecutive actions: Taking rank and instructing. Library that pretend to hold the mother collection, i.e. the largest and best one, must take 1st rank. At this stage, it is also possible to declare not to involve the collection. The grounds for non-participation depend on the situations; let's mention here as an example the legal deposit (administrators may be settled these reasons at any time[1])

We have just given the definition of what is a candidate to deduplication: Any publication for which at least two libraries hold some elements.

By publication, we mean a serial identified by a unique identifier (Serial Identifier) that does not always correspond, and not even generally, to the collection as of the librarians mean it. Doing the way *eplouribousse* does means assuming partitioning risk. Doing that is not a choice among other possibilities. It is the only possibility, at least the only one viable at medium or large scale (You shall convince yourself considering that a collection as of the librarians mean it can not to be the same for all libraries)

3. Legal aspects (data reuse)

Data must be legally enforceable. Data from the original *eplouribousse* application is recovered from the French academic library catalog Sudoc; unless otherwise stated, they are freely reusable (Etalab License)

For any other instance of *eplouribousse*, it will be necessary to ensure that the re-use of the dataset is legal.

4. Overview of Features

In the order of processing, the features are as follows:

- Positioning

- Arbitration

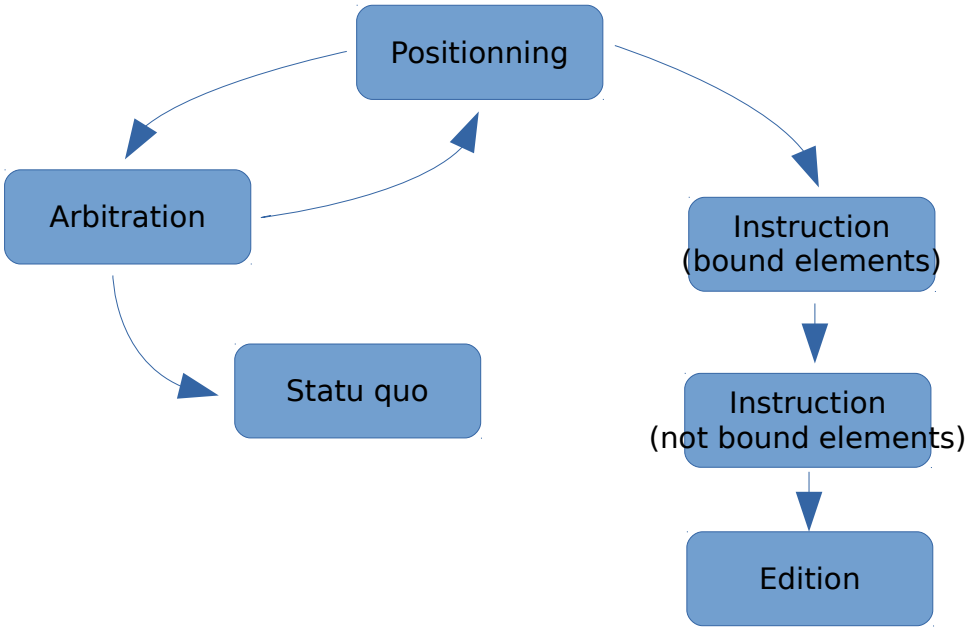
- Instruction

1 Database needs to be updated.

- Edition

These features apply at the level of each publication; therefore, it is not necessary (even if it is possible) to wait for libraries to position themselves on all candidates to begin arbitration or instructions, or to wait until all publications have been fully instructed to begin editing the resultant of one of them. In other words, it is possible to treat each publication end-to-end, leaving all others open.

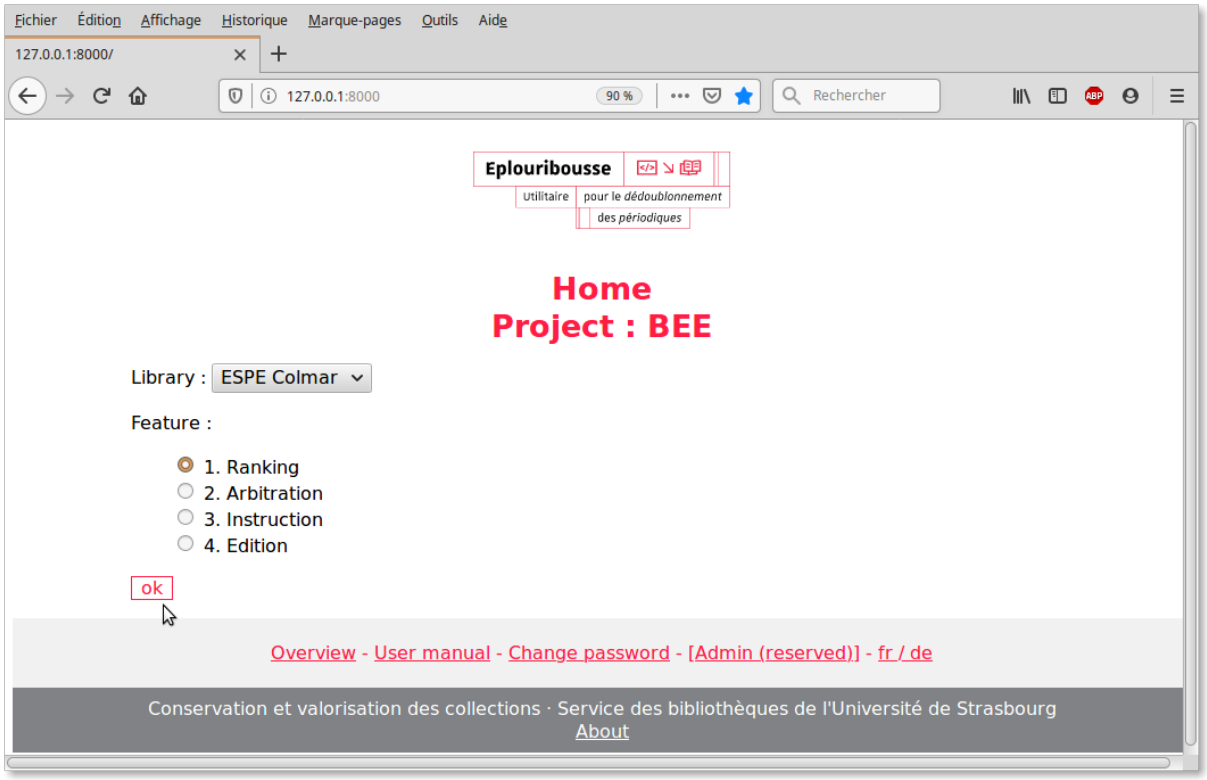
Overall scheme of end-to-end processing:



How to use the forms should not be a problem; we therefore only give some explanations or draw attention to the implications of this or that choice in filling forms.

