

FAILTE System Design

(A Methodology to identify the components of the FAILTE System)

Author	Santiago Chumbe
Related Task	3.2, 4.1, 4.3 and 4.4
Date	13 - November - 2000
Version	1.3
Summary	<p>This is a general description of a model, which is going to be used to design the FAILTE System. It is expected that this model will be employed as a graphical aid for the designers of the FAILTE System. Hopefully, this tool will make easier to understand and visualise things such as the user interfaces and the major components of the FAILTE System.</p> <p>The model uses both, the top-down vision, to get a whole picture of the work that needs to be done, and the bottom-up approach, to produce a more detailed description of each component of the FAILTE system (*). The purpose of this methodology is to assist the designers in the identification of tasks such as the system input/output requirements, the data transactions involved in its process and all the set of critical Web/Database operations for the FAILTE System.</p> <p>This document is intended to give the FAILTE Staff an opportunity to comment and improve the design of the FAILTE System. The model is not a final scheme. Moreover, it is open to suggestions coming from any source. Any feedback to improve the design is very welcome.</p> <hr/> <p>(*) Only the top-down approach is discussed in this report</p>

1. FAILTE System Main Processes

“Main purpose of the FAILTE Project: to create a database which is searchable online via a WWW interface”

The FAILTE System will be a software package composed by the “*Input*” and the “*Main*” databases (the “*FAILTE Databases Scheme*” report presents a full description of these databases), and a set of programmes mainly installed on a UNIX platform. Its aim is to fulfil the main purpose of the FAILTE Project.

We can identify four main processes in the FAILTE System (Figure 1.1):

- **The Data Entry Process**, to allow the input of new resources into the “*Input*” database. The cataloguer will create the new entries for the database. However, the end-user will also have the chance to suggest new entries.
- **The Data Validation Process**, to review and validate the contents of the new entry prior to its inclusion in the “*Main*” database.
- **The Search/Retrieval Process**, to allow Internet end-users to search and retrieve information from the “*Main*” database.
- **The Data Management Process**, to perform administrative functions on the “*Main*” database, such as: input of new records, index of tables, change of record contents, removal of records, performance optimisation, etc.

The Search/Retrieval Process is the most open one. Anybody can make use of this read-only process to search in the “*Main*” database. On the other side, the Data Management Process is the most restrictive one. Only authorised staff can make use of this process. This approach plays a key role in the prevention of risks, which could harm or damage the FAILTE “*Main*” database.

Following our top-down strategy, we are going to decompose each of these four processes into a set of middle-level or intermediate processes.

