

PROJET INDIVIDUEL

SUJET 66

Gestion de corpus

Anne-Sophie SAINT-OMER

Tutrice : Fanny CHEVALIER

Janvier 2015 - Mai 2015

Remerciements

Ce projet a été réalisé dans le cadre de la première année de Master informatique à l'Université de Lille 1, à Villeneuve d'Ascq.

Je tenais à remercier ma tutrice de projet, Madame Fanny Chevalier pour m'avoir fait confiance en m'attribuant ce sujet ainsi que pour la qualité de l'encadrement dont j'ai pu bénéficier tout au long de ce travail. Elle a toujours été très disponible et encourageante, j'ai appris énormément à ses côtés.

Étudiante en première année de Master informatique à l'Université de Lille 1. Je souhaitais réaliser le projet nommé "Gestion de corpus" encadré par Madame Fanny Chevalier, chercheuse dans le domaine de l'interaction Homme-Machine à l'Inria.

Un corpus est un ensemble de documents, artistiques ou non (textes, images, vidéos) regroupés dans une optique précise. On peut utiliser des corpus dans plusieurs domaines : études littéraires, linguistiques, scientifiques, philosophie etc.

Ce sujet a en effet suscité ma curiosité, il concernait les bases de données, domaine qui m'a toujours fortement intéressée. De plus, lors de l'entretien pour ce projet, j'ai compris que l'interaction Homme-Machine allait être une grande partie de mon travail et c'est un sujet que je souhaitais approfondir.

Table des matières

Remerciements	1
1 Présentation du projet	5
1.1 Exemple de gestion de corpus	5
1.2 Le but du projet	6
2 Analyse du projet	7
2.1 Méthodologie - Interaction Homme-Machine	7
2.1.1 Réflexion sur l'IHM	9
2.1.2 Prise de décision sur l'IHM	12
2.2 Choix des technologies	14
2.2.1 Côté client	14
2.2.2 Côté serveur	15
2.2.3 SQLite	15
3 Implémentation	16
3.1 Hiérarchie de mon projet	16
3.2 Images	17
3.2.1 Stocker des images	17
3.2.2 Afficher des images	17
3.3 Base de données - SQLite	17
3.4 Les difficultés rencontrées lors du développement	17
3.5 Travail effectué pour l'application	18
Conclusion	20
Webographie	21

Table des figures

1	Images - Corpus Souris	5
2	Site donné en exemple pour visualiser des données - Visualisation pour l'ajout d'entrées	6
3	Sketching User Experience	7
4	The design of everyday things by Donald Norman	8
5	Exemple - Plaques de cuisson	8
6	Prototype papier	9
7	Défiler les éléments de la base de données	11
8	Vue d'ensemble de la base de données	12
9	Principe de filtres sur les éléments de la base	13
10	Champs éditables pour ajouter le nom de la base de données et la table associée - Bulle contenant deux champs éditables	13
11	Fonctionnement des différents langages	15
12	Hiérarchie du projet	16
13	Aperçu du contenu d'un fichier avec l'extension "sqlite"	17
14	Interface en cours de finalisation	19

1 Présentation du projet

1.1 Exemple de gestion de corpus

Le premier exemple qui m'a été donné concernant la gestion de corpus est l'histoire de la souris en informatique. L'utilisateur souhaite créer une base de données à partir d'un corpus d'images contenant différentes photos de souris informatique (Voir Figure 1). Il souhaite enregistrer des informations sur tous les types de souris ayant vu le jour.



FIGURE 1 – Images - Corpus Souris

Si on défile les images une par une (Voir Figure 1).

La souris numéro 1 :

En voyant cette image, nous pouvons décider de stocker les informations suivantes :

- Le nombre de boutons présents sur la souris, dans un champ nombreDeBoutons.
- Préciser la présence d'une molette, dans un champ molette.
- Préciser la présence d'un fil pour connecter la souris à l'ordinateur, dans un champ fil.

La souris numéro 2 :

Nous pouvons voir que cette souris est beaucoup plus moderne, nous pouvons donc avoir l'idée d'ajouter à la liste d'informations :

- L'année de sortie de cette souris, dans un champ annee.

Les champs précédents, c'est à dire : nombreDeBoutons, molette et fil pourront être complétés pour cette souris.

La souris numéro 3 :

Nous pouvons voir une nouvelle catégorie de souris. Nous pouvons donc par exemple ajouter :

- Un nom correspondant à la catégorie de la souris, dans un champ categorie.
- La marque de la souris, dans un champ marque.

La souris numéro 4 :

Nous pouvons voir une souris très récente et originale. Nous pouvons donc penser à ajouter :

— Un prix, dans un champ prix.

Nous pouvons constater que les informations que nous souhaitons stocker ne peuvent pas toujours être anticipées. Nous les trouvons en défilant les images au fur et à mesure. En ajoutant un champ, l'utilisateur peut se rendre compte qu'il doit modifier les informations de sa base. Par exemple, le trackpad (Voir Figure 1 - Photo 3) est une catégorie de souris bien particulière, si on stocke la catégorie, quelles catégories attribuées aux premières souris ? Ou encore le prix de la dernière souris sortie sur le marché (Voir Figure 1 - Photo 4) peut être pertinent mais la première souris (Voir Figure 1 - Photo 1) n'est plus vendue, quel prix ajouter ? Il faut permettre à l'utilisateur de faire ce qu'il souhaite au fur et à mesure et ne pas le limiter dans ces choix.