5. Homepage:

You can access all the features of the application from this page: Positioning, arbitration, instruction, editing. Checks are made to prevent any out of delay actions (eg by clicking on a previously received link or using the back to previous page arrow of your browser)



(Note the project name)

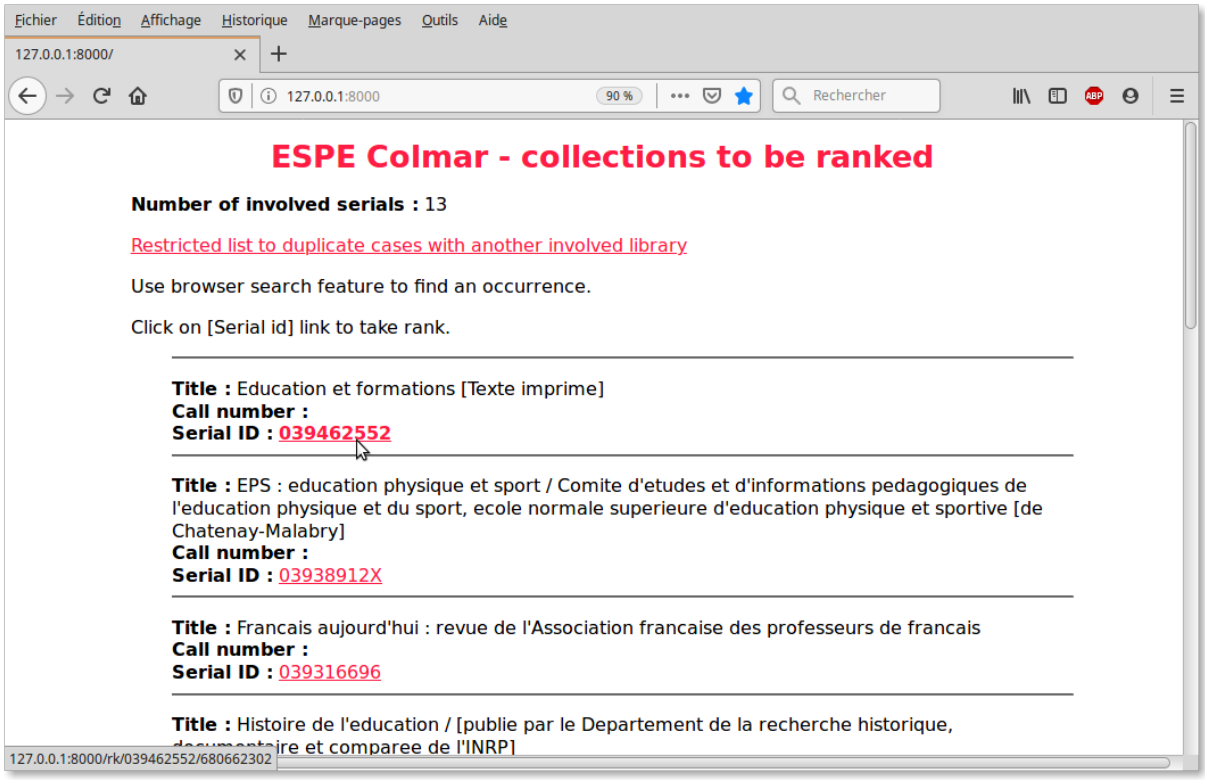
From this homepage, you may access to the dashboard, the user manual, password change, administration or access to change language page.

6. Positioning

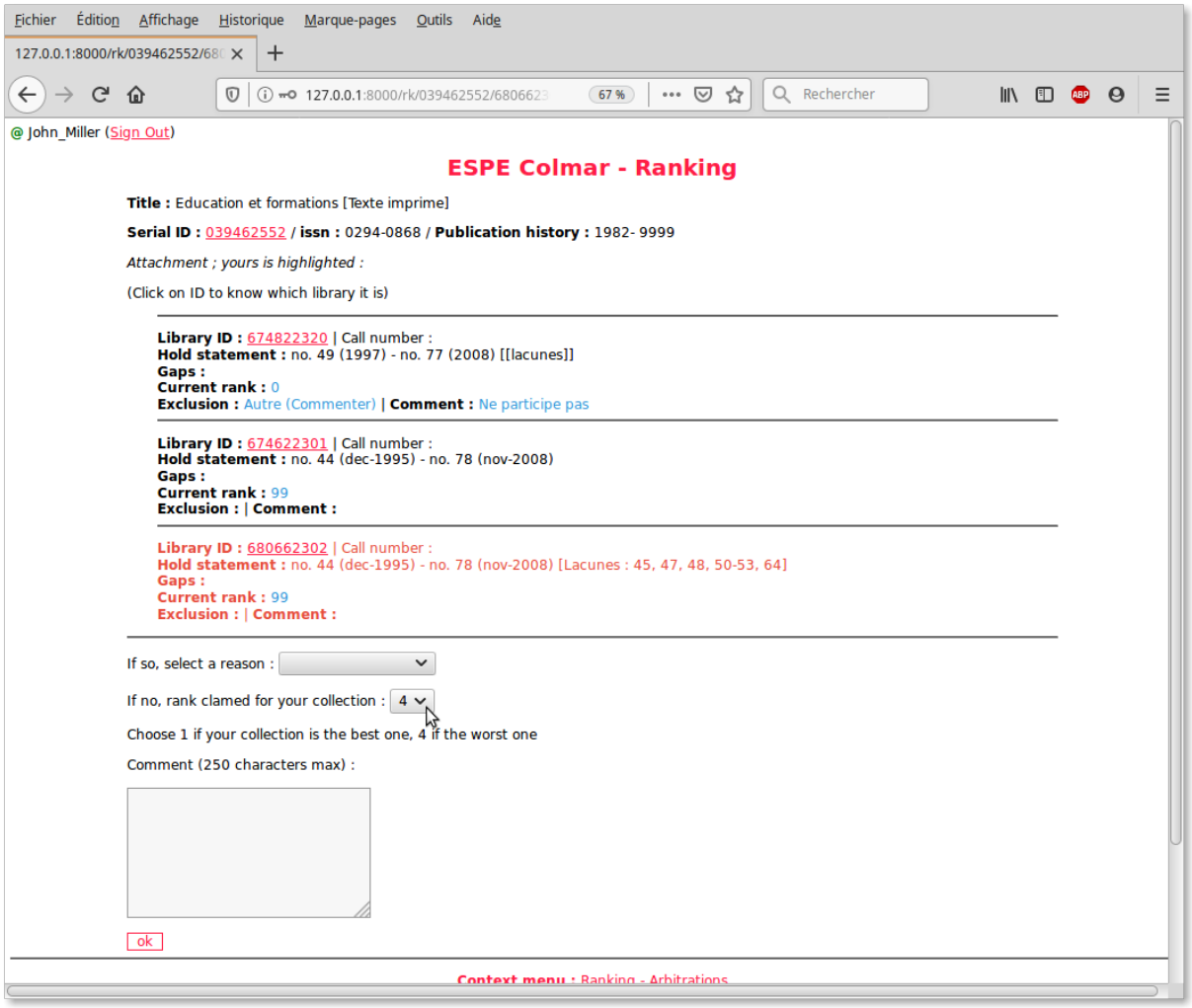
This step is necessary before any other. Following features become relevant only once all the libraries have positioned their collection.

