



Brand Repository

Functional Specifications

Release 1.0

1 Content

2	Document Information.....	4
2.1	Versions	4
3	Data	5
3.1	Definitions	5
3.2	Perimeter.....	5
3.3	Schema	5
3.4	Data Model.....	5
3.4.1	BRAND table	6
3.4.2	BRAND_TYPE table	6
3.4.3	BRAND_OWNER table	6
3.4.4	Visitor proposal table	6
4	General Functionalities.....	7
4.1	Users.....	7
4.1.1	Type of users	7
4.1.2	Authorizations	7
4.2	Log	7
5	Admin Functionalities.....	8
5.1	Authorizations	8
5.1.1	Groups	8
5.1.2	Users.....	8
5.2	Brand management.....	8
5.2.2	Validate a brand	8
5.2.3	Add a brand	9
5.2.4	Modify a brand	9
5.2.5	Delete a brand	9
5.3	Owner management.....	10
5.3.1	Listing of owner	10
5.3.2	Add an owner	10
5.3.3	Modify an owner	10
5.3.4	Delete an owner	11
5.4	Brand Type management	11
6	Public Functionalities.....	11
6.1	Home page	11

6.2	Menu	11
6.3	Form to add a brand.....	11
6.4	Brand List.....	12
6.5	Owner list	12

2 Document Information

2.1 Versions

Version	Author	Modifications
2013.11.13 #2	P. Plagnol	Creation of the document
2013.11.13 # 3	P. Plagnol	Modifications in blue
2013.11.14 # 1	P. Plagnol	Modifications in blue
2013.11.20 #1	P. Plagnol	Modifications in blue
2013.11.26 #1	P. Plagnol	Modifications in blue
2013.11.27 #1	P. Plagnol Nicolas Pieuchot	Modifications in blue

3 Data

3.1 Definitions

BSIN: The **B**rand **S**tandard Identification **N**umber is a 6-character alphanumeric unique (0 and O excluded) identifier assigned by OKFN. Each brand stored on this repository is given a unique BSIN.

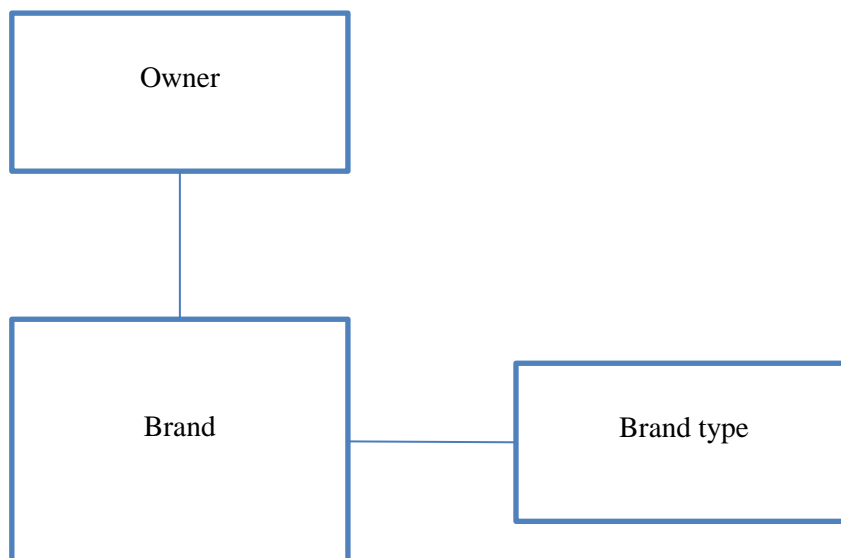
Owner: Legal owner of brand

Type of brand: Brand owner by manufacturer and retailers are of 2 different types. The repository contains groups of brands when the number of brand is too important and that the number of product is small (Wine, Cheese,...).

3.2 Perimeter

The repository excludes brands assigned to books and magazine.