1.2 Le but du projet

Ce projet consiste à réaliser l'application permettant à l'utilisateur de créer pour un sujet donné une base de données sous-jacente. L'utilisateur pourra ajouter des entrées à cette base de données et en modifier le contenu. La structure de la base de données est dynamique. Les colonnes doivent être facilement modifiables. Cette application permet de faciliter le travail de catégorisation de l'utilisateur tout au long de son travail sans que celui-ci n'ait à déterminer une structure définitive. La structure de la base de données n'est donc pas fixée a priori.

Pour visualiser la base de données créée, nous pourrions afficher les éléments du corpus sur une page web qui contiendrait un mécanisme de requêtes tel que des filtres. Les internautes pourront contribuer à l'aide de cette page web à enrichir la liste en ajoutant de nouvelles entrées.

Pour afficher ces informations, on pourrait prendre le site web <http://dataphys.org/list> pour exemple (Voir Figure 2) Ce site présente une liste chronologique de visualisations physiques et artefacts liés. Il est maintenu par Pierre Dragicevic et Yvonne Jansen. Il est possible pour les utilisateurs de proposer une autre visualisation physique qui leur paraît intéressante et qui pourra être ajoutée au site. Dans cet exemple, la structure est fixée, on ne peut pas ajouter d'entrées.

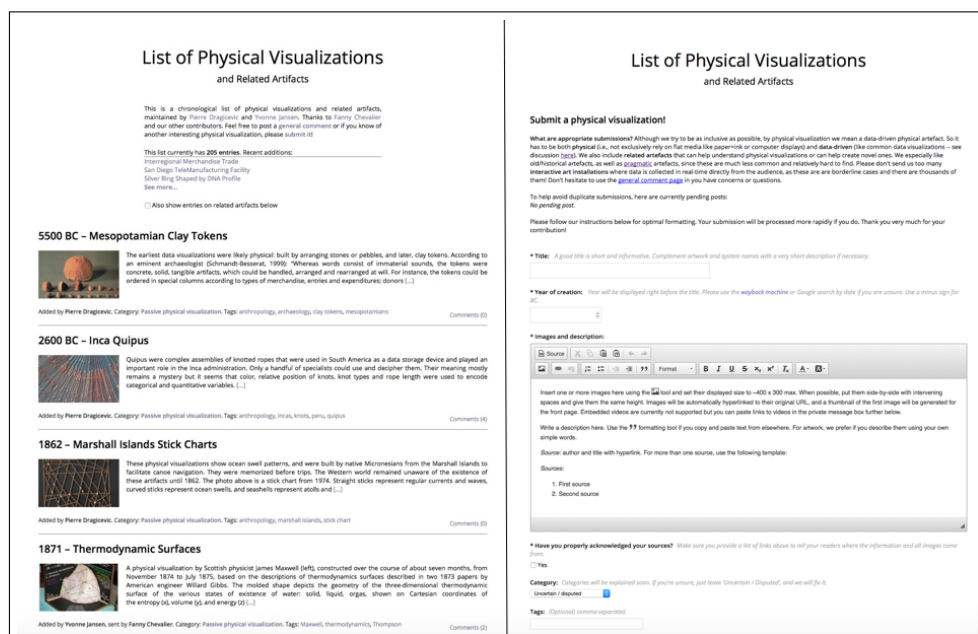


FIGURE 2 – À gauche : Site donné en exemple pour visualiser des données - À droite : Visualisation pour l'ajout d'entrées

2 Analyse du projet

2.1 Méthodologie - Interaction Homme-Machine

En première année de Master informatique, au cours du second semestre, les étudiants ont la possibilité de choisir les matières qu'ils souhaitent étudier. J'ai décidé de suivre l'option Interaction Homme-Machine. Cette option m'a donné des bases solides pour la réalisation de mon projet. J'apprenais la théorie lors de mes cours et je pouvais la mettre en pratique en réalisant mon projet en parallèle. Pour m'aider dans la réalisation de ce projet, ma tutrice m'a présenté le livre "Sketching User Experience"¹. Il contient différents exemples pour représenter des prototypes d'interfaces graphiques.

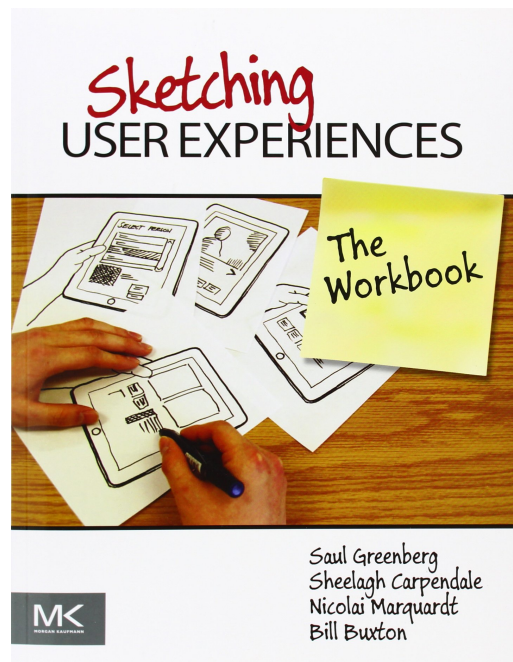


FIGURE 3 – Sketching User Experience

Il existe trois niveaux différents de fidélité pour représenter des interfaces graphiques.

- Fidélité basse - Low-Fidelity

Ce prototype est souvent présenté sous la forme d'un croquis dessiné à la main. Il est très rapide à réaliser car on ne développe pas. Il permet de générer de nombreuses idées à moindre coût et on peut identifier facilement ce qui peut être modifié dans l'interface.