Access: Select your library, select ‘Positioning’ and confirm.

Result: List of resources waiting for positioning (for your library):



Click on the link (Serial ID) to access the input form:



At initial state, the indicated positioning is 99; you can choose to exclude your collection for any of the reasons from the drop-down list[2]. Otherwise, you have to position your collection in a scale of 1 to 4: 1 if you have the most important collection and this collection is felt to be the one to which the elements of other collections will be added to improve it or 'enrich (mother collection) 4 if you have very few items for the collection. 2 and 3 for intermediate situations. The instruction order of the cards will follow the order of positioning.

~~At this level, the clickable link (Serial ID) refers to the Sudoc record which should be used to check the exact numbers of the beginning and end of publication as well as the filiations.~~ (french instance)

It is possible to leave a comment in one or the other of the aforementioned cases (note that you can widen the window of entry of the comment by activating the corner in bottom right)

NB: The connection is necessary to indicate the rank of a collection; a connection indicator appears at the top left (with the possibility of disconnecting)

2 Administrator may modify this list (Database update is necessary)

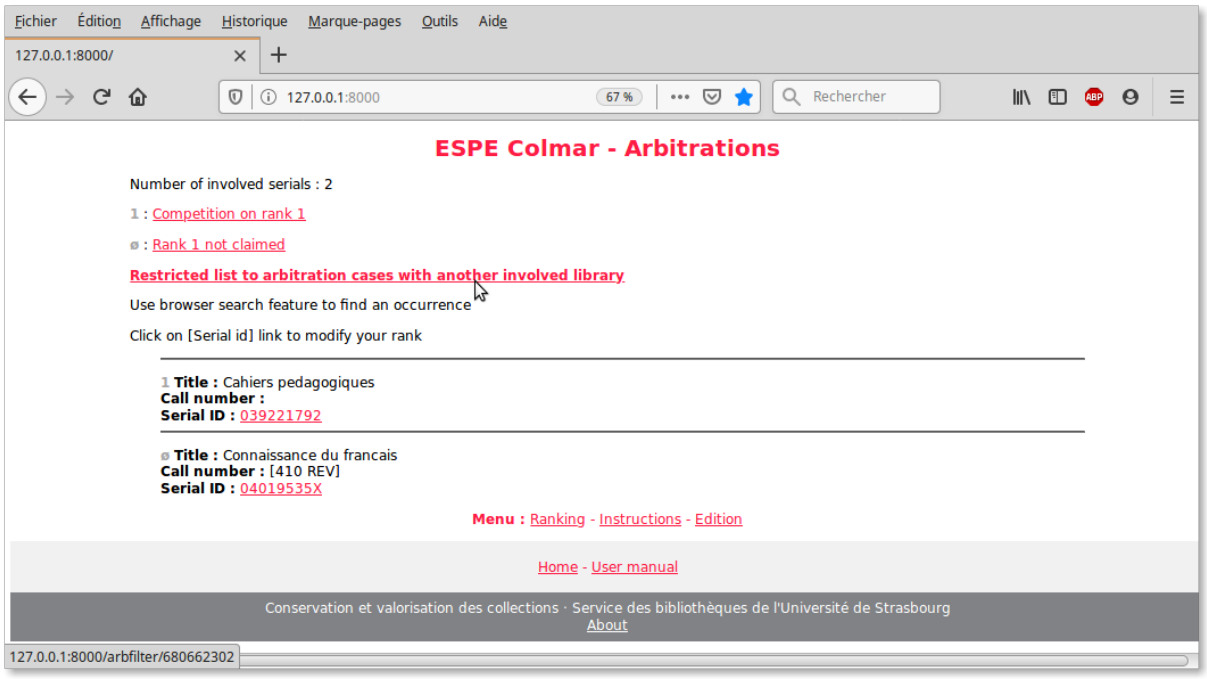
The connection is required for all the other crucial actions: Instruction, cancellation of an instruction line, statement of end of instruction.

7. Arbitration

Arbitration is necessary in two cases: Two libraries claim the first position or all libraries have taken a position, but none has claimed the first one.