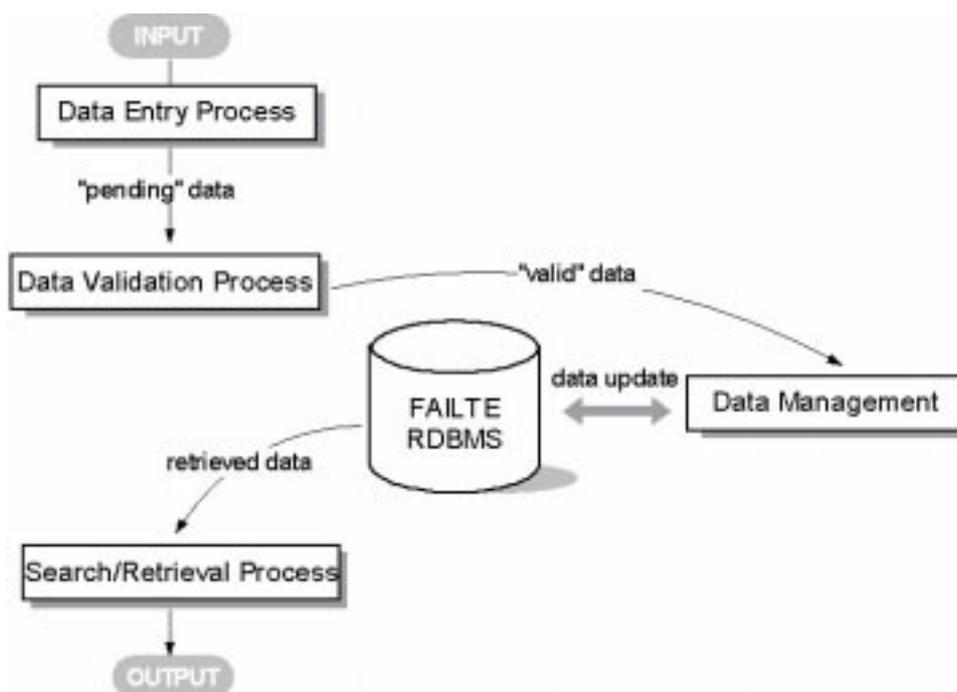


Figure 1.1. The FAILTE System and its Main Processes

2. Decomposition of the main processes into sub-processes

2.1. Data Entry Process

This process will be remotely accessible. The cataloguer will have control on all the steps involved in this process. She can create new records, analyse their contents, change their status and decide to add or remove records from the “*Input*” database of the FAILTE RDBMS. Also the end-user can contribute with new records. He will be encouraged to suggest new learning and teaching resources in Engineering for the FAILTE service, by filling an online form, in a similar way to the <http://www.eevl.ac.uk/suggest.html> page.

ADMIT Task⁽¹⁾

- The cataloguer may access the “*Input*” database via a MyODBC driver from the MS Access platform. She also can make use of a web-based user-friendly interface. Then, she can create new entries as “*admitted*” records, assess the user suggestions (if a “*suggested*” record is suitable for FAILTE, then its status will be changed to “*admitted*” record. If not, it will be labelled as an “*unsuitable*” record.) She also can complete and modify the fields of previous “*admitted*” records.

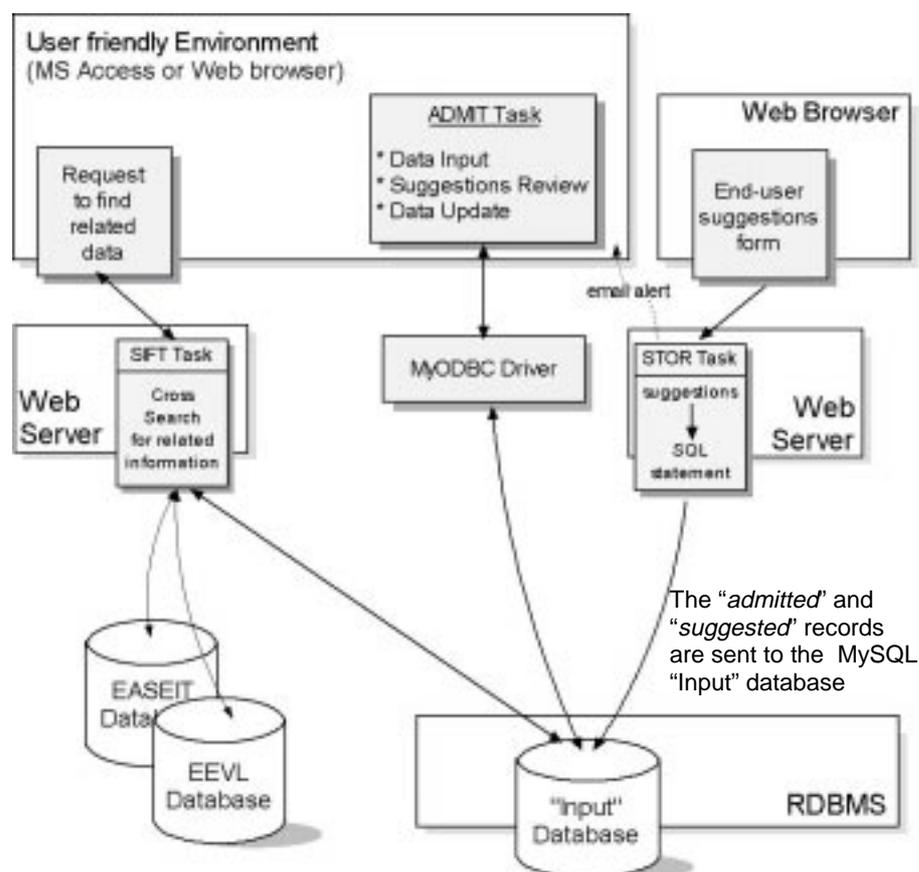


Figure 2.1. The Data-Entry Process

The cataloguer will use a Web browser to find out if the “*admitted*” records have related information already stored in the EASEIT, EEVL or FAILTE main databases. An “*admitted*” record will become “*pending*” if it doesn’t exist in the EASEIT and EEVL catalogues as well as in the FAILTE “*Main*” database. Otherwise, the cataloguer can decide to remove it or to keep it as a “*pending*” record if she thinks the resource has a learning and teaching value, which has not been properly described in any of the main databases. It is expected that the LTSN-Eng database will be included into the FAILTE RDBMS.

