

Table des matières

Introduction	3
I. Présentation des cas cliniques à l'Ecole Vétérinaire d'Alfort	4
A. Historique	4
B. Insertion au sein de l'enseignement	4
1. Consultations de médecine et hospitalisation des animaux	4
2. Gestion des animaux hospitalisés	5
3. Gestion des dossiers des animaux	6
4. Préparation des cas cliniques	6
C. Etude du déroulement de la présentation des cas cliniques	6
1. Diversité des outils de présentation	7
2. Critique des présentations	7
3. Variations autour d'un plan classique	8
D. Devenir des cas	8
E. Idée de création de l'outil d'aide à la réalisation de cas cliniques et à leur mise en ligne ...	9
II. Réalisation du projet	10
A. Etude des cas et création d'un modèle polyvalent de cas cliniques	10
1. Base de travail	10
2. Présentation du modèle de cas cliniques	10
B. Les bases informatiques : fonctionnement de l'Internet	15
1. Historique de l'Internet	15
2. Les pages web statiques	15
3. Les pages web dynamiques	17
4. Les bases de données et les SGBD	18
C. Choix informatiques pour le projet	19
1. Choix des langages	19
2. Choix du serveur et du SGBD	20
3. Choix des programmes	20
a) Programme d'écriture du code et des pages	20
b) Programme de test	21
D. Création d'une base de données	22
1. Principes de création d'une base de données : la méthode MERISE	22
2. Analyse des besoins et inventaire des informations	23
3. Bilan des flux par rapport à la base de données	23
4. Création du Modèle Conceptuel de Données (MCD)	24
a) Le modèle entité-relation	24
b) Ecriture du Modèle Conceptuel de Données (MCD)	25
c) Ecriture du Modèle Logique de Données (MLD)	28
5. Création des tables	30
E. Création du site Web	33
1. Architecture du site : utilisation de pseudo-frames	33
2. Sécurité du site	35
a) Affichage des pages et des menus	35
b) Identifiant de session	36

c)	Navigation entre les pages.....	38
d)	Configuration de la connexion avec la base de données.....	38
e)	Bilan sur la sécurité du site	39
3.	Etude de choix techniques.....	39
a)	Valeurs des listes dans les formulaires.....	39
b)	Les pages comportant des formulaires simples.....	40
c)	Les pages comportant des examens avec entrées multiples.....	40
d)	Formatage des données avant affichage.....	41
e)	Les pages de l'exposé final.....	42
f)	Utilisation des feuilles de style.....	43
4.	Création d'une aide à l'attention des étudiants.....	45
F.	Résultats	46
1.	Script de création de la base de données.....	46
2.	Fichiers du site	46
III.	Discussion	47
A.	Finalisation du site	47
1.	Installation des fichiers et de la base de données.....	47
2.	Finalisation de la base de données	47
3.	Tests et déboguages.....	48
a)	Tests en local de l'affichage	48
b)	Tests en local de la fonctionnalité	49
c)	Test en ligne	50
B.	Réflexion, critiques	50
1.	Choix du sujet.....	50
2.	Réalisation du site, écriture des pages.....	51
3.	Bilan du site.....	52
a)	Bilan de la base de données.....	52
b)	Limites pour les présentations.....	52
c)	Limites intrinsèques à l'outil	53
d)	Conditions d'utilisation.....	53
e)	Pérennité.....	53
f)	Principes concernant l'amélioration du site.....	54
g)	Suggestions d'amélioration du site	55
C.	Evolution, développement.....	55
1.	Réalisation d'un site complet autour des cas cliniques.....	55
2.	Evolution et développement de la base de données	56
3.	Evolution et développement du site lui-même	56
	Index des figures	58
	Index des fichiers php détaillés et des fonctions	59
	Bibliographie.....	60
	ANNEXES	62
	Guide d'utilisation	63
	Script de création de la base de données	64

Introduction

Depuis plus de trente ans, les étudiants de l'Ecole Nationale d'Alfort exposent en amphithéâtre des cas cliniques concernant les animaux hospitalisés : consultations, suivi, examens complémentaires, conclusions et traitements proposés au patient. L'auditoire est composé de professeurs, d'internes et d'élèves.

Les cas présentés ont une vocation pédagogique :

- pour l'étudiant, qui réalise des recherches bibliographiques et rédige son exposé ;
- pour l'assistance, car ils sont sélectionnés et présentent un intérêt au niveau de la maladie, du suivi, des examens complémentaires.

Toutes ces présentations sont éphémères : aucune sauvegarde à l'attention des étudiants n'a été réalisée depuis leur création bien que ce fût le souhait de nombreux professeurs.

J'ai donc voulu réaliser un outil qui permette l'archivage des présentations de cas cliniques.

Le développement actuel d'Internet, sa démocratisation et les possibilités qu'offrent les différents langages de programmation ont orienté mes recherches dans ce domaine.

Mon objectif a été de créer un outil qui stocke les cas cliniques dans une base de données, et qui aide les étudiants à réaliser leur exposé. Cet avantage devrait en favoriser l'utilisation par rapport à ceux employés actuellement.

Dans un premier temps, je présente la situation actuelle des cas cliniques à l'ENVA, les intervenants et leurs méthodes de travail. Je retranscris la volonté des professeurs en matière de présentation sous forme d'un modèle polyvalent de cas cliniques.

J'expose ensuite la création de l'outil informatique. La méthode MERISE permet de construire la base de données. S'ensuit l'écriture des pages web dynamiques, en langage PHP. Je présente ma méthode et les choix que j'ai fait au niveau sécurité et technique. Cette partie s'adresse aux lecteurs ayant déjà des bases en informatique et programmation.

Dans la dernière partie, j'aborde la finalisation de l'outil et les améliorations qui pourraient lui être apportées. J'expose ses défauts et limites qu'il faudra essayer de corriger. Je présente enfin les perspectives d'évolution et d'utilisation, à la fois au sein de l'ENVA, mais aussi pour d'autres écoles ou universités.

I. Présentation des cas cliniques à l'Ecole Vétérinaire d'Alfort

A. Historique

Les présentations de cas cliniques sont apparues au cours des années 1970. Les étudiants exposaient leurs cas dans le même amphithéâtre que de nos jours, accompagnés de l'animal malade. L'évolution au cours des années a vu :

- l'arrêt de la présence du patient en amphithéâtre ;
- l'amélioration des méthodes d'exposé : transparents, diapositives puis documents informatiques ;
- la diminution du nombre de spectateurs : au début l'amphithéâtre était plein (environ 60 personnes) et de nos jours, seulement une dizaine de personnes assiste aux présentations. Le désintéressement des étudiants pour les cours magistraux, l'horaire tardif de ces présentations (entre 18h et 20h) peuvent expliquer ce phénomène.

B. Insertion au sein de l'enseignement

1. Consultations de médecine et hospitalisation des animaux

Les consultations pour les chiens et les chats se déroulent :

- sur rendez-vous au service de médecine, en salle de consultation dans le bâtiment Brion. Il s'agit des consultations de médecine générale, de cardiologie, d'uro-néphrologie, de neurologie, d'endocrinologie, de cancérologie et de gériatrie ;
- directement au service des urgences, sans rendez-vous, entre 13h et 8h tous les jours en semaine, et sans interruption le week-end et les jours fériés.

Il existe d'autres types de consultations au sein de l'ENVA, mais elles ne nous concernent pas dans le cadre du développement du projet. En effet, seuls les animaux présentés en consultation de médecine ou aux urgences sont admis en hospitalisation, font l'objet d'un suivi et sont ensuite présentés lors des cas cliniques.

Les animaux sont hospitalisés soit parce que leur état clinique le nécessite, soit pour que des examens complémentaires soit réalisés. Dans tous les cas ils sont placés en box individuels. Ils sont répartis dans différentes salles :

- dans la chatterie pour les chats et les chiens de petites tailles (18 box disponibles) ;
- dans le chenil principal pour les chiens (16 box) ;
- dans un chenil dit "semi-contagieux" (4 box) ;
- dans un chenil et une chatterie isolés pour les animaux atteints (ou supposés atteints) d'une maladie contagieuse (6 box).

Remarque : il existe également un chenil pour le service de chirurgie.

Les étudiants des deux dernières années réalisent les consultations aux urgences et en médecine générale : en DCEV-3 et T1-pro durant le premier semestre, en DCEV-2 et DCEV-3 durant le second semestre. Aux urgences, ils sont encadrés par les internes. En consultation de médecine générale, le professeur responsable varie en fonction du jour de la semaine. Les autres consultations de médecine sont réalisées par les étudiants de dernière année.

La plupart des examens complémentaires sont réalisés dans les différentes structures de l'ENVA, interprétés puis validés par les différents professeurs responsables. Les autres examens sont envoyés à des laboratoires extérieurs. Dans tous les cas, les résultats sont reçus au secrétariat des hospitalisations puis insérés dans le dossier du patient.

2. Gestion des animaux hospitalisés

Les soins aux animaux hospitalisés sont réalisés par des étudiants regroupés en binômes ou trinômes, encadrés par des internes, et sous la responsabilité d'un professeur. Les groupes effectuent une rotation chaque semaine.