- Fidélité haute - High-Fidelity

Ce prototype prend plutôt la forme d'un véritable logiciel qui fonctionne mais qui n'a pas encore la robustesse d'un produit fini. Il est plus coûteux mais on peut le tester dans des conditions réelles.

- Fidélité moyenne - Middle-fidelity

Ce prototype ressemble à un prototype de fidélité basse et possède moins de détails qu'un prototype de fidélité haute. On commence à se rapprocher de la réalité.

Pour représenter une interface graphique dans le contexte low-fidelity, le dessin est la plus immédiate des représentations mais pas la plus dynamique. Représenter une action demande beaucoup de travail car il faut créer plusieurs dessins pour représenter les changements dans l'interface. En ce qui concerne les post-its, cette idée de représentation est dynamique car modulable. Les post-its peuvent avoir plusieurs tailles, différentes couleurs. De plus, on peut les déplacer comme on le souhaite. Si un post-it ne convient pas, il suffit de prendre un nouveau post-it et de jeter l'ancien.

1. Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt and Bill Buxton. Sketching User Experiences, the workbook. Morgan Kaufmann Press. 2011.

Réaliser une interface graphique est loin d'être trivial. Il est important de passer par différentes étapes car cela permet de consolider un concept et d'identifier les erreurs de conception rapidement. Il ne faut jamais développer directement. Prenons un exemple de tous les jours, celui des boutons permettant d'allumer, d'éteindre et de régler les plaques de cuisson. Exemple tiré du livre "The design of everyday things"².

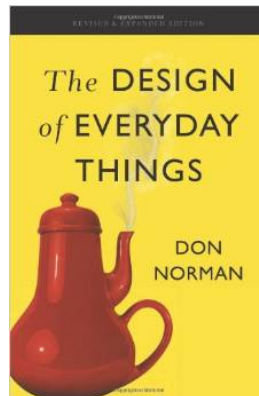


FIGURE 4 – The design of everyday things by Donald Norman



FIGURE 5 – Exemple - Plaques de cuisson

Donald Norman met l'accent sur l'importance de consacrer une attention toute particulière aux actions que l'utilisateur souhaite réaliser et comment il va la réaliser via l'interface. Les boutons associés aux anciennes plaques de cuisson n'étaient pas vraiment évidents à utiliser comme on peut le voir sur l'image de gauche de la Figure 5 . Le dessin destiné à préciser quel bouton est associé à quelle plaque n'est pas révélateur. De plus, dans quel sens doit on tourner le bouton pour monter ou descendre la température des plaques de cuisson ?

La plaque de cuisson moderne présentée sur l'image de droite de la Figure 5 indique à l'aide d'un point plus foncé l'emplacement de la plaque correspondante. Pour le réglage de la température, la plus petite température est représentée par une petite flamme, la plus haute température par une plus grosse flamme. On ne se pose pas de questions.

2. Donald Norman. The design of everyday things. New York : Basic Books. 1988.

2.1.1 Réflexion sur l'IHM

J'ai suivi la méthode de conception décrite précédemment en commençant par la première étape : low-fidelity.

Voici un scénario éventuel :