Record status in the “Input” database

- Suggested:* It is a new record suggested by the end-user. It has only its “*obligatory*” fields filled. The record is waiting to be assessed by the cataloguer. If the assessment is satisfactory, the record will become an “*admitted*” record.
- Admitted:* It is a new record, which doesn’t have all its “*required*” fields properly filled. Its existence or not in one of the main databases has not yet been determined.
- Pending:* It is an admitted record with all its core fields properly filled. It doesn’t exist in any of the main databases. Therefore, it can be included in the “*Main*” database. It is ready to be validated.
- Valid:* The record contents have been validated by an authorised member of the FAILTE staff to find for spelling mistakes, empty “*required*” fields, duplicate information, etc. It is ready to be checked by the automatic inspection.
- Complete:* It is equivalent to “to-be-included”. The record has successfully qualified the automatic inspection and it is ready to be included into the “*Main*” database.
- Included:* The record has been successfully included into the FAILTE “*Main*” database. The record exists in both the “*Input*” database and the “*Main*” database.
- Incomplete:* The automatic inspection has found that some of the “*required*” or “*obligatory*” fields of a “*valid*” record have not been filled properly.
- Unsuitable:* It is a record not suitable for the “*Main*” database. It is kept in the “*Input*” database for future references.
- Aged:* It was an “*included*” record, which for some reason has been removed from the “*Main*” database (i.e. the URL not longer exists, the resource version is not supported any more, etc.)