Les binômes ou trinômes comportent au minimum un étudiant de DCEV-3 et un autre de T1-pro durant le premier semestre, et un étudiant de DCEV-2 et un autre de DCEV-3 durant le second semestre. Chacun se voit attribuer des cas dont il assure le suivi du lundi matin au dimanche soir. Ils effectuent les examens cliniques, les traitements et réalisent ou font réaliser des examens complémentaires, si nécessaire : radiographie, échographie...

La durée d'hospitalisation est gérée par les internes et professeurs, en accord avec les propriétaires.

3. Gestion des dossiers des animaux

Tous les examens cliniques, les résultats et comptes rendus des examens complémentaires sont consignés sur papier et conservés dans un dossier propre à chaque animal, numéroté et archivé. L'école conserve les dossiers des 5 années précédentes ; ceux-ci sont stockés au niveau du secrétariat des consultations.

Depuis deux ans, un logiciel informatique nommé CLOVIS permet une saisie informatique des données du dossier.

4. Préparation des cas cliniques

Durant la semaine de chenil, les étudiants de DCEV-3 ayant assuré les soins choisissent le cas qu'ils vont présenter, en accord avec les internes et le professeur responsable. Il doit présenter un intérêt pédagogique : maladie rare, présence d'une démarche diagnostique logique, examens complémentaires nombreux, évocateurs, etc. Chaque étudiant de DCEV-3 doit préparer un cas.

Il rédige sa présentation à partir d'une recherche bibliographique. Il est en général aidé par son binôme, étudiant de T1-PRO, et par les internes.

C. Etude du déroulement de la présentation des cas cliniques

Durant cette partie, je vais vous présenter le déroulement actuel des cas cliniques, et les analyses que j'ai pu en faire, à partir des commentaires que j'ai recueilli auprès des professeurs, des assistants de clinique, des internes et des étudiants.

La présentation a lieu tous les lundi soirs, dans l'amphithéâtre de médecine.

L'auditoire est composé d'étudiants, d'internes et de professeurs. Généralement, seuls sont présents les étudiants du groupe qui était au chenil la semaine précédente.

Chaque élève dispose d'une vingtaine de minutes pour exposer oralement son cas.

1. Diversité des outils de présentation

L'étudiant présente son cas clinique :

- soit sous forme de transparents dactylographiés, projetés sur un écran par un rétroprojecteur qui lui est fourni ;
- soit sous forme de transparents imprimés à partir d'un document informatique ;
- soit sous forme de présentation PowerPoint® [3], grâce à un projecteur relié à un ordinateur.

Durant les dernières années, avec la démocratisation de l'informatique, on a observé une évolution radicale des outils. Il y a encore 5 ans, on observait une répartition à peu près égale dans les trois formes précédentes. Actuellement, en 2002-2003, la quasi-totalité des cas sont des présentations PowerPoint®. Le cas est alors enregistré sur support disquette, Cd-rom ou clé USB de stockage.

2. Critique des présentations

Lorsque j'ai interrogé les professeurs et l'auditoire en général sur ces outils de présentation, il est ressorti que :

- les présentations dactylographiées sont souvent les moins lisibles : exclusion faite des écritures indéchiffrables, les transparents sont trop denses (les étudiants essaient de tout écrire), la taille du texte est insuffisante ;
- les présentations imprimées à partir de document informatique présentent parfois la même densité que les précédentes. On peut noter un effort de concision et de mise en forme. La lisibilité est meilleure que dans le cas précédent ;

- les présentations PowerPoint® sont, quant à elles, généralement correctement lisibles, et relativement concises. L'esthétique rend ce type de cas agréable à suivre. On rencontre parfois des diapositives trop denses, comme dans les cas précédents.

3. Variations autour d'un plan classique

Il est demandé aux étudiants de respecter un plan de base lors de leurs exposés :

- présentation du cas ;
- historique (commémoratifs, anamnèse) ;
- examen clinique initial à l'école ;
- conclusions de l'examen clinique initial ;
- examens complémentaires réalisés le premier jour ;
- conclusion de la première consultation et traitement mis en place ;
- suivi clinique lors de l'hospitalisation ;
- examens complémentaires lors de l'hospitalisation ;
- conclusion finale : nom de la maladie ou hypothèses les plus probables, traitement et pronostic ;
- discussion rapide sur le cas et sur la maladie à partir d'une recherche bibliographique ;
- présentation de l'aspect financier, à la fois dans sa gestion sur l'école et dans le suivi envisagé ;
- bibliographie.

Ce plan permet d'exposer le cas de façon claire et complète, tout en incluant l'ensemble des examens réalisés.

Il est respecté dans la majorité des cas.

D. Devenir des cas

Après les exposés du lundi soir, aucune présentation n'est archivée. Chaque étudiant repart avec son support de présentation.

Finalement, il ne reste aucune trace des cas cliniques présentés, et ce depuis leur création !

E. Idée de création de l'outil d'aide à la réalisation de cas cliniques et à leur mise en ligne

Le travail réalisé par les étudiants est généralement très intéressant : il repose sur des cas riches en examens cliniques et en examens complémentaires. Les recherches bibliographiques, l'aide apportée par les internes et les professeurs permettent d'obtenir des présentations d'un intérêt pédagogique considérable.

Chaque étudiant aura présenté un cas pendant sa scolarité. Tous les ans, entre 100 et 150 présentations sont réalisées, soit entre 3000 et 4500 depuis les années 70.

Dans ce contexte, on ne peut que regretter l'inexistence d'archivage.

De nombreux professeurs ont souhaité trouver un moyen de conserver, classer et rentabiliser tout ce travail.

Pour ma part, j'ai souhaité également créer un outil qui permette la conservation des cas, en vue de leur consultation par les étudiants.

Depuis quelques années l'informatique a beaucoup progressé, et je me suis intéressé aux possibilités de réalisation dans cette voie.

En complément de l'archivage, il m'a également paru intéressant de développer une aide à la création de présentations informatiques comme celles qui peuvent être réalisées avec PowerPoint[®]. Le but est de faciliter et accélérer le travail de rédaction.

Un tel outil devrait permettre :

- de stocker les cas dans une base de données ;
- de créer facilement des présentations claires, prêtes à être projetées lors de l'exposé oral du lundi soir.

II. Réalisation du projet

A. Etude des cas et création d'un modèle polyvalent de cas cliniques

L'outil final devra être polyvalent, et accepter tous les cas. J'ai donc commencé par analyser les présentations actuelles de cas cliniques, pour créer un modèle de cas.

1. Base de travail

Pour réaliser un modèle polyvalent de cas clinique, il m'a fallu :

- étudier les présentations réalisées par les étudiants lors des exposés oraux du lundi soir ;
- étudier les nouvelles feuilles de cas cliniques que remplissent les étudiants lors des consultations ;
- travailler à partir d'un Cd-rom regroupant l'ensemble des cas de l'année 2002, réalisé par les étudiants de T1-pro canine ;
- travailler avec les professeurs responsables pour connaître leurs volontés concernant la logique des exposés. Il s'agit là d'un travail essentiellement oral.

2. Présentation du modèle de cas cliniques

La présentation actuelle est réalisée de façon linéaire, sous forme de diapositives. Elle satisfait pleinement les différents intervenants. Le modèle que je vous présente ici constitue la trame qui peut être reprise pour toute présentation.

- **Diapositive 1 : présentation du cas**

Nom, prénom de l'étudiant qui présente (et qui s'est occupé de l'animal pendant la semaine)

Numéro de son groupe de clinique

Date du début de sa semaine de permanence au chenil

Nom du professeur responsable lors de cette semaine

Nom de l'animal

Numéro de dossier

Espèce, race

Sexe, stérilisé ou non

- **Diapositive 2 : introduction**

Commémoratifs (historique des maladies de l'animal)

Etat vaccinal, dernières vermifugations

Anamnèse (historique de la maladie en cours)

Traitements en cours, posologie et efficacité

- **Diapositive 3 : examen clinique initial à l'école**

Poids lors de la consultation, évolution depuis 3 mois

Paramètres vitaux : température, couleur des muqueuses, temps de recoloration capillaire, taux de déshydratation, fréquence cardiaque

Examen clinique :

- du système réticulo-hématopoïétique ;
- de l'appareil digestif ;
- de l'appareil respiratoire ;
- de l'appareil cardio-vasculaire ;
- de l'appareil locomoteur ;
- du système nerveux ;
- de l'appareil urinaire ;
- de l'appareil génital ;
- ophtalmologique ;
- O.R.L. ;
- dermatologique ;
- autre.

- **Diapositive 4 : conclusion de l'examen clinique initial et hypothèses diagnostiques**

Tableau des points importants de l'examen clinique et des hypothèses diagnostiques associées.

- **Diapositive 5 : imagerie réalisée lors de la consultation initiale**

Tableau avec l'image, le type d'examen (radiographie, échographie, scanner, IRM), la zone étudiée, et la description.

Conclusion.

- **Diapositive 6 : examens biochimiques sanguins réalisés lors de la consultation initiale**

Tableau avec le nom du paramètre, sa valeur et la norme.

Conclusion.

- **Diapositive 7 : numération-formule sanguine réalisée lors de la consultation initiale**

Tableau avec le nom du paramètre, sa valeur et la norme.