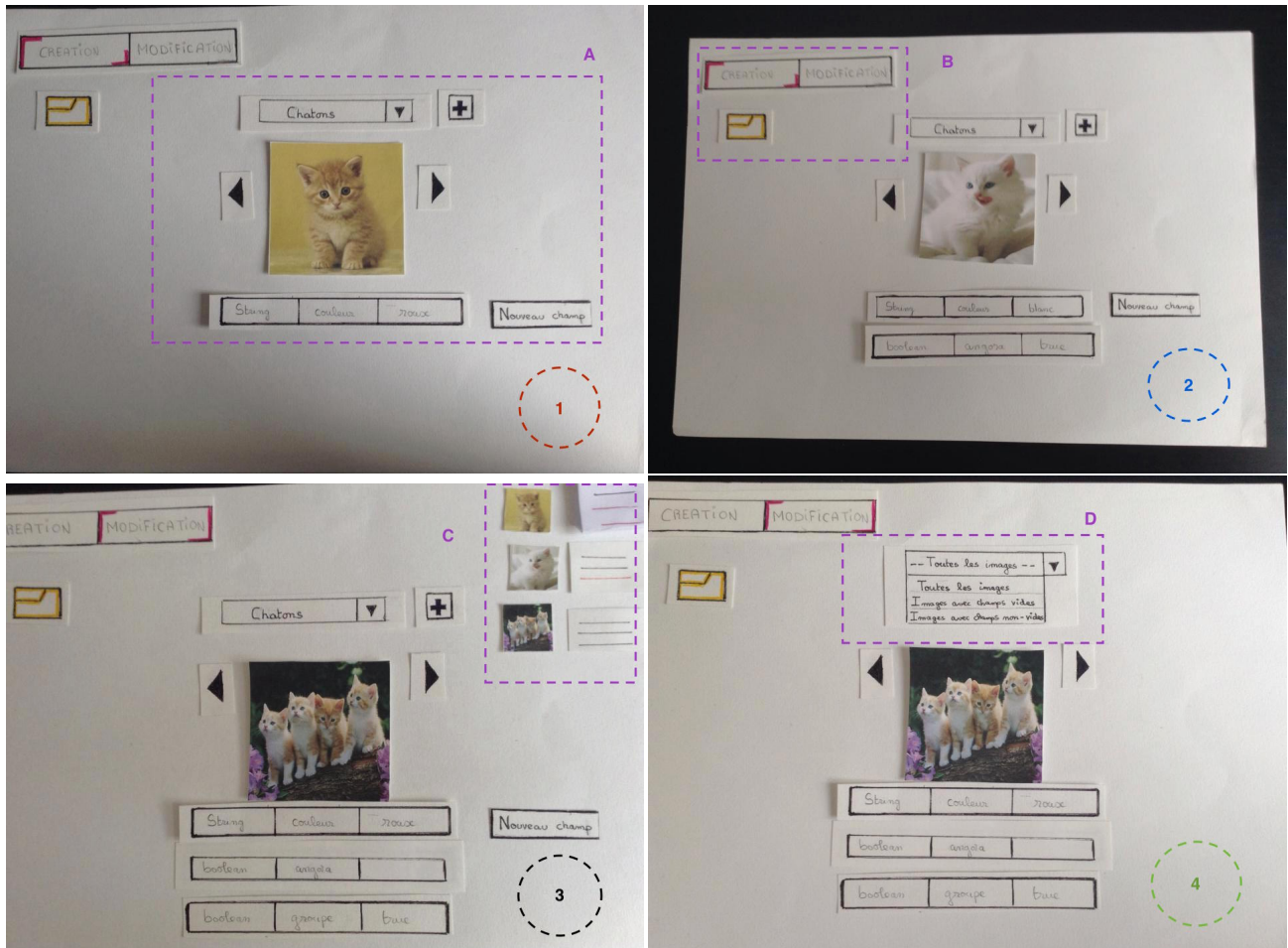


FIGURE 6 – Prototype papier

On peut voir sur l'Image 1 :

L'ensemble A, il correspond à la partie principale de l'interface et il contient :

- Une liste déroulante affichant les bases de données disponibles.
- Un bouton permettant de créer une nouvelle base de données.
- Une image correspondant à une image du corpus de la base de données.
- Les champs associés à l'image dans la base de données.
- Un bouton permettant de créer un nouveau champ dans la base de données.
- Des flèches d'interaction permettant de passer à l'image précédente ou à l'image suivante.

On peut voir sur l'[Image 2](#) :

L'ensemble B, il contient :

- Deux onglets : Création et Modification. L'onglet à un bord rose quand il est sélectionné.
- Un dossier permettant de charger des images dans la base choisie.

On peut voir sur l'[Image 3](#) :

L'ensemble C, il contient :

- La vue d'ensemble qui affiche toutes les miniatures des photos de la base de données.

On peut voir sur l'[Image 4](#) :

L'ensemble D qui contient :

- Une liste déroulante permettant d'appliquer des filtres sur les éléments de la base de données.

Pour réaliser ce prototype, je me suis concentrée sur un sujet très simple mais révélateur, les animaux et plus précisément les chatons. En défilant les photos des chatons que j'avais sélectionnées, je me suis rendue compte des champs possibles que je devais créer au fur et à mesure de la consultation de la base de données. Pour la première photo que j'ai examiné, la première chose qui m'a frappé est la couleur rousse du chaton ce qui m'a valu d'ajouter un champ "couleur". Puis en continuant de défiler les images, j'ai pu voir la photo d'un chaton angora (Voir [image 2](#) - Figure 6). C'était un champ que je n'avais absolument pas anticipé. Ce qui m'a fait réfléchir quant à l'ajout d'un nouveau champ. Doit-on attribuer la valeur du champ angora à false pour toutes les autres photos examinées précédemment ? Si je n'avais pas pensé à ce champ avant c'est peut-être tout simplement parce qu'aucun des autres chatons n'étaient angora. Par la suite, on peut voir une photo de chatons en groupe (Voir [image 3](#) - Figure 6). On ajoute donc un champ groupe. On veut voir dans la vue d'ensemble que les nouveaux champs sont rouges pour les miniatures précédentes. Par exemple, pour le chaton roux présent sur l'image 1 (Voir [image 1](#) - Figure 6), il manque la valeur du champ angora et la valeur du champ groupe (Voir [image 3](#) - Figure 6). Sur l'image 4 (Voir [image 4](#) - Figure 6), on peut voir l'utilisateur choisir un filtre dans la liste déroulante permettant d'afficher certains éléments, par exemple, les éléments pour lesquels il manque la valeur d'un champ.

En réalisant ce prototype papier, j'ai initialement conçu une partie création et une partie modification.

Dans l'onglet "Création" on peut trouver différentes fonctionnalités

Création de base de données

L'utilisateur souhaite créer une nouvelle base de données. Il doit renseigner le nom de sa base de données ainsi que le nom de la table associée. L'utilisateur peut consulter toutes les bases qu'il peut exploiter à l'aide d'une liste déroulante.

Ajouter des photos à la table de la base de données.

Ces photos représentent le corpus à partir duquel l'utilisateur va créer sa base de données. Lors du chargement du dossier, quand l'utilisateur clique sur l'image représentant un dossier, trois champs sont créés. On associe chaque photo à un champ représentant un identifiant unique, un autre contenant une image redimensionnée, elle apparaîtra au centre de la fenêtre principale permettant son exploitation par l'utilisateur. Pour finir, un champ contenant une image miniature qui apparaîtra dans la vue d'ensemble de la base. La vue d'ensemble correspond à un affichage de toutes les miniatures des images de la base de données. Elles sont entourées d'un cadre rouge lorsqu'un champ n'est pas complété.

Ajouter des champs à la table correspondant à la base de données sélectionnée

L'utilisateur souhaite ajouter un nouveau champ à la table correspondant à la base de données sélectionnée. Lors du clic sur un bouton "Nouveau Champ", l'utilisateur renseigne le nom du champ qu'il souhaite créer et sélectionne dans une liste déroulante son type, cela lui permet de ne pas faire d'erreur puisque chaque SGBD nomme les types différemment. De plus, lors de l'ajout d'un nouveau champ, ce dernier sera vide pour les documents traités précédemment. Il faut donner à l'utilisateur la possibilité de choisir si il remplit ce nouveau champ à l'aide d'une valeur par défaut ou si il le complètera manuellement plus tard.