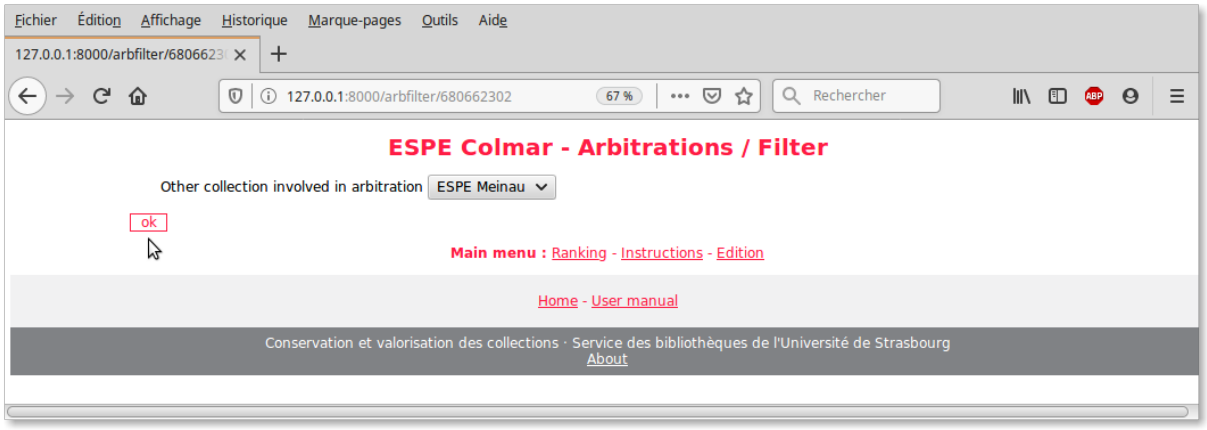
Access: Select your library, select ‘Arbitration’ and confirm.

Result: List of resources in any of the arbitration situations listed above:



Expected action: Revision of the positioning. If necessary, click on the link (Serial ID) to call up your positioning sheet and follow the procedure already described.

Note that the arbitrations cases are shown; you can limit to the restricted lists and even filter to get a list of arbitration cases with another involved library:

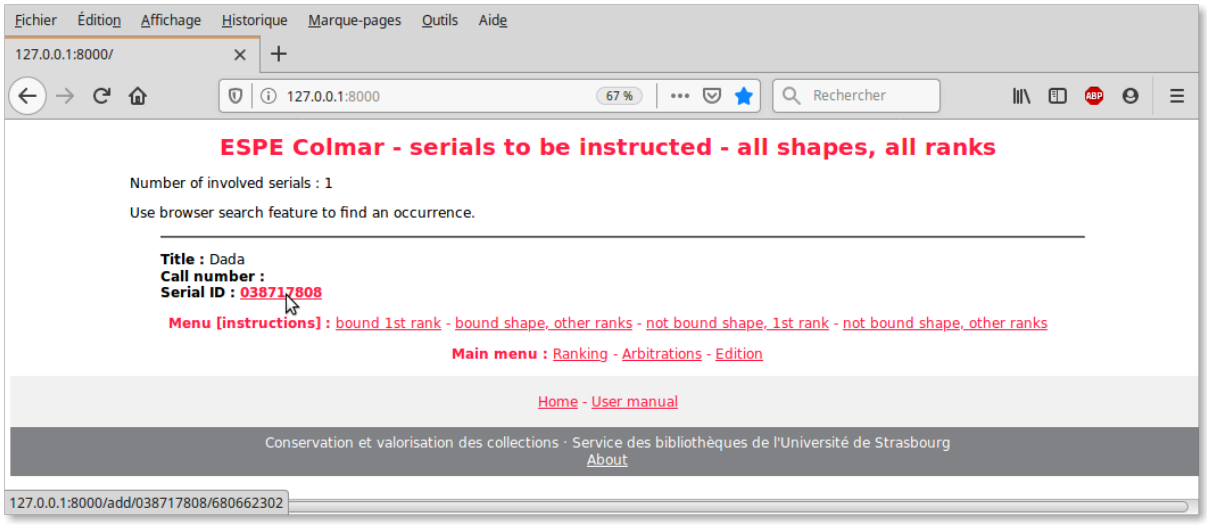


8. Instruction

The instruction of a resultant becomes possible as soon as all the attached libraries are positioned for the considered resource.

Access: Select your library, select ‘Instruction’ and confirm.

Result: A list of resources for which you can start the resultant statement:

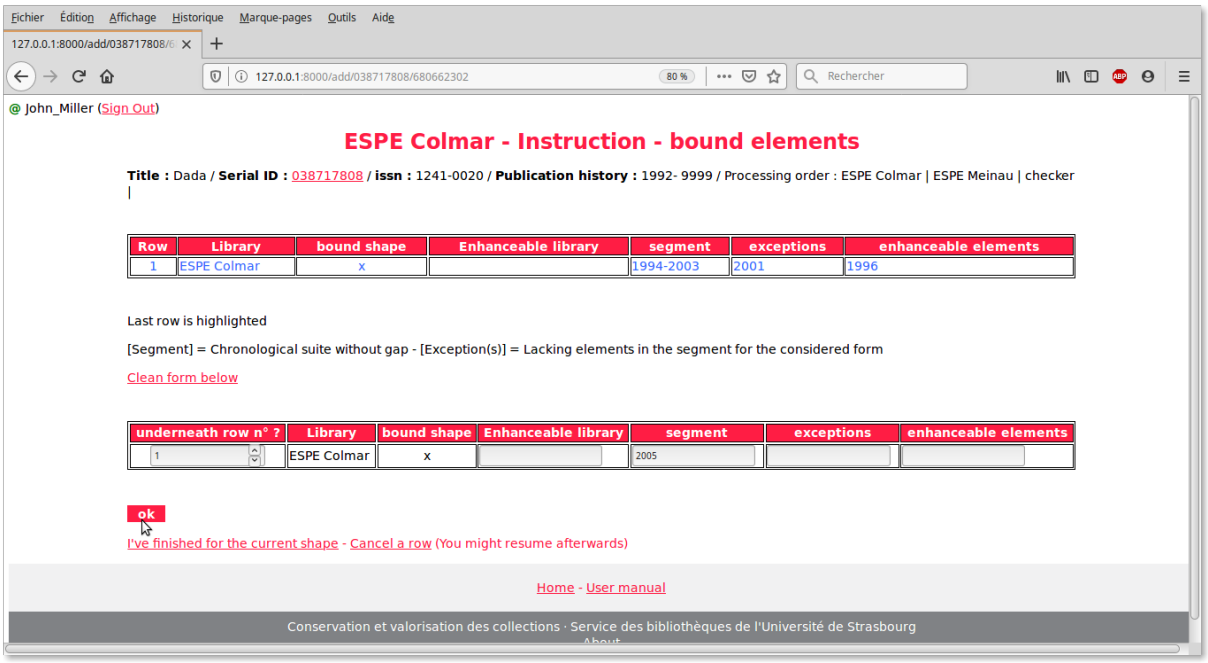


This method is the only way to start the instructions. Once a form has started to be instructed in this way, another channel is added to this possibility: The link automatically received by email when the previous library declared to have completed its instructions for the considered shape (see below). This solution makes it possible to subordinate the start of the instructions to the preliminary positioning of all the candidate collections.