Conclusion.

- **Diapositive 8 : analyse urinaire réalisée lors de la consultation initiale**

Tableau avec le nom du paramètre, sa valeur et la norme.

Conclusion.

- **Diapositive 9 : exploration hormonale réalisée lors de la consultation initiale**

Tableau avec le nom du paramètre, le cas échéant le nom et la posologie de la molécule utilisée lors du test, sa valeur et la norme.

Conclusion.

- **Diapositive 10 : examens cardiologiques réalisés lors de la consultation initiale**

Tableau avec le type d'examen, l'image et le résultat.

Conclusion.

- **Diapositive 11 : examens anatomo-pathologiques réalisés lors de la consultation initiale**

Tableau avec le type d'examen, le type de prélèvement, l'image et le résultat.

Conclusion.

- **Diapositive 12 : examens dermatologiques réalisés lors de la consultation initiale**

Tableau avec le type d'examen, le type de prélèvement, l'image et le résultat.

Conclusion.

- **Diapositive 13 : autres examens réalisés lors de la consultation initiale**

Tableau avec le type d'examen, l'image et le résultat.

Conclusion.

- **Diapositive 14 : conclusions et traitement lors de la consultation initiale**

Conclusions et traitements mis en places lors de la consultation initiale.

- **Diapositive 15 : suivi clinique**

Tableau avec les dates d'hospitalisation et un résumé des points importants des examens cliniques réalisés.

- **Diapositive 16 : imagerie réalisée lors du suivi clinique**

Tableau avec la date, l'image, le type d'examen (radiographie, échographie, scanner, IRM), la zone étudiée, et la description.

Conclusion.

- **Diapositive 17 : examens biochimiques sanguins réalisés lors du suivi clinique**

Tableau avec la date, le nom du paramètre, sa valeur et la norme.

Conclusion.

- **Diapositive 18 : numération-formule sanguine réalisée lors du suivi clinique**

Tableau avec la date, le nom du paramètre, sa valeur et la norme.

Conclusion.

- **Diapositive 19 : analyse urinaire réalisée lors du suivi clinique**

Tableau avec la date, le nom du paramètre, sa valeur et la norme.

Conclusion.

- **Diapositive 20 : exploration hormonale réalisée lors du suivi clinique**

Tableau avec la date, le nom du paramètre, le cas échéant le nom et la posologie de la molécule utilisée lors du test, sa valeur et la norme.

Conclusion.

- **Diapositive 21 : examens cardiologiques réalisés lors du suivi clinique**

Tableau avec la date, le type d'examen, l'image et le résultat.

Conclusion.

- **Diapositive 22 : examens anatomo-pathologiques réalisés lors du suivi clinique**

Tableau avec la date, le type d'examen, le type de prélèvement, l'image et le résultat.

Conclusion.

- **Diapositive 23 : examens dermatologiques réalisés lors du suivi clinique**

Tableau avec la date, le type d'examen, le type de prélèvement, l'image et le résultat.

Conclusion.

- **Diapositive 24 : autres examens réalisés lors du suivi clinique**

Tableau avec la date, le type d'examen, l'image et le résultat.

Conclusion.

- **Diapositive 25 : conclusion finale et traitement**

Nom de la maladie ou hypothèses les plus probables, et traitement établi.

Pronostic de l'évolution de la maladie.

- **Diapositive 26 : discussion**

Discussion sur la maladie et la façon dont a été géré le cas.

- **Diapositive 27 : aspect financier**

Aspect financier du cas lors de sa gestion sur l'école, et budget prévisionnel des traitements envisagés ou mis en place.

- **Diapositive 28 : bibliographie**

Bibliographie.

B. Les bases informatiques : fonctionnement de l'Internet

1. Historique de l'Internet

L'U.S. Air Force, en 1962, en pleine guerre froide, demande à un groupe de chercheurs de réaliser un réseau de communication capable de résister à une attaque nucléaire. Le concept de ce réseau reposait sur la décentralisation. Paul Baran imagina cette première "toile", réseau hybride d'architecture étoilée et maillée dans lequel les informations se déplacent de façon dynamique, en cherchant le chemin le moins encombré. L'Arpanet voit ainsi le jour en 1975. Il est associé au CSNET en 1980 pour former les bases d'Internet.

Le World Wide Web est développé en Suisse en 1991. Le premier logiciel de navigation apparaît cette année-là. L'arrivée du navigateur Netscape en 1994, doté d'une interface graphique spectaculaire (présence de documents multimédias en particulier) entraîne l'explosion de l'Internet [15].

En 1995, il y a environ 20 000 sites à travers le monde, puis 1 million en 1997 et plus de 40 millions en juillet 2003. Il y a eu plus de 600 millions d'internautes en 2003, et on en prévoit 100 millions de plus en 2004 [7].

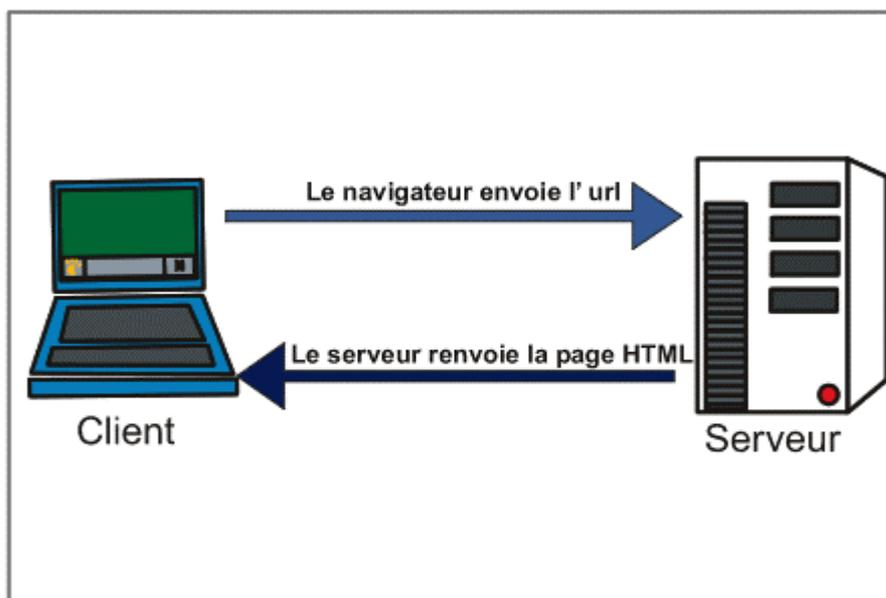
2. Les pages web statiques

Lorsqu'on "surfe" sur Internet, on indique à un navigateur (programme informatique, par exemple Internet Explorer ou Netscape) une adresse URL (Uniform Resource Locator). C'est l'adresse d'un fichier stocké sur un ordinateur relié au réseau. Dans le cas des pages web statiques, l'adresse se termine par ".html" (extension du fichier).

Le serveur est un logiciel permettant d'interpréter des requêtes HTML, et d'y répondre. Parmi eux on trouve Apache (le plus répandu), Microsoft IIS, et Microsoft PWS.

Le navigateur envoie une requête HTML au serveur, qui va vérifier l'existence du fichier dans son arborescence, et renvoie au navigateur la page HTML. Ce dernier interprète les différents langages se trouvant dans le fichier (HTML, javascript, CSS) et affiche la page [11, 15].

Figure 1 : principes d'une page web statique [15]



Les pages web statiques sont très pratiques pour créer des sites composés d'une dizaine de pages, mais possèdent des limites :

- maintenance difficile : obligation de modifier chaque page ;
- impossibilité de personnaliser la page en fonction du visiteur ;
- impossibilité de relier la page à une base de données.