Choix de l'affichage des données de la base.

La conception nous a permis d'identifier différentes questions.

Question 1 :

Où doit-on afficher le nom des champs ainsi que la valeur qui leur correspond ? Sur la gauche pour gagner de la place et ainsi réaliser une vue d'ensemble complète sur le côté ? Ou au centre pour visualiser correctement le tout ?

Parcourir les éléments de la base de données.



FIGURE 7 – Défiler les éléments de la base de données

Lorsque l'utilisateur souhaite afficher le contenu d'une base de données, il la sélectionne dans la liste déroulante. C'est le premier élément de la base qui s'affiche, c'est à dire la première photo de la base de données correspondant à l'identifiant 0 avec les éventuels champs qui lui sont associée. On souhaite afficher un élément à la fois. Pour parcourir la base en passant à l'élément précédent ou à l'élément suivant, il utilise la flèche de gauche ou de droite (Voir Figure 6).

Editer la valeur des champs

Question 2 :

La valeur des champs doit-elle être éditée en permanence ou non ? On pourrait donc imaginer deux orientations pour cette application :

Une application destinée à un utilisateur novice.

- C'est à dire que tout est contrôlé. L'utilisateur serait averti par des pop-ups qu'il n'a pas le droit de faire cette action par exemple.

Une application destinée à un expert.

- C'est à dire quelqu'un d'expérimenté et qui utilise l'application en toutes connaissances de cause. Il n'y aura donc pas de pop-ups incessantes.

Vue d'ensemble de la base de données

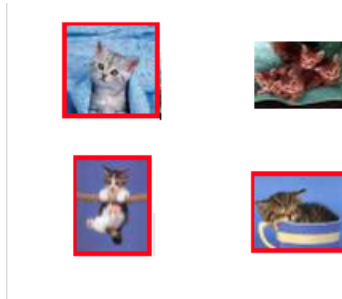


FIGURE 8 – Vue d'ensemble de la base de données

Cette vue affiche toutes les miniatures des photos de la base de données. Lorsqu'un élément de la base contient au moins un champ vide la miniature de l'image correspondant à cet élément est entourée d'un cadre rouge (Voir Figure 7). Elle permet à l'utilisateur de connaître plus rapidement les images qu'il doit traiter.

Dans l'onglet Modification, on peut trouver :

Si l'utilisateur souhaite se soucier du contenu d'un champ en particulier dans la base de données, il faut lui en donner la possibilité. Par exemple, si il ne souhaite voir que les images où le champ "couleur" est vide, il faudra donc lui permettre d'appliquer un filtre. J'ai donc pensé à une liste déroulante permettant d'appliquer des filtres sur les éléments de la base de données choisie.

- Un filtre permettant d'afficher les images avec tous les champs remplis.
- Un filtre permettant d'afficher les images avec certains champs complets.
- Un filtre permettant d'afficher les images avec certains champs non remplis.

2.1.2 Prise de décision sur l'IHM

Au fur et à mesure du développement, nous avons pris différentes décisions :

Supprimer les onglets Création et Modification.

L'ensemble des éléments pouvait en effet être présent sur la même page :

- La liste déroulante permettant de sélectionner la base de données sur laquelle on souhaite travailler.
- Le bouton permettant de créer une nouvelle base de données.
- Le bouton permettant de créer un nouveau champ dans la base de données sélectionnée dans la liste déroulante.
- Le dossier permettant de charger les photos représentant le corpus de la base.
- La liste déroulante permettant d'appliquer des filtres pour afficher certains éléments de la base.

Afficher les valeurs de la base de données.

Réponse à la question 1 :

Nous avons décidé d'afficher les données au centre de l'interface sous l'image correspondante. C'est une des parties que l'utilisateur va le plus exploiter. Il aurait donc été dommage de ne pas la placer au centre et de ne pas la mettre en valeur.

Filtres sur les éléments de la base de données.



FIGURE 9 – Principe de filtres sur les éléments de la base

La liste déroulante permettant d'appliquer des filtres sur les éléments de la base de données a été remplacée. Si on appliquait le principe d'une liste déroulante, l'utilisateur aurait été vite limité quant au choix des champs sur lesquels il souhaite travailler. De plus, créer un filtre pour chaque champ, ainsi que pour toutes les combinaisons possibles de champs aurait été fastidieux pour le développement et illisible pour l'utilisateur. Désormais, l'utilisateur coche les checkboxs correspondantes aux champs sur lesquels il souhaite se concentrer. Dans cet exemple (Voir Figure 8), si l'utilisateur coche le champ "couleur" et qu'il utilise la flèche représentant une avance rapide, il passera à la prochaine image dont le champ "couleur" est vide. L'utilisation de filtres est ainsi simplifiée et on peut avoir plusieurs filtres à la fois.

Interface graphique épurée.



FIGURE 10 – À gauche : Champs éditables pour ajouter le nom de la base de données et la table associée - À droite : Bulle contenant deux champs éditables

Nous souhaitons une interface épurée. Nous voulions donc voir apparaître au clic sur le bouton "Création de base de données", les deux champs éditables permettant d'indiquer le nom de la base de données et de la table. Il est inutile qu'ils se trouvent en permanence sur l'interface. Au clic sur le bouton permettant la création de la base de données, une bulle apparaît permettant de voir des champs éditables que l'utilisateur complètera avec le nom de la future base de données et le nom de la table associée à cette base.