How to instruct the resultant: Start by pointing out the bounded elements of your collection. Put it in the appropriate boxes : continuous segments or single elements in ‘segment’ box; enhanceable element is an element which can be replaced with a better one.

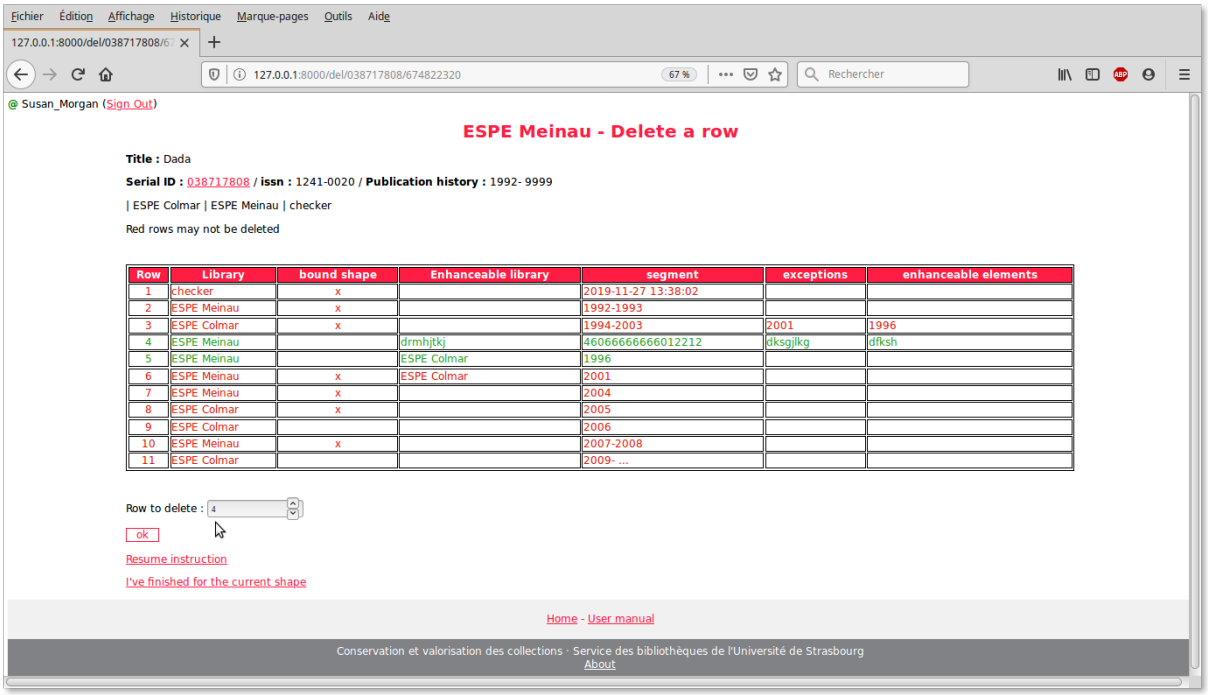
Indicate the line below which you want to insert the new line (‘segment’ logical order: chronological or by number)

The last instructed line is integrated and highlighted:

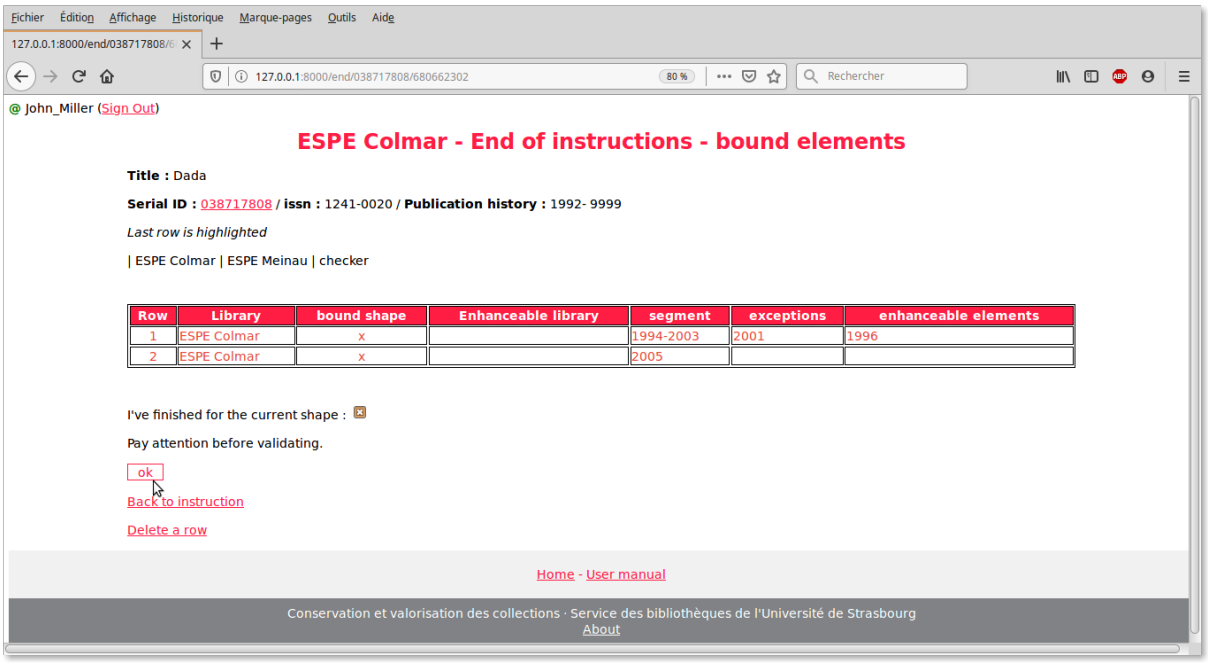


In case you have no contribution to bear for the resultant, you can directly indicate that you have finished, tick and validate, except other method agreed in your project (empty line for example).