3. Les pages web dynamiques

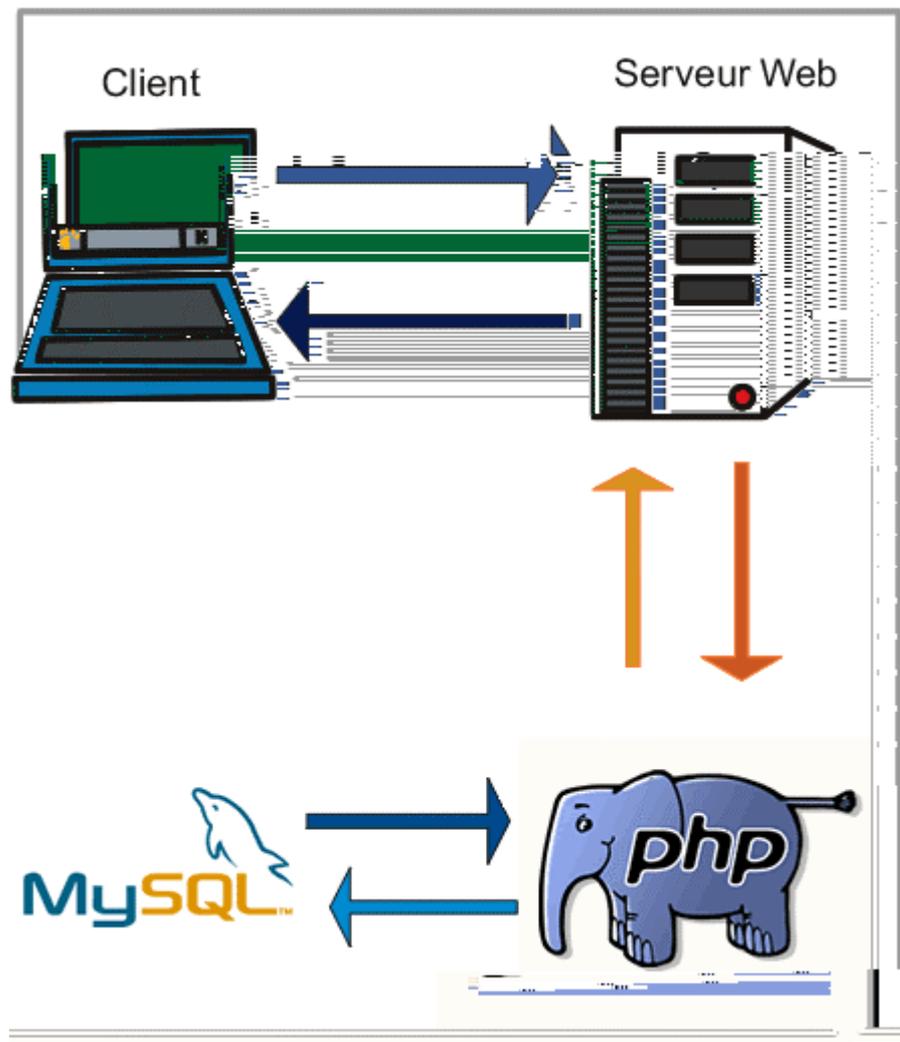
Compte tenu des limites des pages statiques, il a été mis au point des solutions permettant de générer des pages du côté du serveur. Ce sont les langages de scripts (lignes de commandes effectuées par le serveur) :

- le CGI (Common Gateway Interface) est historiquement le premier : il consistait à interpréter des programmes (écrits en Perl ou en C), puis de leur faire renvoyer un contenu compatible avec le protocole HTTP ;
- le langage ASP (Active Server Pages) développé par Microsoft a permis de simplifier l'écriture des scripts en manipulant des objets en VBScript ;
- le PHP (Hypertext Protocol) emploie son propre langage, dérivé du Perl et du C++, est l'équivalent du précédent ;
- le JSP (Java Server Pages) est le plus récent ; il permet d'utiliser la puissance de Java.

Comme précédemment, on indique une adresse URL au navigateur. Le serveur recherche dans son arborescence l'existence du fichier, et vérifie si celui-ci possède une extension connue (par exemple .php, .php3, .php4, .asp, etc.). Un logiciel analyse le code qui lui est destiné (le script contenu entre les balises, par exemple les balises `<?php` et `?>`) et dans le cas où il y ait des requêtes vers une base de données, il les effectue. La base de données renvoie les informations et le script peut les exploiter. Le serveur renvoie ensuite un fichier dépourvu de script au navigateur, une page ne contenant que du HTML. Toute la partie exécutée côté serveur n'apparaît pas : le navigateur ne reçoit que le résultat [11, 15].

Figure 2 : principes d'une page web dynamique

Exemple ici avec serveur, PHP et MySQL [15].



4. Les bases de données et les SGBD

Une base de données est une entité dans laquelle il est possible de stocker des données de façon structurée, et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, des utilisateurs différents.

Une base de données permet de mettre des données à disposition d'utilisateurs pour une consultation, une saisie ou une mise à jour, tout en s'assurant des droits accordés à ces derniers [16, 17, 19].

Afin de pouvoir contrôler les données ainsi que les utilisateurs, il a fallu développer un outil : le Système de Gestion de Bases de Données (SGBD). C'est un ensemble d'applications logicielles permettant de gérer les BDD, c'est-à-dire :

- permettre l'accès aux données de façon simple ;
- autoriser un accès aux informations à de multiples utilisateurs ;
- manipuler les données présentes (insertion, suppression, modification).

Le SGBD peut se décomposer en trois sous-systèmes :

- le système de gestion des fichiers, qui permet le stockage des informations sur le support physique ;
- le SGBD interne, qui gère l'ordonnancement des informations ;
- le SGBD externe, qui représente l'interface utilisateur.

Les principaux SGBD sont les suivants : Borland Paradox, Filemaker, Microsoft SQL Server, Microsoft Access, Oracle, Sybase, MySQL [15].

La communication entre le serveur et le SGBD se fait par l'intermédiaire d'un langage particulier. Le plus fréquent est le SQL (Structured Query Language) [12]. Il permet :

- de créer, modifier ou supprimer des tables (définition de données) ;
- de sélectionner, insérer, modifier, ou supprimer des données dans les tables (manipulation de données) ;
- de définir des permissions au niveau des utilisateurs (contrôle des données).

Les requêtes sont écrites à l'intérieur d'un autre langage (par exemple langage C, PHP...).

C. Choix informatiques pour le projet

1.Choix des langages

Comme nous l'avons vu, le langage HTML est incontournable. Toute l'architecture de la page est basée sur lui. Il faut non seulement écrire une partie des pages web en HTML, mais en plus le comprendre car les requêtes exécutées sur le serveur vont permettre de générer une page HTML.

Pour la partie dynamique, la communication avec la base de données est réalisée par des scripts écrits dans les différents langages vus ci-dessus : PHP, ASP et JSP essentiellement [10].

Le PHP allie de nombreux atouts : simplicité, gratuité et très forte utilisation. On trouve facilement sur Internet de nombreuses aides, ressources, scripts préconstruits, gratuits, écrits dans ce langage. En plus, en commençant le projet, j'avais une expérience du langage PHP, et il est toujours plus simple de s'améliorer dans un langage que d'en apprendre un autre.

Les pages du site seront donc écrites en PHP et HTML.

2.Choix du serveur et du SGBD

Comme nous l'avons vu, les fichiers écrits en PHP, et portant l'extension .php, .php3, ou .php4, sont reconnus par un serveur et exécutés. Les scripts qui communiquent avec la base de données emploient eux aussi un langage, en fonction du SGBD.

De nombreux outils ont été créés autour du PHP, et les hébergeurs ont choisi différentes solutions. La gratuité de certains produits a permis leur large distribution : c'est ainsi que l'association PHP - serveur Apache – SGBD MySQL s'est largement imposée. Parallèlement, des sites de documentation sont apparus, et de nombreux ouvrages ont été publiés. Ceci facilite le travail de création.

Toutes ces raisons m'ont fait choisir l'association PHP – MySQL – Apache.

3.Choix des programmes

a) Programme d'écriture du code et des pages

Pour écrire du code PHP ou HTML, un simple éditeur de texte suffit (par exemple Wordpad[®], fourni avec le système d'exploitation Windows[®]).

Pour le langage HTML, de nombreux logiciels d'aide ont été développés : il s'agit de logiciels WYSIWYG (What You See Is What You Get) où apparaît à l'écran non pas le code, mais l'équivalent de ce qu'affiche finalement le navigateur Internet. Le code est généré par le logiciel. Parmi ces programmes, on peut citer : Word[®], FrontPage[®], Dreamweaver[®], Adobe Golive[®], Namo WebEditor[®].

Ces logiciels permettent également d'écrire directement du code : ils fonctionnent soit en mode WYSIWYG, soit en mode éditeur de code.

Comme précédemment avec les langages, la connaissance et la maîtrise d'un programme conditionnent son utilisation : c'est pour cette raison que j'ai utilisé Dreamweaver Mx[®] [1].

b) Programme de test

Lorsqu'on crée un site web, il faut pouvoir tester les pages au fur et à mesure. Dans le cas de pages dynamiques, il faut non seulement vérifier l'affichage des pages, mais aussi les connexions avec la base de données.

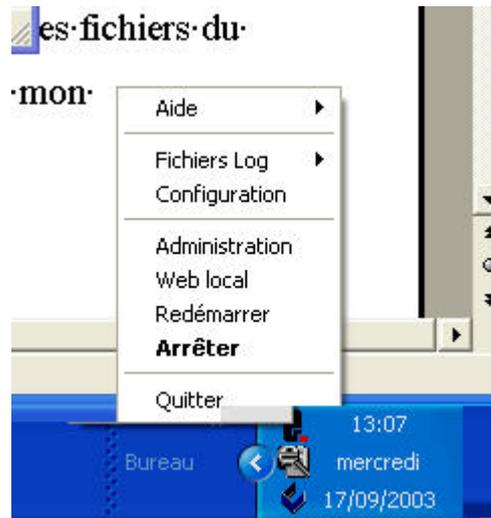
Deux solutions sont alors possibles :

- mettre le site (pages, base de données) en ligne, chez un hébergeur qui peut gérer PHP et MySQL ; ceci est possible mais généralement payant, impose une perte de temps considérable (mise en ligne avant de pouvoir effectuer les tests) et une connexion permanente au web ;
- tout réaliser "en local", c'est-à-dire installer sur son ordinateur l'équivalent de ce que fournit un hébergeur, à savoir Apache, PHP et MySQL ; c'est la solution la plus simple, d'autant que de nombreux logiciels gratuits ont été développés (Easyphp, Phptriad, ...). Pour cette option, le logiciel le plus utilisé, et qui apporte le plus de satisfaction aux utilisateurs est Easyphp ; c'est également mon choix. J'ai utilisé la version 1.6.0.0 du programme [4].