Choix de l'orientation de l'application : expert.

Réponse à la question 2 :

Nous avons choisi d'orienter l'application côté expert. De mon point de vue utilisateur et comme vu dans l'option interaction Homme-Machine au cours de mon second semestre, il n'y a rien de plus désagréable que de voir apparaître des pop-ups au moindre problème.

2.2 Choix des technologies

Les différentes tâches à réaliser pour ce projet se décomposent en deux parties. La première concerne l'application permettant au chercheur de réaliser sa propre base de données, la seconde partie concernera l'affichage sur une page web permettant aux internautes de participer à son enrichissement.

En ce qui concerne le choix des langages : nous avons porté notre choix sur deux possibilités.

Java, un langage que je connaissais très bien, permettant de communiquer facilement avec une base de données mais qui avec Swing ne permettait pas de réaliser une interface graphique très agréable.

Du développement web que je connaissais peu mais permettant de faire ce que l'on désirait en terme d'interface graphique et de la partager sur le réseau plus facilement.

Le développement web apportant plus d'avantages nous nous sommes donc tournées vers le HTML5 associé à PHP, à JavaScript ainsi que JQuery et des requêtes Ajax.

L'interface et les interactions seront gérées côté client. Les requêtes SQL, côté serveur.

En ce qui concerne la base de données, SQLite semblait convenir car cette bibliothèque est réputée pour être légère. De plus, lorsque la base est créée, le contenu se résume à un simple fichier avec l'extension "sqlite".

La hiérarchie de mon projet se trouve Figure 12.

2.2.1 Côté client

Côté client, on utilise :

HTML5 - CSS

On affiche le contenu de base de l'interface graphique de l'application.

JavaScript : jQuery - Write less, do more.

jQuery, est une librairie JavaScript. Elle permet d'accéder à toute la puissance de JavaScript en utilisant des instructions simples, logiques, faciles à comprendre et à maintenir. jQuery détecte les interactions de l'utilisateur avec l'interface.

Ajax

Les requêtes Ajax permettent d'exécuter du code PHP suite à une interaction avec l'interface graphique côté utilisateur. Lorsque l'utilisateur interagit avec l'interface graphique, par exemple au clic sur le bouton permettant de créer une nouvelle base de données. JQuery détecte le clic sur le bouton "Création de base de données", ensuite une requête Ajax est exécutée, elle appelle le fichier "creationBDD.php" qui va exécuter le code attendu par l'utilisateur.

2.2.2 Côté serveur

Côté serveur, on utilise :

PHP

Dans ce projet, les fichiers PHP contiennent des requêtes SQL. Par exemple, lors de la perte de focus sur un champ éditable correspondant à la valeur d'un champ de la base données, on lance un fichier qui mettra à jour le champ à l'aide d'une requête SQL.

Serveur Apache

Un serveur Apache est utilisé pour exécuter le code PHP approprié à l'action réalisée par l'utilisateur. Ce serveur est géré par le logiciel MAMP. MAMP est réputé pour sa simplicité. Une simple interface graphique permet de lancer un serveur Apache. Les fichiers du projet doivent être stockés dans le dossier "htdocs" du dossier MAMP. Les pages web sont ensuite accessibles en local.

2.2.3 SQLite

La base de données est gérée côté serveur. Lors de l'affichage, l'insertion, la mise à jour ou la suppression de données on appelle le fichier correspondant à la base de données et on réalise des requêtes SQL sur ce fichier.

Ce schéma résume le fonctionnement de ces langages entre eux.