3.3 Schema



3.4 Data Model

3.4.1 BRAND table

Field	PK/FK	Type	Null	Description
BSIN	PK	char(6)	N	BSIN code
BRAND_NM		varchar(255)	N	Name of the brand
OWNER_CD	FK	int	Y	Owner of the brand
BRAND_TYPE_CD	FK	int	N	Type of the brand
BRAND_LINK		varchar(255)	Y	URL of the brand
FLAG_DELETE		tinyint(1)	N	Deletion flag 0: not deleted, 1: deleted
LAST_MODIFIED		datetime	N	Last modification date
COMMENTS		varchar(255)	Y	

3.4.2 BRAND_TYPE table

Field	PK/FK	Type	Null	Description
BRAND_TYPE_CD	PK	int	N	Type code
BRAND_TYPE_NM		varchar(255)	N	1: Manufacturer-brand 2: Retailer Brand 3: Group of brands

3.4.3 BRAND_OWNER table

Field	PK/FK	Type	Null	Description
OWNER_CD	PK	int	N	Owner code
OWNER_NM		varchar(255)	N	Owner name
OWNER_LINK		varchar(255)	Y	URL of the owner website
OWNER_WIKI_EN		varchar(255)	Y	URL of the English wiki (end of URL)

3.4.4 Visitor proposal table

Field	PK/FK	Type	Null	Description
BRAND_NM		varchar(255)	N	Name of the brand
BRAND_TYPE_CD		int	N	Type of brand (1 or 2 allowed)
OWNER_NM		varchar(255)	Y	Name of the owner
BRAND_LINK		varchar(255)	N	Link of the website
BRAND_LOGO		varchar(255)	Y	URL of the picture
INPUT_DATE		datetime	N	Date of proposal
COMMENT		Text	Y	Comment
EMAIL_ADDRESS		varchar(255)	N	Email address of the visitor

Structure to determine (including the date, status, user assigned to validate etc.....)

4 General Functionalities

4.1 Users

4.1.1 Type of users

3 types of user are defined:

- **Visitor**: they access only to the public part of the website and can propose to add a brand using a form
- **Moderator** : they access to the admin part of the website and manage the content of the the repository but can't achieve structural actions (see the listing of actions allowed)
- **Administrator**: they can access and achieve all actions according to the application functionalities and constraints

4.1.2 Authorizations

TBD = To Be Discussed

Actions	Visitor	Moderator	Administrator
Access to the public interface	Y	Y	Y
Propose a new brand	Y	Y	Y
Access to the admin interface	N	Y	Y
Manage types of brands	N	N	Y
Validate a brand proposal	N	Y	Y
Manage owners	N	Y	Y
Manage brands	N	Y	Y
Manage authorizations	N	N	Y
Delete a brand or an owner	N	N	Y

4.2 Log

Each change applied on the database table must be stored in a log table which indicates:

- the user who did the change
- the date and time
- actions (table, fields,...)

Question : it's possible and easy with Django ?

5 Admin Functionalities

5.1 Authorizations

5.1.1 Groups

Groups of user can be added, modified, deleted (standard functionalities of Django)

5.1.2 Users

Users can be added by providing the following information:

- Username
- Email address
- First Name
- Last Name
- aut_user_user_permissions

The user management follows the rules of Django.

5.2 Brand management

5.2.1.1 Listing of brands

Brands are displayed in a table with the following elements:

- BSIN
- Brand name
- Brand logo
- Delete button

The delete button displays a pop-up to ask confirmation.

5.2.2 Validate a brand

When a visitor submits a new brand, a control must be done to check that the brand doesn't exist and that all elements are corrects.

Cases:

- The brand already exists : a mail is sent to inform the visitor and provide the BSIN
- The brand doesn't exist but is not correct : a mail is sent to ask more precisions

It's critical to check that the brand doesn't already exists : a tool helping to search must be provided (to be discussed).

The moderator in charge of the check must be able to modify the name, type and url of the brand and to load another picture.

When brand is validated, the status changes in the visitor proposal table and a BSIN is created in brand table. A mail is sent to the visitor to announce the creation of BSIN.