L'installation est classique : choix du répertoire de destination, choix du nom dans le menu "Démarrer". Il suffit ensuite de déposer les fichiers du site dans le dossier "www" du répertoire d'installation d'Easyphp (par exemple sur mon ordinateur : C:\Program Files\EasyPHP\www\).

Une fois lancé, le programme place une icône dans la barre des tâches :

Figure 3 : copie d'écran montrant l'icône Easyphp et ses fonctions



A partir de cette icône, l'utilisateur peut, après avoir démarré EasyPHP :

- consulter son site en Web local : le site s'ouvre dans le navigateur par défaut de l'ordinateur (cliquer sur "Web local") ;
- accéder à sa base de données pour l'administrer, à partir d'une interface phpMyAdmin (cliquer sur "Administration").

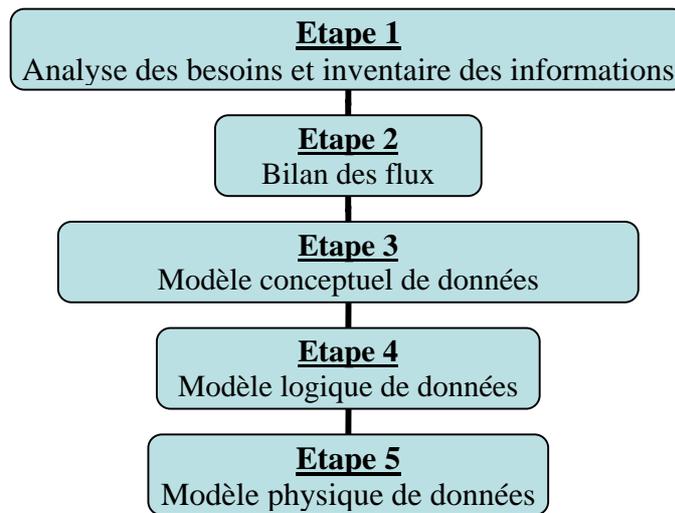
D. Création d'une base de données

1. Principes de création d'une base de données : la méthode MERISE

La création d'une base de données est complexe, car il faut avoir une vision de l'ensemble des données et de leurs interactions. On a donc créé des méthodes de modélisation des données. La plus utilisée en France est la méthode MERISE, conçue en 1978 suite à une demande du ministère de l'industrie [15, 16, 19].

L'analyse des données est le point de départ de la création d'une application mettant en œuvre un SGBD relationnel. La méthode MERISE, basée sur le modèle entité-relation, est un outil simple et efficace pour effectuer cette démarche. Elle utilise la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. Ceci assure une longévité du modèle, car l'agencement des données n'a pas à être constamment remanié [8, 18].

Figure 4 : les différentes étapes de la méthode MERISE



Ces étapes vont être décrites ci-dessous, au fur et à mesure.

2. Analyse des besoins et inventaire des informations

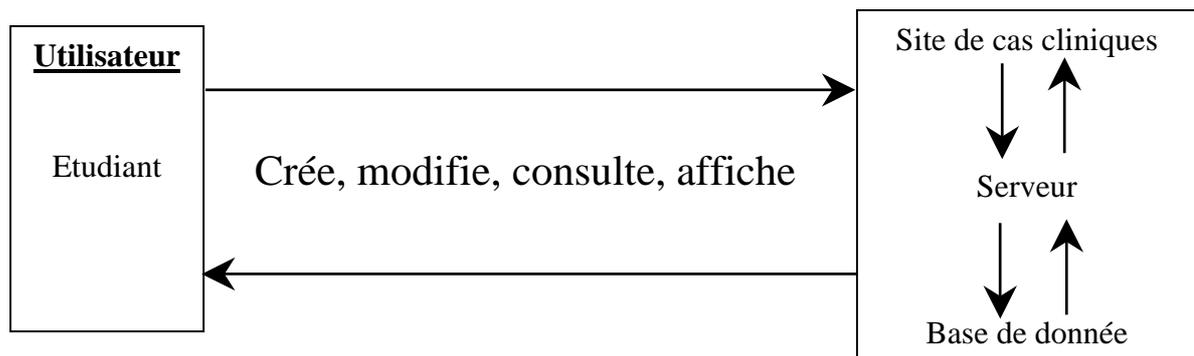
Les besoins ont été analysés en première partie : il faut que la base de données accepte l'ensemble des informations contenues dans les diapositives du modèle de cas cliniques.

Je ne ferais pas ici d'inventaire détaillé.

3. Bilan des flux par rapport à la base de données

Tout est basé autour des cas cliniques : les étudiants se connectent au site pour créer, modifier et consulter leur cas.

Figure 5 : bilan des flux



4. Création du Modèle Conceptuel de Données (MCD)

La base de travail est constituée par

- le modèle polyvalent de cas cliniques présenté en première partie,
- le bilan des flux.

On obtient alors un aperçu de l'ensemble des informations que doit contenir la base de données.

a) Le modèle entité-relation

On va créer des entités, ou classes d'entités, qui sont la représentation d'un élément ayant un rôle dans le système. Chaque entité est composée de propriétés, données élémentaires permettant de la décrire. Chaque élément d'une entité est appelé occurrence. Une entité est généralement créée si elle possède ou possèdera au moins deux occurrences.

Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule occurrence. La définition originale est la suivante : "l'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété pourrait prendre une même valeur" [15].

Une relation (ou association) représente les liens sémantiques qui existent entre plusieurs entités.

La cardinalité permet de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. Elle est composée d'un couple comportant une borne minimale et une maximale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur. La borne minimale, généralement 0 ou 1 décrit le nombre minimal de fois qu'une entité peut participer à une relation. La borne maximale, généralement 1 ou n décrit le nombre maximal de fois qu'une entité peut participer à une relation [8, 18].

b) Ecriture du Modèle Conceptuel de Données (MCD)

Le MCD est la transcription schématique des entités, avec leurs propriétés et des relations qui existent entre elles.

Dans un premier temps, on spécifie les entités. On écrit ensuite l'ensemble des propriétés qui la caractérisent. Enfin, on choisit un identifiant : soit il fait parti des propriétés déjà identifiées, soit on crée une nouvelle propriété. Il faut préférer les identifiants numériques dans les bases de données : ceux-ci facilitent et accélèrent les recherches.

Une fois les entités créées, on écrit les relations qui existent entre elles, avec les cardinalités [8, 18].

Exemple de MCD avec deux entités :

L'entité utilisateur a comme propriétés : nom, prénom, pseudo, et mot de passe. L'identifiant est le pseudo, ce qui implique que deux utilisateurs ne pourront avoir le même pseudo.

L'entité cas clinique a comme propriétés : nom de l'animal, espèce, numéro de dossier. On crée un identifiant numérique : numéro de cas.

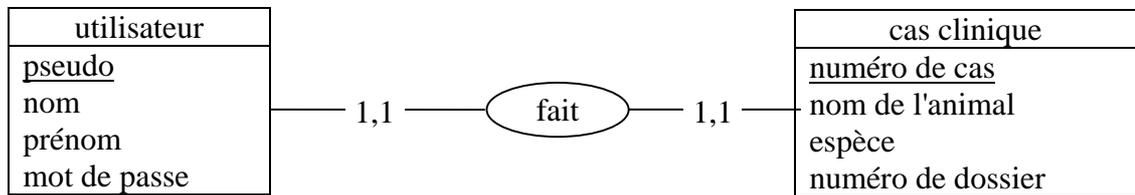
Les relations entre ces deux tables sont les suivantes :

Un utilisateur écrit un et un seul cas.

Un cas est écrit par un et un seul utilisateur.