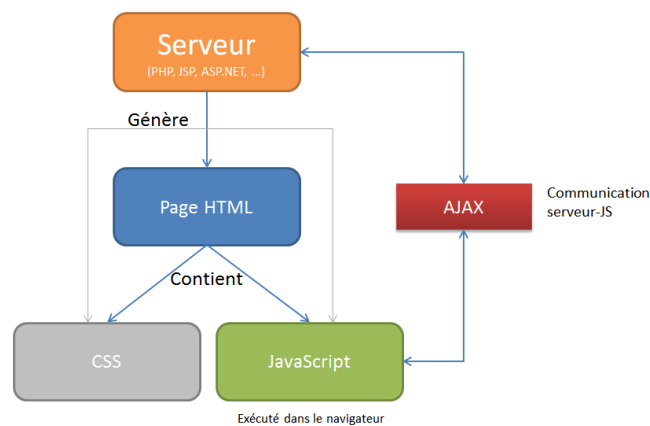


FIGURE 11 – Fonctionnement des différents langages

3 Implémentation

3.1 Hiérarchie de mon projet

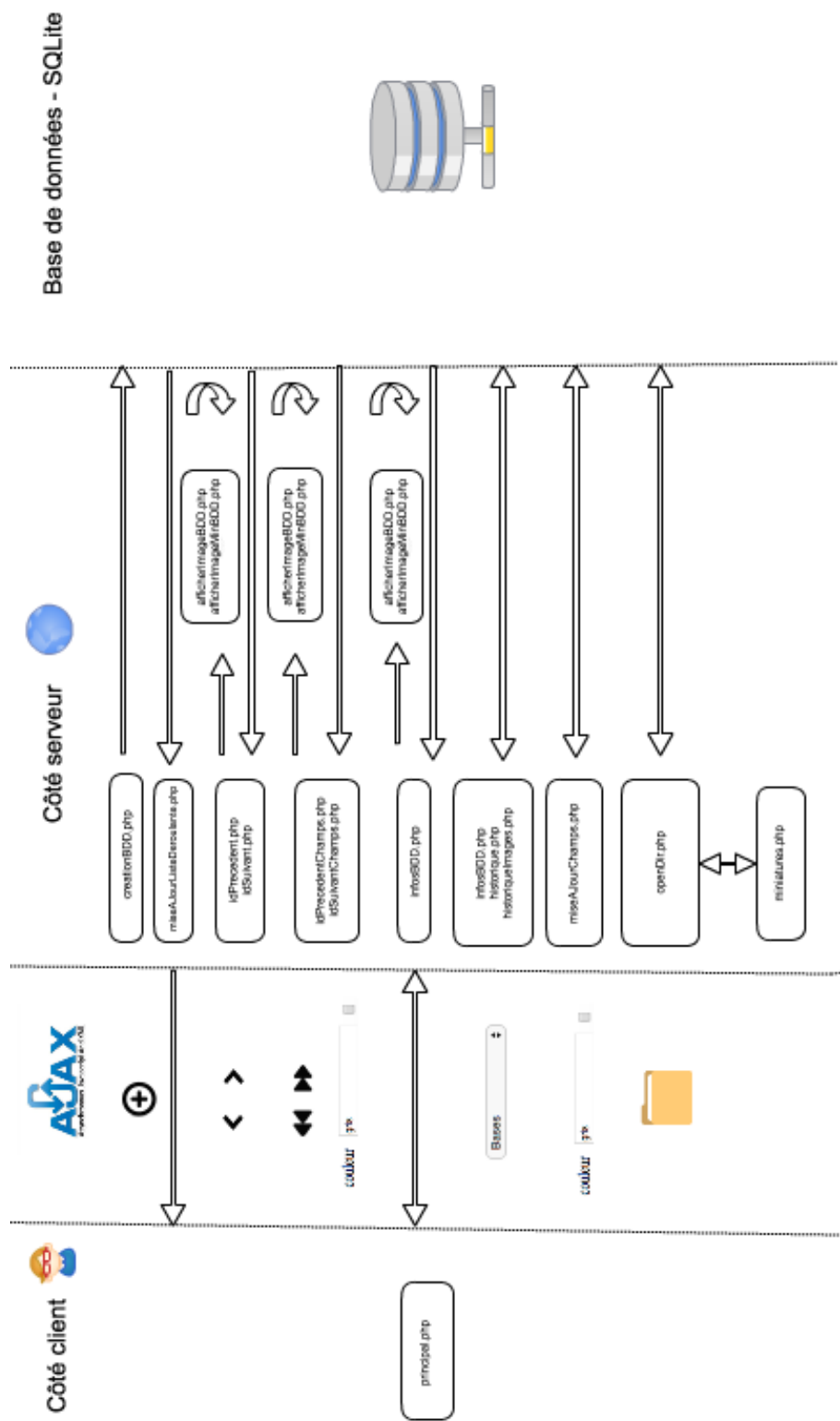


FIGURE 12 – Hiérarchie du projet

Mon projet contient un fichier nommé "principal.php". Il fonctionne côté client. Ce fichier est le centre du projet. Il contient les éléments présents initialement comme par exemple la liste déroulante contenant la liste des bases de données exploitables ou encore le dossier permettant de charger les images. Il contient également toutes les requêtes Ajax appelées par jQuery. Chacune de ces requêtes appelle un fichier PHP réalisant une action bien précise. Par exemple, lors du clic sur le bouton "Nouveau Champ", un script "nouveauChamp.php" est lancé. Les fichiers PHP contiennent chacun différentes requêtes SQL dont j'exploite les résultats pour obtenir les informations que je souhaite afficher.

3.2 Images

3.2.1 Stocker des images

Nous avons choisi de stocker les images en dur dans la base. Ainsi un simple fichier SQLite permet d'avoir toutes les informations nécessaires concentrées au même endroit. De plus, si on charge une image via un URL, rien n'est dit que ce lien existera encore après un certain temps.

3.2.2 Afficher des images

Le contenu d'une image stockée dans une base de données ne peut être affiché au milieu d'un langage web car elle serait interprétée ainsi et des caractères spéciaux apparaîtraient. Elle doit être générée par un autre script et ce dernier doit être appelé dans une balise image HTML.

3.3 Base de données - SQLite

Lors de la création d'une base de données, un fichier avec l'extension "sqlite" est créé. Tous les fichiers correspondant aux bases de données sont stockés dans un dossier "Bases" de mon projet. Nous avons choisi de créer une table par base de données. Il ne semblait pas nécessaire d'avoir plusieurs tables car chaque corpus correspond à un sujet donné. Les autres tables nous venant à l'esprit correspondaient à des catégories, ce qui ne semble pas essentiel pour ce sujet. On peut ouvrir un fichier SQLite à l'aide de logiciel comme SQLite Free - Datum.

	idTable2	imageTable2	imageMinTable2	couleur	groupe
0	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	gris	<null>
1	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	maron	true
2	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	<null>	<null>
3	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	<ffd8ffe0 00104a46 494600	<null>	<null>

FIGURE 13 – Aperçu du contenu d'un fichier avec l'extension "sqlite"

3.4 Les difficultés rencontrées lors du développement

Lors de mon DUT informatique, j'ai étudié la programmation web et jusqu'à maintenant je n'avais pas eu l'occasion de pratiquer à nouveau ces langages de programmation. Ce projet m'a donc permis de retrouver mes habitudes mais également d'apprendre de nouvelles choses. Je devais concevoir tout en apprenant à nouveau les syntaxes des différents langages, ce qui n'était pas évident.