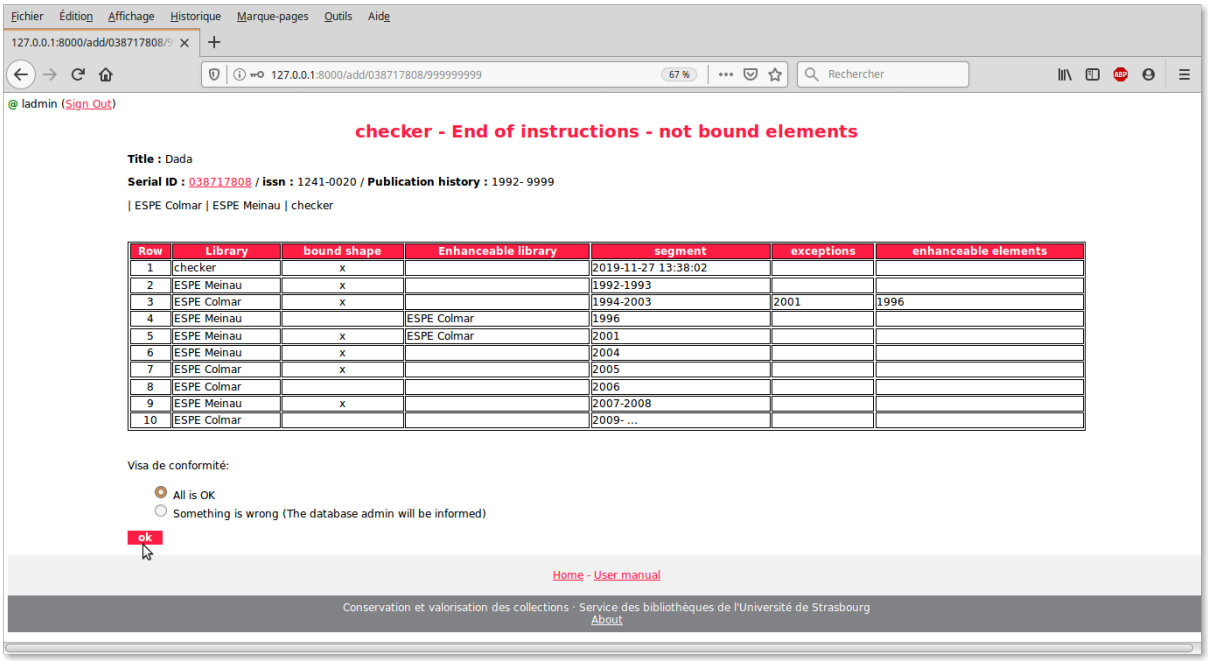
In case of error, you must delete the incorrect line clicking on the appropriate link, then return to the input form as you're invited to after deletion. It is possible to delete a row only for the account of its own collection and for the state being instructed (bound or not-bound) The lines that can be deleted are distinguished from the lines that can not be deleted. In this case, an alert message indicates that the line can not be deleted:



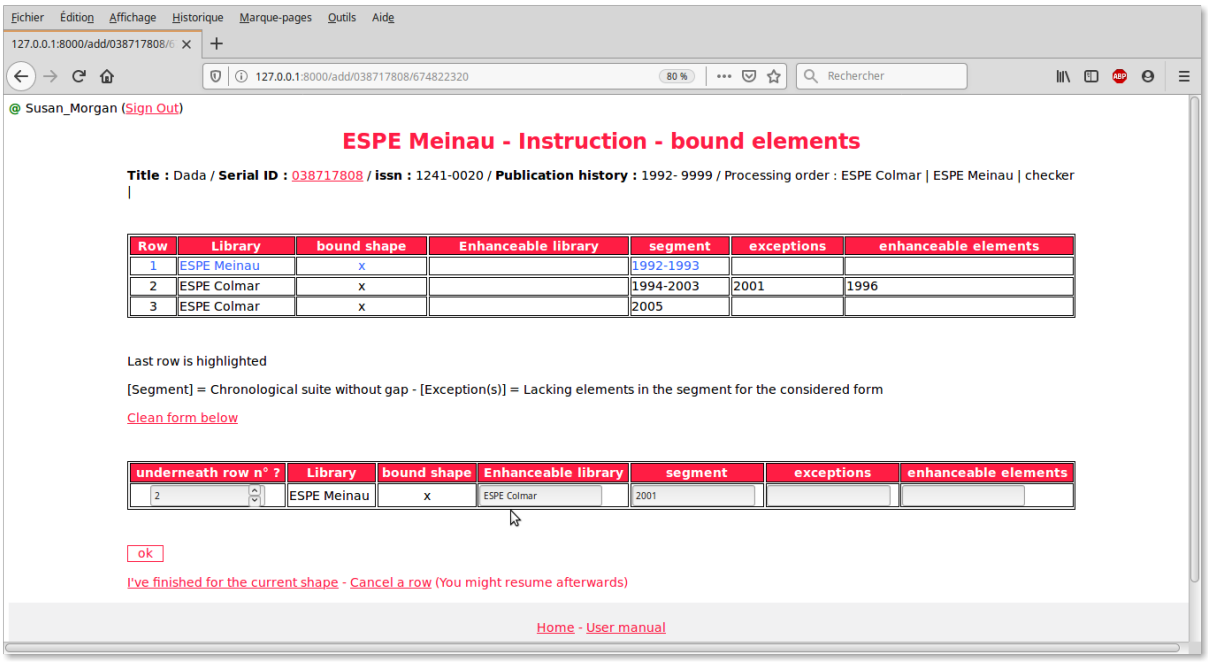
Once all the related elements have been entered, indicate that you have finished by clicking on the appropriate link, then confirm by checking and validating after taking good care to verify the accuracy of the instructions:



The next library contact will automatically receive a message warning him that his turn has come to instruct. The instruction is done in two cycles, the first for the bound elements, the second for the unbound elements. At the end of each cycle, the administrator checks that there is no inconsistency: If there is none, he validates. If the card contains inconsistencies, he informs the database administrator (checking the appropriate radio button):



When a library completes something that was previously carried by another library as an exception or an enhanceable element, and only in those cases, the name of that other library must be entered in the input box provided for this purpose. This ensures an optimal edition of the results when the time comes (see edition):



The order of improvement of the elements contributing to the resulting collection is as follows:

Bound in good condition > Not bound in good condition > Not bound > Bound in poor condition > (Nothing)

- NB 1: "Exception" means exception for the considered form (bound or unbound) This is not necessarily a real gap in the bound phase of instruction.