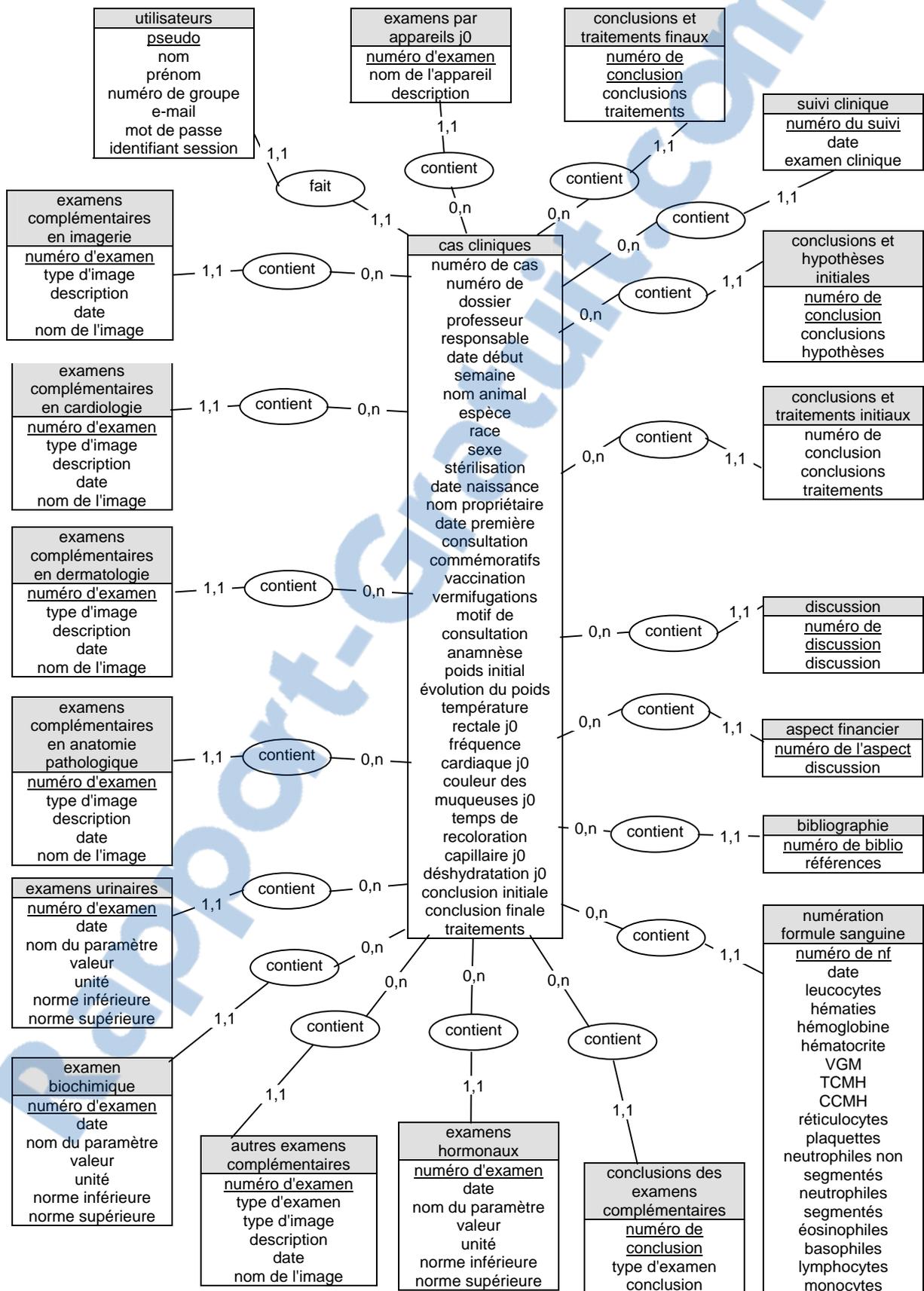
Dans l'écriture du MCD, l'identifiant est souligné.

Figure 6 : exemple de MCD à 2 entités



On réalise cette opération pour l'ensemble des entités et des relations. Je vous présente le MCD final sur la figure 7, à la page suivante.

Figure 7 : MCD final



c) Écriture du Modèle Logique de Données (MLD)

On déduit le MLD directement à partir du MCD.

Le MLD est l'écriture des tables comme elles vont apparaître dans la base de données, avec l'ensemble des champs qu'elles comporteront, en particulier les champs permettant de les relier.

Chaque classe d'entité du modèle conceptuel devient une table dans le modèle logique. Les identifiants de la classe d'entité sont appelés primaires, tandis que les attributs standard deviennent des attributs de la table.

Les liaisons entre les tables sont réalisées par l'insertion de clés étrangères : c'est une nouvelle propriété de l'entité, qui correspond à la clé primaire de la table liée.

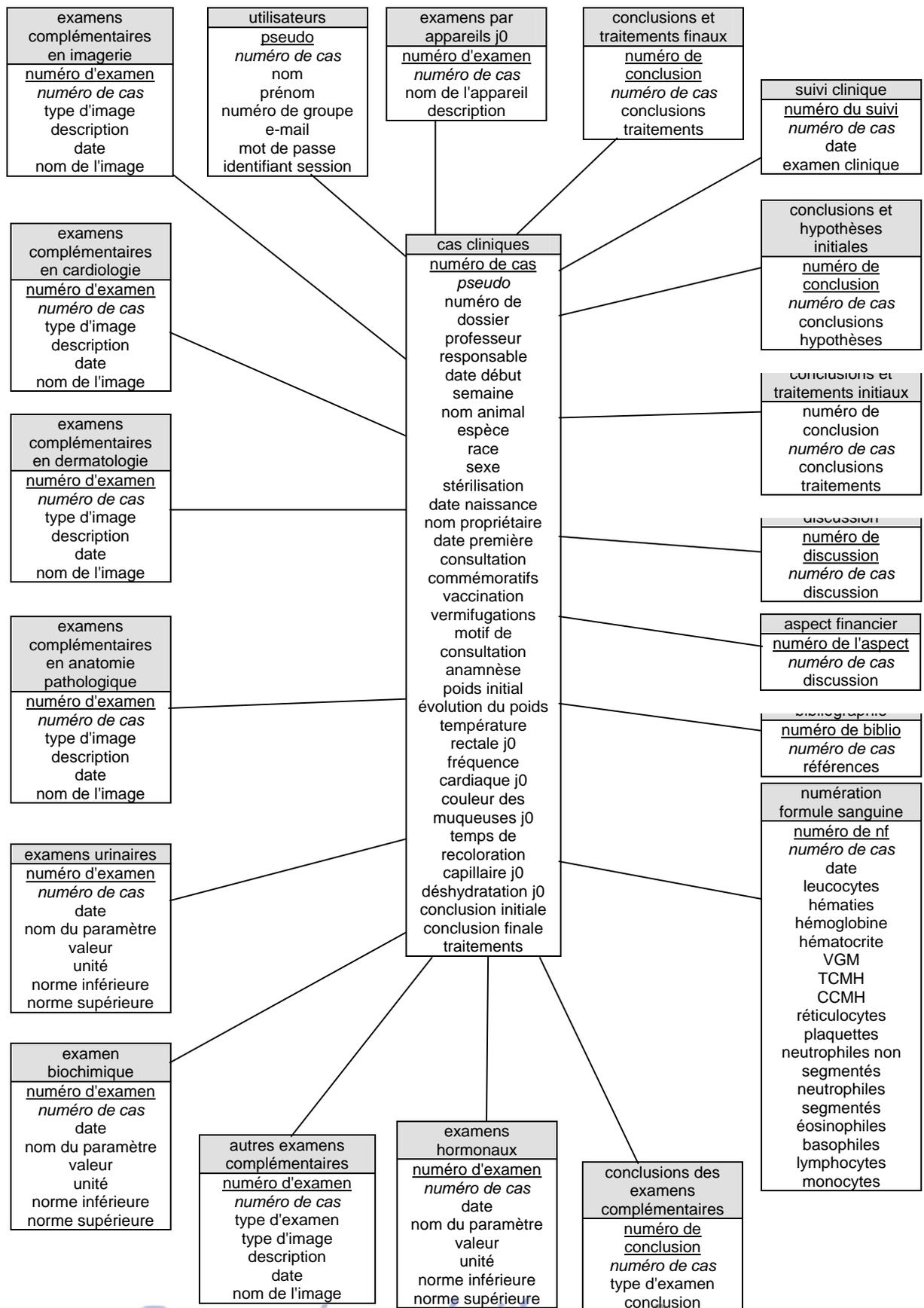
Les cardinalités permettent de définir quelles sont les tables qui hériteront de clés étrangères [8, 18].

On déduit donc le MLD suivant, représenté sur la figure 8.

Dans ce schéma :

- les clés primaires sont soulignées ;
- les clés étrangères sont en italique.

Figure 8 : le MLD



5. Création des tables

La création des tables peut être réalisée en utilisant plusieurs méthodes :

- écriture directe des scripts en langage SQL, qui permet de créer un fichier .PHP à exécuter sur le serveur ;
- création via l'interface phpMyAdmin contenue dans EasyPHP : dans ce cas, on remplit des formulaires et le programme crée les tables. Aucune connaissance en SQL n'est demandée. C'est la solution que j'ai choisie.

Pour cela, on utilise le MLD : le nom de la table est calqué sur celui des entités, et chaque champ correspond à une propriété de l'entité. La clé primaire de la table est celle déjà définie dans le MLD.

Figures 9 à 11 : les différentes étapes de la création d'une base de données avec phpMyAdmin.