5.2.3 Add a brand

5.2.3.1 Interface

A text field is used to get the brand name.

1 type of brand is selected among all the types stored in brand_type.

An owner is selected among all the owners of the brand_owner table

A tool propose to upload the logo and automatically put him to a format of 150 * 150 pixel

5.2.3.2 Controls

These fields are required: Brand name and brand type.

Remark: The brand must be unique so it's very important to ensure that the brand is not already used. This could be auto

5.2.3.3 Functional rules

When a brand is added in the repository, The BSIN is randomly generated. The BSIN is composed of 6 alpha numeric characters in **upper case** and **excluding 'O' and '0'** (zero) to avoid confusion.

The picture uploaded is resized to 150px*150px and stored locally **in jpg format** (the storage on amazon will be done in a future release)

5.2.4 Modify a brand

BSIN is a master data and is stable. Name of a brand is linked to the BSIN and so the field BRAND_NM can't change easily. This critical action must be checked by several users before to be done.

- If a brand is not spelled correctly ("Coca cola" instead of "Coca-Cola"), it's possible to change the field
- If a brand change of name, a new BSIN code must be created (with in comments the link between the 2 BSIN) the change of name is forbidden.
- In all other cases, it's not possible to change the brand name

All the other elements can be modified.

5.2.5 Delete a brand

When a brand is deleted, clients of the repository must be informed of it and must have the reason and the potential BSIN replacing the one deleted.

Only administrator can delete a brand.

So, the deletion needs to be logic:

In the Brand table:

- the field *flag_deleted* becomes true
- the explanation is required in the *comment* field
- the date of deletion is indicated in the field *Last_modified*

5.3 Owner management

5.3.1 Listing of owner

Owners are displayed in a table with the following elements:

- Owner name
- Owner logo
- Owner website link

The delete button displays a pop-up to ask confirmation and change the FLAG_DELETE of the brand in the brand table.

5.3.2 Add an owner

5.3.2.1 Interface

The following elements must be field to add a owner

- Name (required)
- Logo (required)
- Website link (required)
- Wikipedia English link

5.3.2.2 Controls

No controls

5.3.2.3 Functional rules

When an owner is added, an owner code is created.

5.3.3 Modify an owner

It's not possible to change the owner code of to exchange the name of an owner codes.

Only little corrections are allowed (problem of spelling in the name, change of website or logo,...). Controls or texts displayed on this interface should help moderators to respect this rule.

5.3.4 Delete an owner

Only owners not assigned to at least one brand can be deleted.

5.4 Brand Type management

There's few elements to manage (add, edit, delete) in a standard way.

The most important thing is to ensure that a type can't be deleted if used by at least one brand.

6 Public Functionalities

6.1 Home page

The home page display very clearly:

- A browser to give to the user an overview of the brand repository
- A link to collaborate entering a brand which doesn't exist
- List of the 10 (or 20) last BSIN added

6.2 Menu

The menu contains the following links:

- Home
- Brand list
- Owner list
- Add a brand
- Download and API

6.3 Form to add a brand

A form is displayed to let visitors propose a brand not existing:

- The name (required)
- The type of brand (only 1 and 2)
- The owner (text field)
- The URL (required)
- A logo to upload
- An email address (with confirmation)
- A comment text area

The email address is used to inform the visitor of the creation of a BSIN for this brand

A captcha can be added to avoid attacks

6.4 Brand List

The brans list displays the following information:

- BSIN
- Name
- Logo
- Website link

A search engine is configured to search on the name of the BSIN.

BSIN, Name and Logo are clickable and display the brand page containing:

- BSIN
- Name
- Owner
- Logo
- Type of brand
- Link

Owner is clickable and leads to the owner's page.

To be discussed in an issue: the mode of navigation on the brands

6.5 Owner list

The owner list displays the following information:

- Name of the owner
- Logo
- Website link

The Name and logo are clickable and display the owner page.

The owner page contains the following information:

- Name
- Logo
- Website link
- English Wikipedia website

The brand list of owner is displayed in the same way as the brand list view.