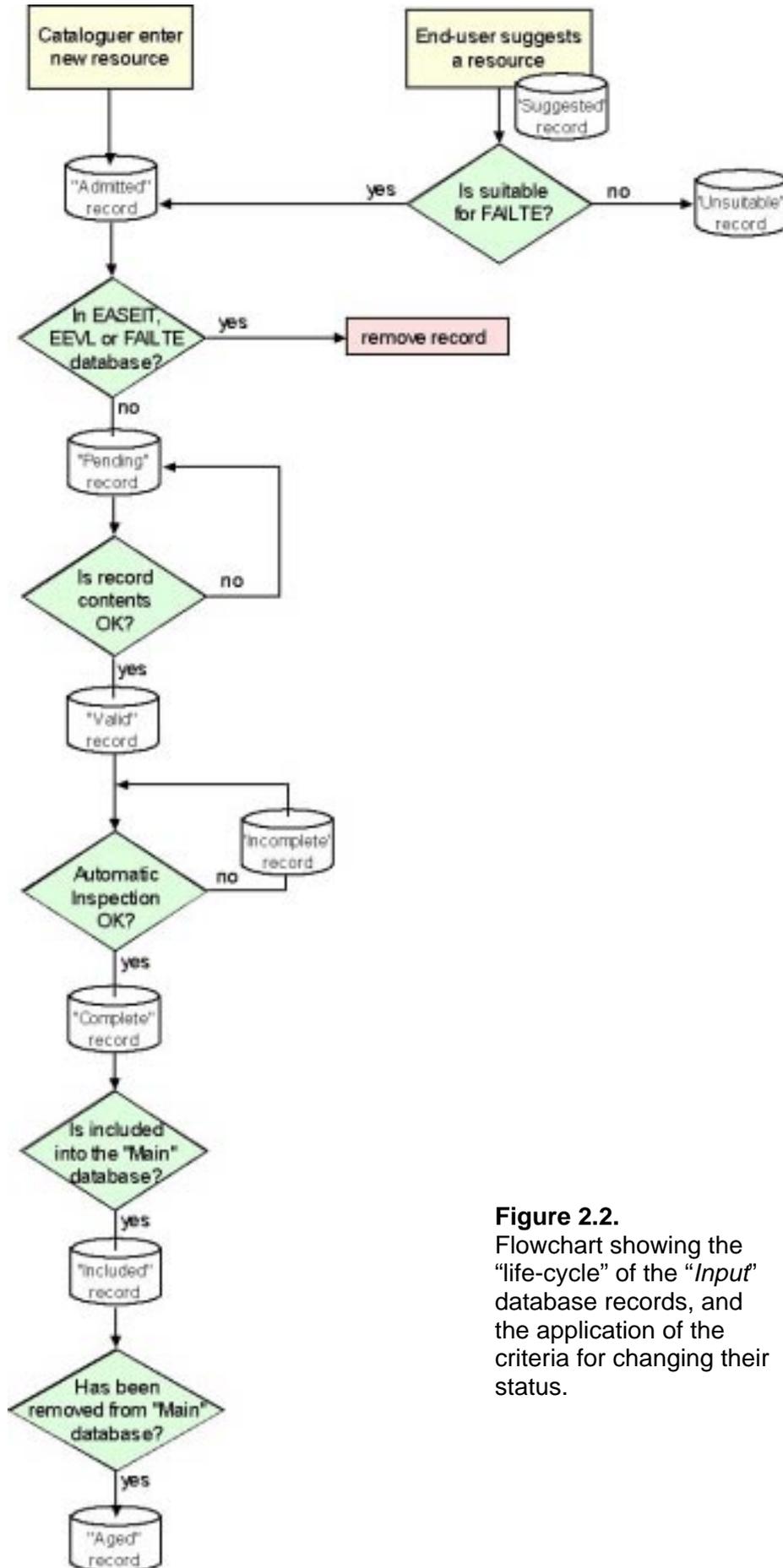


Figure 2.2. Flowchart showing the “life-cycle” of the “Input” database records, and the application of the criteria for changing their status.

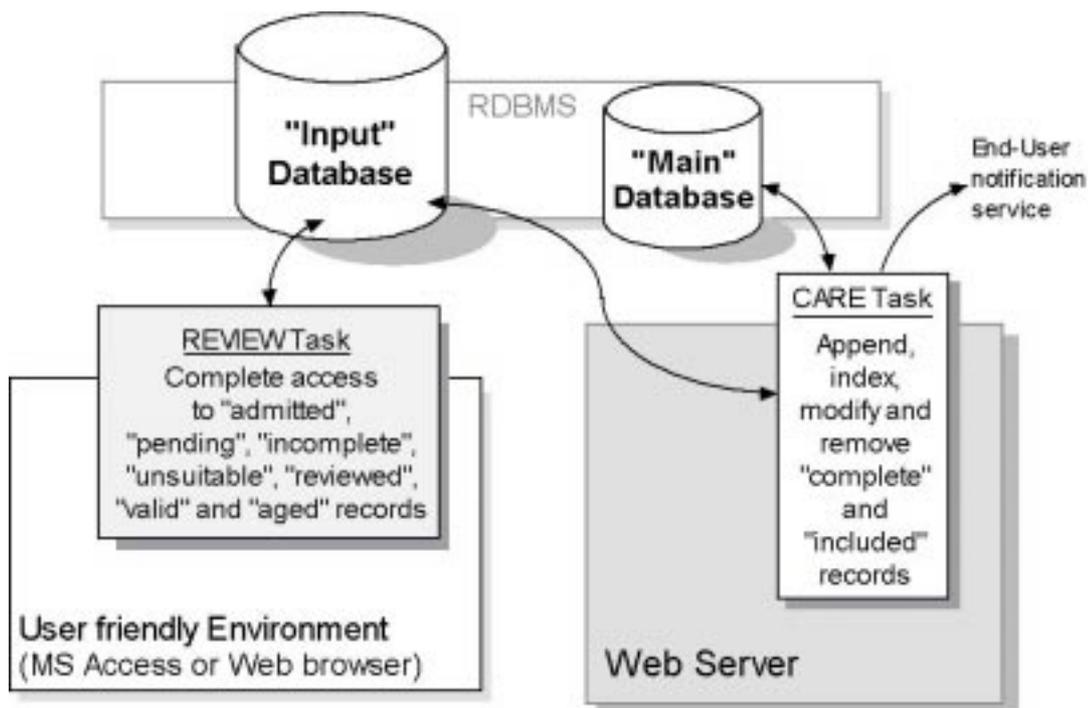


Figure 2.3. The Data Validation and Data Management Processes

2.2. Data Validation Process

The REVIEW Task will assist the validator in the validation of a new resource. REVIEW will be able to show the records of the "Input" database selected and arranged by date and status.