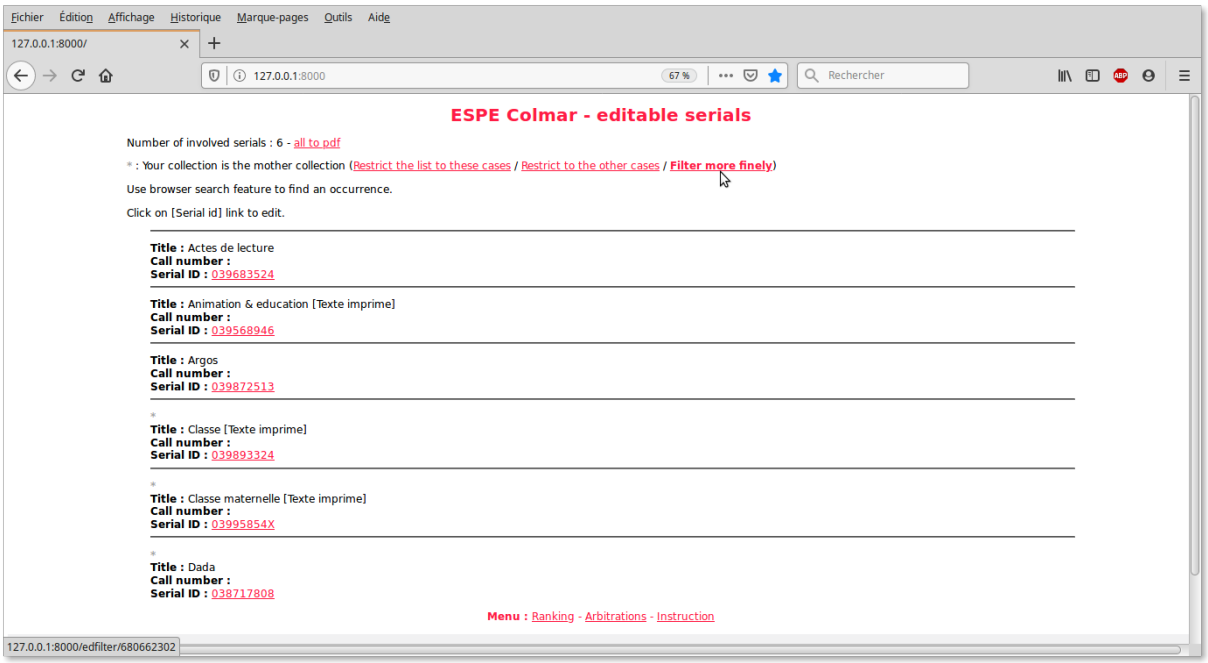
- N.B. 2: By ‘enhanceable’, we mean either degraded elements (for example, when the binding compromises the integrity of the contents), or elements that we know that another participating library has elements in better condition. Normally, this latter case is rare, as the library claiming the mother collection usually has the best collection.

9. Edition

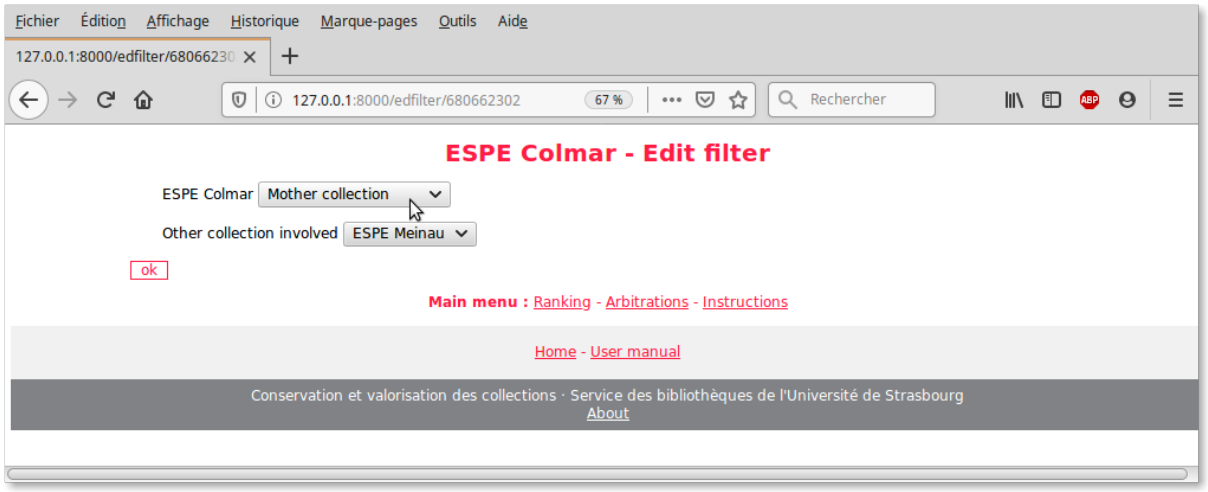
The edition is only possible for fully instructed resources.

Access: Select your library and ‘edit’ and confirm.