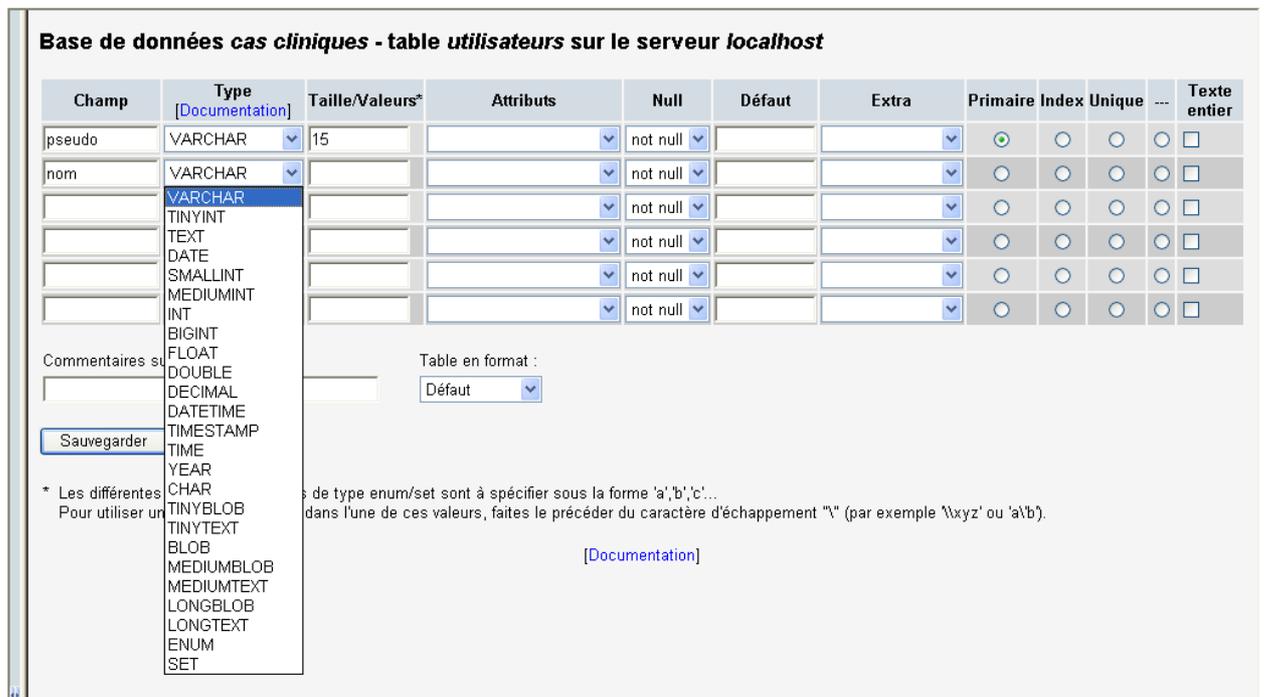
Figure 9 : choix du nom de la base de données avec phpMyAdmin



Figure 10 : choix du nom de la table et du nombre de champs avec phpMyAdmin



Figure 11 : choix du nom des champs et de leurs propriétés avec phpMyAdmin



Pour chaque table, la présence d'un index sur les clés primaires et étrangères permet au SGBD de réaliser des recherches rapidement : lorsqu'on lance une recherche sur des champs indexés, le programme n'a pas à parcourir l'ensemble de la table, mais seulement l'index. Les recherches sont d'autant plus accélérées que la table est volumineuse par rapport à l'index.

L'index est créé automatiquement par phpMyAdmin pour la clé primaire. Pour la clé étrangère, il est possible de le demander lors de la création des champs (cocher "index", voir image 4), ou de le rajouter plus tard au niveau de l'affichage des propriétés de table : sélectionner "créer une clef sur x colonne(s)" puis suivre les instructions.

Il est nécessaire de choisir correctement le type de données en fonction du champ : date, nombre décimal, nombre réel, etc. Il existe 25 types de données [12, 16].

Une fois l'ensemble de la base créé, l'interface phpMyAdmin permet de récupérer les scripts de création au format texte. Ce script est affiché sur une page web. Si on coche "Transmettre", il est proposé de le sauvegarder sous forme de fichier .sql, fichier contenant le script en langage SQL.

Remarque : cette méthode permet aussi de sauvegarder l'ensemble de la table, avec les données. Ceci peut être utile en cas de "déménagement" de la BDD : changement de serveur par exemple.

Figure 12 : récupération du script de création de la base de données avec phpMyAdmin

• Afficher le schéma de la base

- Structure seule
- Structure et données
- Données seulement

Tout sélectionner / Tout désélectionner

- Ajouter des énoncés "drop table"
- Insertions complètes
- Insertions étendues
- Protéger les noms des tables et des champs par des ""
- Transmettre ("bzippé")

Exécuter

E. Création du site Web

Il serait beaucoup trop fastidieux d'expliquer les langages PHP et HTML. La connaissance de ces langages sera donc un pré-requis pour aborder cette partie, dans laquelle j'exposerais mes choix techniques pour le site, et non pas la façon d'écrire le code.

Cette partie s'adresse surtout à ceux qui désirent créer un site similaire, et au futur webmaster du site de l'ENVA. J'y explique ma démarche dans l'écriture et la construction des pages.

Le langage PHP permet d'inclure des lignes de commentaires au sein du code, ignorées par le serveur mais conservées dans le fichier. Chaque ligne doit commencer par //. Ceci m'a permis de placer des explications directement au sein des pages principales du site (fichiers en annexes sur Cd-rom).

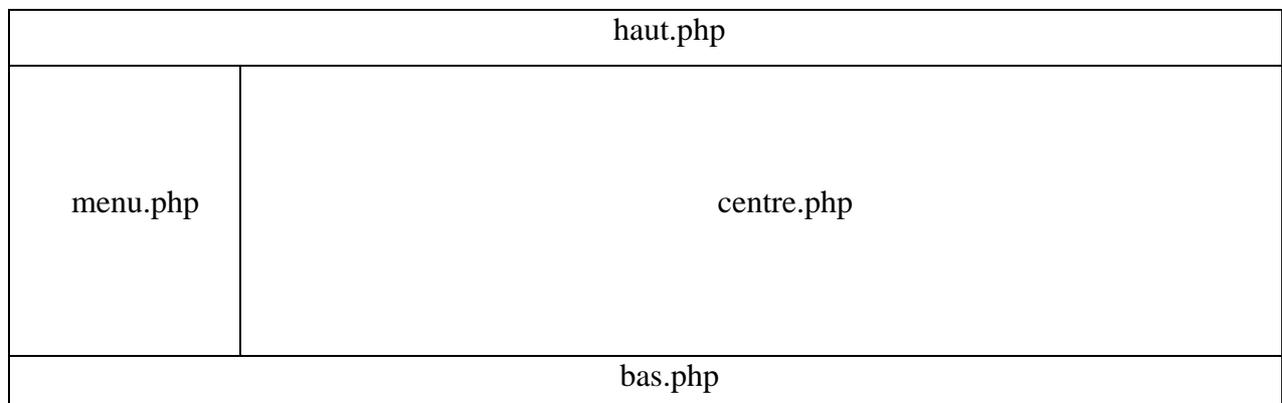
1. Architecture du site : utilisation de pseudo-frames

Je me suis inspiré de nombreux sites utilisant PHP dans des pseudo-frames.

La page `index.php` comprend donc un tableau écrit en HTML dans lequel s'affichent les différentes parties sous forme de script PHP.

Les scripts sont appelés avec la fonction *include* [9].

Figure 13 : les pseudo-frames de la page `index.php`



Les différentes pages s'afficheront dans la partie "centre".

Les différents menus s'afficheront dans la partie "menu".

La partie "haut" est réservée à une utilisation future (présence de logos, de menus complémentaires, par exemple).

Le code d'index.php est le suivant :

Code 1 : fichier index.php

```
<html>
<head>
<title>index du site</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr align="center">
<td colspan="3">
<?
include('haut.php');
// la partie haute de votre site
?>
</td>
</tr>
<tr>
<td width="17%" align="center" valign="top">
<?
include('menu.php');
// la partie menu de votre site
?>
</td>
<td width="3%" align="left" valign="top">&nbsp;&nbsp;&nbsp;</td>
<td width="80%" align="left" valign="top">
<?
include('centre.php');
// la partie centrale
?>
</td>
</tr>
</table>
</body>
</html>
```

Cette méthode permet :

- de changer une zone très facilement, en conservant l'aspect global du site ;
- de modifier la mise en page très facilement, car elle ne dépend que d'un seul fichier.

2.Sécurité du site

a) Affichage des pages et des menus

Lors de la navigation, les liens font toujours appel au fichier index.php, pour lequel on va définir des variables :

```
href="index.php? page=valeurpage&menu=valeurmenu"
```

Par ce moyen, on fait appel au fichier index.php en définissant les variables

- \$page qui a la valeur '*valeurpage*' ;

- \$menu qui a la valeur '*valeurmenu*'.

Ces variables sont utilisées par les scripts contenus dans le fichier index.php. On peut ainsi définir un nombre illimité de variables, en ajoutant le caractère "&" entre chaque.

Il faut interdire l'affichage de n'importe quel lien, ou la réalisation de scripts externes sur le site, par exemple par le lien href=index.php?page=www.script.scriptext.php", qui afficherait la page."www.script.scriptext.php" dans la partie centrale.

Pour cela, on établit une liste de correspondance entre les valeurs de la variable et les fichiers à inclure. Par exemple, si la valeur \$page est "formulaire", alors on fait appel à "formulaire.php". Si la valeur n'est pas définie, on affiche une page par défaut [6, 14].

Le fichier centre.php fait appel à cette méthode. Le fichier menu.php fait appel à une méthode similaire. Leurs codes sont commentés. Je vous présente ici le code partiel de centre.php.