We envisage that the same software modules will be used for ADMIT and REVIEW. Typically, the validator will choose a "pending" record and she will investigate its suitability. She also can change or add information in the "pending" record fields. If the resource is considered suitable for inclusion in the "Main" database, the reviewer will change the status of the record from "pending" to "valid". If not, the record status should be changed from "pending" to "unsuitable." The validator should leave the record status as a "pending" if she considers that further review is necessary to be done. REVIEW will also allow the validator to access and modify "complete", "incomplete", "unsuitable", "valid" and "aged" records. However, she will not be able to access to the "included" records.

2.3. Data Management Process

This process will allow authorised FAILTE staff to access to the data stored in both the "Input" and the "Main" databases of the FAILTE RDBMS. At least two functions can be identified to perform the FAILTE Database Management:

Data Append function

The input of new records into the FAILTE “*Main*” database will be basically an automatic task activated periodically by the database administrator. This task will transfer an exact copy of all the “*complete*” records from the “*Input*” database to the “*Main*” database and then, it will change the status of these records from “*complete*” to “*included*” in the “*Input*” database.

The Data Append task will check the “well-formedness” of each “*valid*” record before changed its status to “*complete*”. If the record accomplishes the established “well-formed” condition and validity criteria for FAILTE records, the task will change the record status and append it to the FAILTE database. Otherwise, the record status will be changed from “*valid*” to “*incomplete*.” This sub-process may generate a log file with all the operations executed on the implicated records.

Database Update function

The CARE Task is the only one, which has a complete and direct control on the FAILTE “*Main*” database⁽²⁾. It will be the only means to index, modify and remove records from this database. Moreover, it can be used to modify the contents and the status of any record of the “*Input*” database. Also, it will display all the database contents, generate reports and provide tools to maintain the RDBMS. Eventually, this sub-process will be used to create backups and perform database recovery and optimisation tasks. The user of this sub-process must be aware of the risks involved when the CARE task or any other task, is executed against any of the FAILTE databases.

Periodically, the main tables of the FAILTE RDBMS will be re-generated. This regeneration will involve tasks such as: rebuilding tables; running an optimiser to maintain performance and to avoid tables fragmentation; indexing tables to speed-up the searching process, etc.

2.4. Search/Retrieval Process

This process will interact with the clients, who will be requiring information to the FAILTE System. The EEVL, LTSN-Eng and EASEIT-Eng services are the clients of FAILTE. The Internet end-user will access the FAILTE “*Main*” database via these clients, which will invoke directly to the FAST Task (see Fig. 2.4.) FAST is the key sub-process to keep an excellent performance for FAILTE⁽³⁾. It will make use of an efficient and appropriate algorithm for performing the search/retrieval process in the minimum time possible. The figure 2.4 shows the data flow for the Search/Retrieval process and its self explained steps.

The FAILTE System will provide three automatic e-mail services, which may be implemented in the Data Entry and Data Management process respectively:

- An alert e-mail to be sent to the cataloguer each time that an end-user suggests a new resource.
- A notification service, which will send an e-mail to the validator when the status of a record in the “*Input*” database changes from “*admitted*” to “*pending*”.
- A notification e-mail service for registered end-users, which will be automatically activated by the CARE task when a new resource is included in the “*Main*” database.