J'ai également dû comprendre le fonctionnement de SQLite : SQLite est assez récent et a subi quelques modifications, j'ai donc dû trier les informations que je trouvais sur l'internet.

Enfin, ce projet démarrait de zéro. J'ai donc dû réfléchir à l'interaction Homme-Machine, au choix des technologies, des langages de programmation et bien évidemment développer. Au commencement, je voulais tout réaliser en même temps. J'ai appris à mieux gérer mon projet grâce à ma tutrice. J'ai compris qu'il valait mieux un code qui fonctionnait en dur, que plusieurs fichiers en cours d'élaboration qui ne fonctionnaient pas. On constate qu'on avance et c'est beaucoup plus encourageant.

3.5 Travail effectué pour l'application

À ce jour, il est possible dans mon application de :

- Créer une base.
- Choisir une base et afficher ses informations.
- Télécharger un dossier d'images qui s'ajoute dans la base en créant pour chacune : un identifiant, une image redimensionnée et une miniature.
- Ajouter un champ.
- Mettre à jour un champ.
- Naviguer entre les éléments de la base en fonction de leurs identifiants.
- Naviguer entre les éléments de la base en fonction des champs choisis à l'aide des checkboxes cochées.
- Visualiser une vue d'ensemble de la base à l'aide de miniatures entourées de rouge si tous les champs ne sont pas remplis.

J'apporterais quelques améliorations avant ma soutenance à ce projet que ce soit sur l'interface ou les fonctionnalités.

Voici l'interface en cours de finalisation.

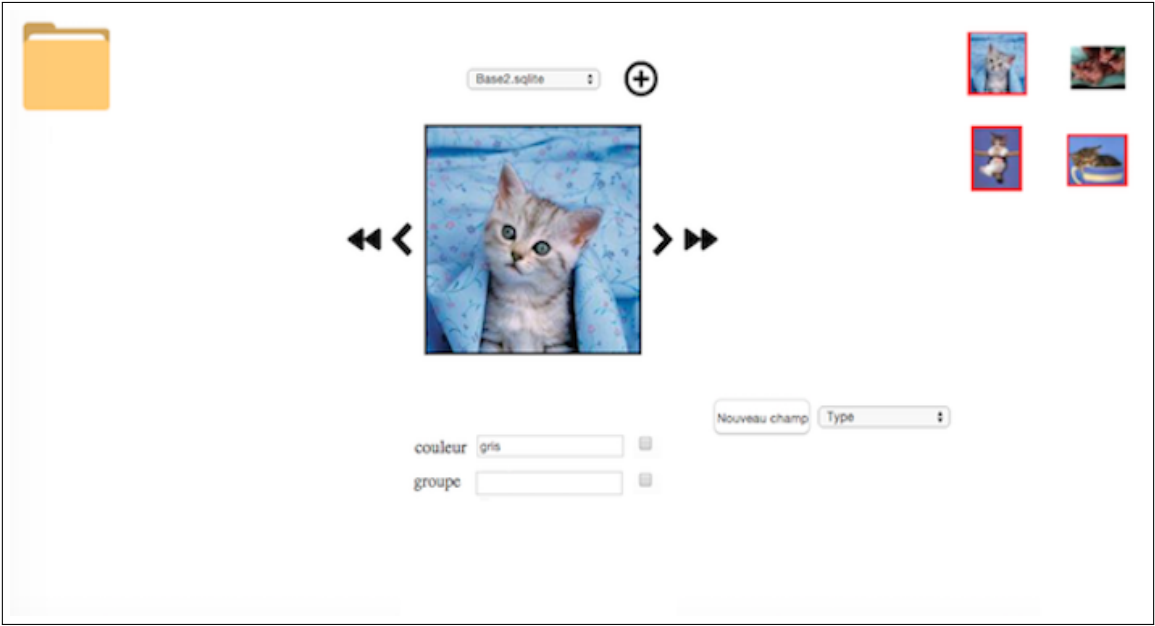


FIGURE 14 – Interface en cours de finalisation

Conclusion

En conclusion, ce projet a été pour moi une opportunité. Il a été très instructif sur le plan technique ainsi que sur le plan personnel.

J'ai en effet acquis des compétences solides en interaction Homme-Machine en découvrant de nombreux aspects de ce domaine. Ce projet m'a aussi permis de retrouver mes habitudes et d'apprendre de nouvelles facettes du développement web. J'ai également pu connaître une nouvelle technologie, le moteur de base de données SQLite.

De plus, j'ai appris à gérer un projet partant d'un simple énoncé auquel j'ai dû réfléchir sur tous les plans ce qui a été très bénéfique et valorisant.

Je pense avoir acquis des connaissances essentielles qui pourront appuyer ma candidature en seconde année de Master dans les spécialités choisies, e-Services et IAGL.

Enfin, mon projet pourra être utilisé dans le cadre de la recherche par ma tutrice, le sujet du corpus sur lequel elle travaille concerne l'évolution des images présentes dans les livres scolaires pour enfants.

Pour terminer, je peux affirmer que ce projet fut une expérience enrichissante, qui pourra valoriser mes compétences dans l'avenir.

Webographie

Projet :

Le dépôt contenant le code source du PJI :

<https://github.com/soDisaster/PJI>

Documentation :

PHP - <https://php.net/manual/fr/index.php>

jQuery - <https://jquery.com/>

Rapport :

Ce rapport a été écrit et mis en forme à l'aide d'Overleaf.

<https://www.overleaf.com/>

Schéma :

Le schéma concernant la hiérarchie du projet a été réalisé à l'aide de Cacao.

<https://cacao.com/lang/fr/>