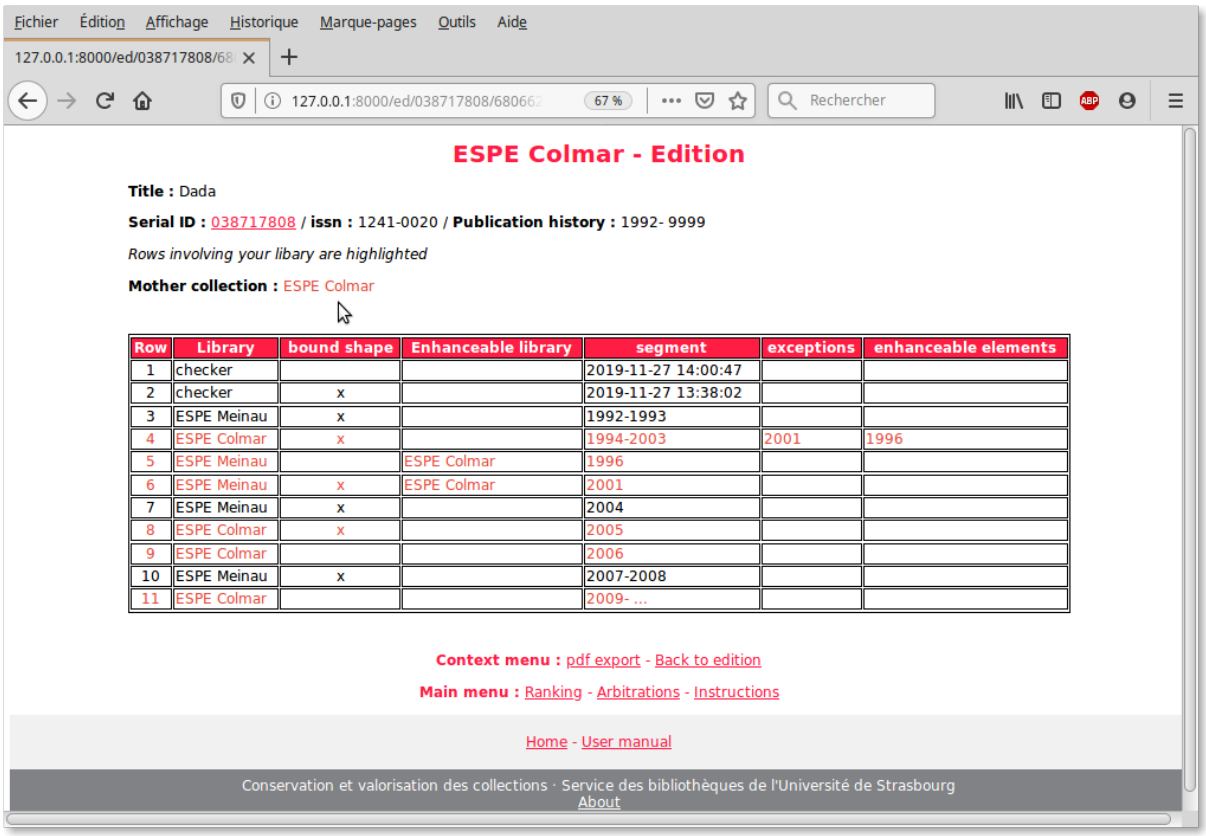
You shall first get the list of editable resources with distinction between the cases : Your collection is or is not the mother collection; you can limit list at each case:



You may also filter with another library collection:



Click on the resource you wish to edit. The lines involving your library are highlighted (hence the importance of correctly indicating the name of the library in the input window as indicated above during the instruction):



You can get neat pdf reports (set or single report) like this:

Project	BEE
Title	Dada
collection	ESPE Colmar -----> call number :

issn : 1241-0020	sid : 038717808	Publication history : 1992- 9999

1	ESPE Colmar	mother collection
2	ESPE Meinau	has taken part

#	library	bound ?	enhanced library	segment	exception	enhanceable
1	checker			2019-11-27 14:00:47		
2	checker	x		2019-11-27 13:38:02		
3	ESPE Meinau	x		1992-1993		
4	ESPE Colmar	x		1994-2003	2001	1996
5	ESPE Meinau		ESPE Colmar	1996		
6	ESPE Meinau	x	ESPE Colmar	2001		
7	ESPE Meinau	x		2004		
8	ESPE Colmar	x		2005		
9	ESPE Colmar			2006		
10	ESPE Meinau	x		2007-2008		
11	ESPE Colmar			2009- ...		

10. Search in lists

It may be useful to look for an occurrence in case of large lists (resources waiting for positioning, resources whose instruction can begin, resources whose result can be edited)

For this, the best way is to use the search function of your browser.

12. Role of the database administrator

The database administrator is alerted by an email in case of inconsistency in the instruction sheet.

After the usual precautions in such cases (server shutdown, backup of the database before modification) the administrator of the database will have to modify the incriminated recordings directly in the database according to the following indications:

For a considered resource (a sid) it will locate the concerned instruction records, identify what the anomaly is and make the necessary corrections by recursively following the processing (which corresponds to the inverse order of the instruction lines)

Once done, it will be possible to assign the new statuses to the attachment records (template: ItemRecord) and change them from status = 6 to status = 0, 1, 2, 3 or 4 according to the following considerations:

If there is no ‘Checker’ instruction yet, the possible statuses can only be 0, 1 or 2

If there is already a ‘Checker’ instruction line, the possible statuses can only be 2, 3 or 4

- 0: initial state (it is not yet the library's turn to instruct)

- 1: related elements to instruct

- 2: related elements learned

- 3: unrelated elements to instruct

- 4: unrelated elements learned

The report of an anomaly by the ‘Checker” to the administrator of the database has the effect of changing the state of all the records concerned (ItemRecord) This state goes to 6. The administrator can make the corrections to measure , or wait until there is a certain number to perform. It is even possible to process them only at the very end of the process, once all the other forms have been fully processed.

For information, the status 5 is assigned to all the attachment records once the ‘Checker’ has provided his compliance visa at the end of the two cycles of instruction.

13. Indicators

A dashboard is available from a link in the homepage.

Indicators are trivial, they may be more in the future, so we do not detail here.

14. Security

All actions involving data manipulation need authentication.

15. Authentication

An identifier and a password are assigned by the administrator of the site; the user can then change the password from the ad-hoc link on the home page.

Authentication is required automatically for any sensitive action (database data changes). A check is made on the email address (the user email and the email of the correspondent for the library must be the same) **If the control is negative, the user is redirected to the general homepage.**

Apart from this, authentication is not required.

16. Users and groups

There are 4 user groups:

- Exterior

- Main users (whose validator = "checker" is only a special case, recognized by its name in the code)

- Administrator of the base

- Site Administrator

Each of these groups has its own rights; in the order of increasing rights:

- External: Access to all pages where no action leads to a crucial change in the database; access to lists and dashboard is allowed without authentication.

- Main users: Can perform all the basic actions: Positioning, arbitration, addition of instructions, deletion of instructions, statement of end of instruction, editing and validation in the case of the validator ("checker") Any attempt to access to one of these features activates an intermediate authentication window.

- Database administrator: Can modify instructions and records in the database via the administration interface (access from the home page) Can modify the reasons for exclusions.

- Administrator of the site: All rights (among them giving a name to the project)

17. Languages

The application uses gettext for translation into other languages. French is the language of origin.

Two other languages are currently proposed on an experimental basis; English and German.

The extension to other languages is quite simple since it relies on the provision of standard .po extension files easily understandable by translators. These files are compiled by a simple command line (.mo machine files)

The change of language is accessible from a link on the homepage.