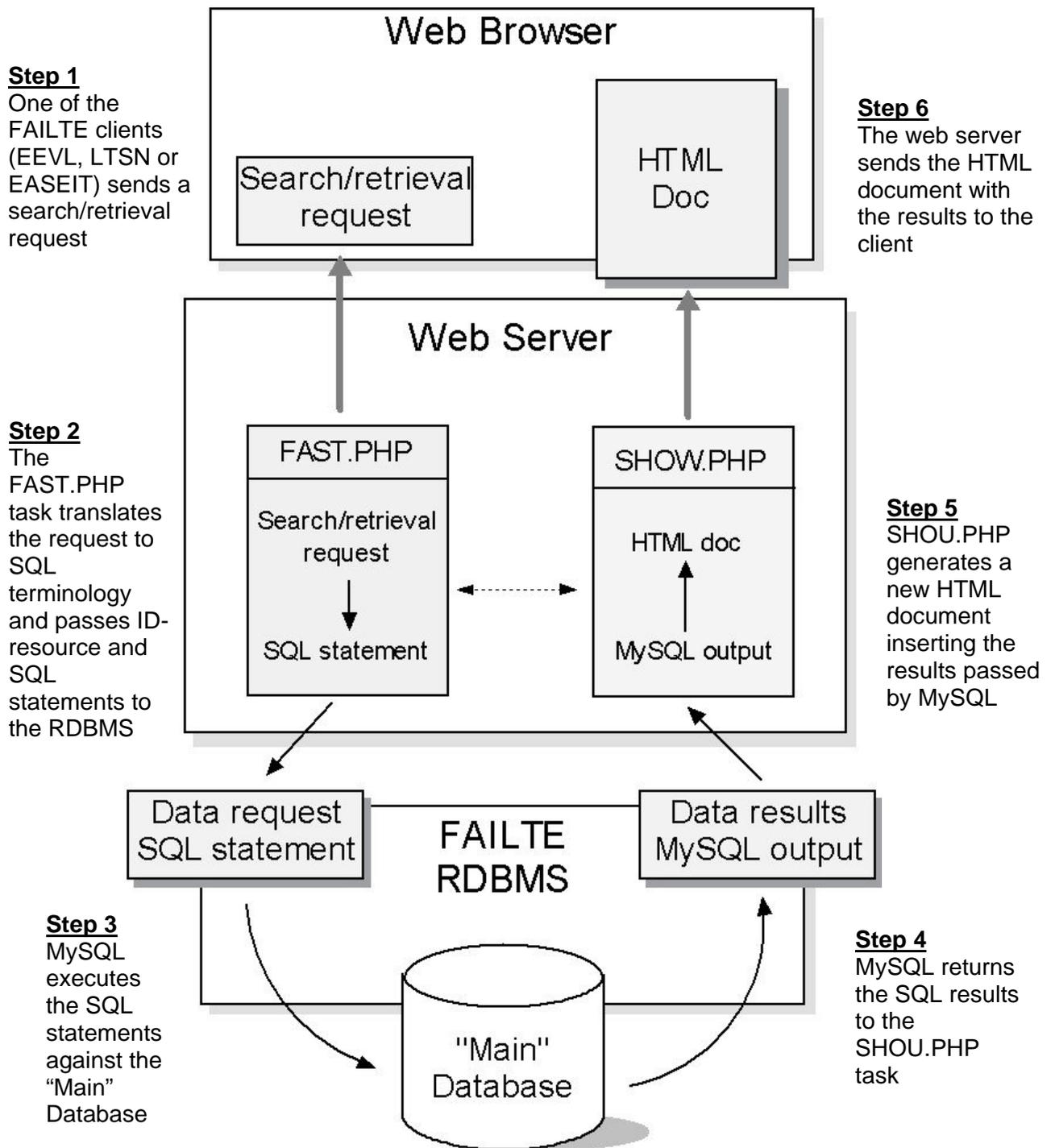


Figure 2.3. The Search/Retrieval Process

Notes:

- (1) The cataloguer and the validator can select between the MS Access software and a web-based implementation to remotely access the “*Input*” database. However, they will need to use a web browser to perform the SIFT task (see Fig. 2.1). This task will take advantage of the WWW interoperability to determine if an “*admitted*” resource exists or not in some of the main databases (EVL stores information on a custom built Java database only accessible from the WWW and the EASEIT-Eng database is based in a MS-Access/ASP implementation.)
 - Most of the tasks are likely to be written in PHP 4.0. In general, a PHP script can include one or more tasks.
 - SIFT stands for **S**earch **I**nformation **F**or FAILTE.
 - STOR : **S**uggestions **T**O be **R**eviewed
 - ADMIT stands for **A**dd, **D**iscover and **M**anage **I**npu**T**s.

- (2) The “*Main*” database is a subset of the “*Input*” database. It will store only “*included*” records. A full description of the FAILTE databases is being prepared in the “*FAILTE Database Scheme*” report.
 - CARE stands for **C**areful **A**lteration, **R**eplacement and **E**limination

- (3) - FAST: **F**ast **A**ccess & **S**earching **T**ool
 - SHOU: **S**how **H**ypertext **O**utput to the **U**ser