Code 2 : code partiel du fichier centre.php

```
<?
//on teste la valeur de la variable $page
switch ($page)
{
//si elle est égale à formulaire
case 'formulaire':
//on fait appel au fichier formulaire.php
include('formulaire.php');
//et on arrête de tester la variable
break;
//sinon on continue les tests...
case 'ajouteruser':
include('ajouteruser.php');
break;

...

//si la variable n'a aucune de ces valeurs
default:
//on fait appel au fichier par défaut, qui ici renvoie vers la page d'identification
include('entree.php');
break;
}
?>
```

Cette méthode donne de bons résultats au niveau de la sécurité des pages, mais nécessite parfois, et c'est le cas ici, de définir un grand nombre de valeurs (70 dans le fichier centre.php).

b) Identifiant de session

Le site est personnalisé : chaque utilisateur inscrit ne crée, modifie et consulte que son propre cas. Les utilisateurs non inscrits n'ont aucun droit.

Il est nécessaire, lors de la navigation, d'avoir un suivi de l'identité de l'utilisateur pendant sa session. Il est également nécessaire que cette identité soit cachée, pour éviter tout piratage [6, 13].

J'ai décidé de créer un identifiant alphanumérique aléatoire unique, généré lors de l'inscription et renouvelé lors de l'entrée sur le site. Il est stocké dans la table utilisateur, dans le champ ID_MB.

La navigation entre les pages inclut donc cette variable identification de session :
`href="index.php?id=identifiant"`

Chaque script d'affichage de page commence par une vérification de la validité de l'identifiant, grâce au fichier testconnect.php. Si l'identifiant n'est pas défini, ou n'existe pas dans la base, on renvoie l'utilisateur vers la page d'identification, et on arrête le script.

Code 3 : fichier testconnect.php

```
<?
// on fait appel au fichier de configuration d'accès au serveur et à la base de données
require("conf.php");

// connection à la base de données
$db_connect = mysql_connect($sqlhostname,$login,$password);
$base_selection = mysql_select_db($base);
if(!$db_connect) {echo "Connexion impossible à la base de données <b>$sql_bdd</b> sur le serveur
<b>$sql_server</b><br>Vérifiez les paramètres du fichier conf.php3"; exit;}

// sélection de l'enregistrement contenant l'id en cours
$query = "SELECT * FROM utilisateur WHERE ID_MB = '$id'";
$req = mysql_query($query);
$row = mysql_fetch_array($req);
//on récupère les données
$pseudo = $row['PSEUDO'];
$num_cas = $row['NUMERO_DE_CAS'];
$res = mysql_num_rows($req);

// si la requête a échoué, on affiche les erreurs
if (!$req)
{ echo "<B>Error ".mysql_errno().":</B> ".mysql_error()."";
exit; }

// si l'id n'existe pas
if($res == 0)
{
    // on affiche le message suivant
    echo "<p align='center'> '$id' vous devez être enregistré; pour faire cette
op&eacute;ration</p><p align='center'><a href='index.php?page=entree'>Aller &agrave;
l'identification</a></p>";
    exit;
}
?>
```

Ce script permet également de définir la variable "numéro de cas clinique" (\$num_cas), qui sert dans de nombreux scripts de recherche. Dans les tables, chaque entité est définie par sa clé primaire et par une clé étrangère qui est le numéro de cas.

c) Navigation entre les pages

La navigation entre les différentes pages se fera donc avec des liens définissant les 3 variables suivantes : menu, page, identifiant de session. Ces liens sont du type :

```
href="index.php?menu=menu&page=page&id=identifiant"
```

d) Configuration de la connexion avec la base de données

La connexion avec les bases de données nécessite de définir plusieurs variables avant la réalisation des scripts :

- nom de l'hôte (serveur) ;
- identifiant de connexion ;
- mot de passe ;
- nom de la base de données.

Ces données sont stockées dans un fichier nommé conf.php, qui est inclus (fonction "include") au début de tous les fichiers comportant un script nécessitant une connexion avec le serveur et la base de données.

Ceci permet si nécessaire, de changer ces variables en ne modifiant qu'un seul fichier, dans les cas de changement d'hébergement, de serveur, de nom de base ou de paramètres de connexion.

Code 4 : fichier conf.php

```
<?
// variables de connexion au serveur et à la base de données
// nom du serveur
$sqlhostname = "localhost";
// identifiant
$login = "root";
// mot de passe
$password = "";
// nom de la base de données
$base = "these";
?>
```

e) Bilan sur la sécurité du site

La sécurité repose donc sur les points suivants :

- obligation d'être un membre inscrit, donc enregistré dans la base de données, pour accéder aux pages de cas clinique ;
- impossibilité de faire exécuter des scripts externes lors de l'affichage des pages.

La sécurité est évidemment basique, car des pirates expérimentés peuvent modifier n'importe quel site, mais elle semble adaptée au faible risque ici présent. Elle permet surtout d'éviter les bugs d'affichage de pages, de perte de session, et de modification des cas par des utilisateurs non autorisés.

3. Etude de choix techniques

a) Valeurs des listes dans les formulaires

J'ai voulu profiter des avantages des tables pour créer des menus de façon dynamique dans les formulaires. La liste des choix correspond aux valeurs d'une table : ceci en facilite la création et la modification. En effet, celles-ci seront effectuées sans modifier les fichiers du site, mais simplement en changeant les valeurs d'une table dans la base de données [5, 13].

Les tables sont les suivantes :

- param_prof : permet d'afficher le choix des professeurs dans la page de présentation des cas ;
- param_bioch et param_horm, param_urine : permettent d'afficher le nom et l'unité des paramètres biochimiques, hormonaux et urinaires dans les pages d'examens complémentaires correspondantes ;
- param_appareil : permet d'afficher le choix de l'appareil lors de l'examen clinique initial (par exemple, appareil locomoteur, appareil cardio-vasculaire) ;
- param_image, param_anapath, param_cardio, param_dermato : permettent d'afficher les listes de choix des examens complémentaires en imagerie, anatomie pathologique, cardiologie et dermatologie. Par exemple, dans la table param_image, on trouvera les valeurs : radiographie, échographie, scanner, IRM.

b) Les pages comportant des formulaires simples

Ces pages n'affichent que le formulaire. Celui-ci permet à la fois la saisie des données, et l'affichage par défaut des données saisies en vue de modifications.

Pour cela, on associe :

- des requêtes PHP qui vont chercher, si elles existent, les données dans la base ;
- un formulaire simple, écrit en HTML, dans lequel on affiche les données à l'aide de scripts [5].

Chaque formulaire appelle un fichier PHP d'ajout des données dans la base, avec l'identifiant de session. Par exemple, la validation du formulaire "introduction" fait appel à la fonction suivante : `action="addintroduction.php?id=<? echo $id ?>`.

Il s'agit des pages :

- introduction ;
- anamnèse et commémoratifs ;
- début de l'examen clinique initial ;
- numération-formule J0 ;
- aspect financier ;
- discussion ;
- l'ensemble des conclusions des examens complémentaires est également de ce type.

Seul le code de `introduction.php` est commenté (fichiers en annexes sur Cd-rom).

c) Les pages comportant des examens avec entrées multiples

Dans ce cas, les pages comportent une partie d'affichage des données déjà saisies et une partie formulaire.

Figure 14 : page biochJ0

Examens biochimiques - premier jour		
hémoglobine (g/l)	12 (norme : 10 - 15)	cliquer ici pour effacer
acide urique (mg/l)	15,2 (norme : 12,2 - 0)	cliquer ici pour effacer
acide urique (mg/l)	2,2 (norme : 0 - 0)	cliquer ici pour effacer
acide urique (mg/l)	12 (norme : 12 - 0)	cliquer ici pour effacer
acide urique (mg/l)	1 (norme : 1 - 0)	cliquer ici pour effacer

Paramètre : Valeur :
 norme inférieure : norme supérieure :

} Affichage des données déjà validées
 } include : voirbiochJ0.php
 } Formulaire

Le code de biochJ0.php permettant l'affichage de la partie formulaire est classique, en majeure partie en HTML. On utilise un script pour afficher la liste des paramètres : ceux-ci sont tirés de la table "param_bioch".

Les lignes et colonnes du tableau sont générées par le fichier voirbiochJ0.php, dont voici le code d'affichage :

Les pages conçues sur ce modèle sont :

- examens complémentaires (sauf numération formule J0),
- conclusion et traitements J0,
- conclusion générale,
- suivi clinique,
- bibliographie.

d) Formatage des données avant affichage

Les données stockées dans la base nécessitent parfois d'être formatées [5].

Les dates sont stockées sous forme année/mois/jour (AAAA/MM/JJ).

J'ai donc créé la fonction datefr (\$date) : à partir d'une date de la table au format AAAA/MM/JJ, elle renvoie une date formatée pour l'affichage français : JJ/MM/AAAA.

Code 5 : fonction datefr

```
<?
function datefr($datefr)
{
$datea=substr($datefr, 0, 4);
$datem=substr($datefr, 5, 2);
$datej=substr($datefr, 8, 2);
$datejm=$datej .' ' .$datem .' ' .$datea ;
$datefr=$datejm;
global $date_fr, $date_en;
$date_fr=$datej .$datem .$datea;
$date_en=$datea .$datem .$datej;
return($datefr);
}
?>
```

Dans les tables de la base de données, dans les champs de type TEXT ou BLOB, les retours à la ligne sont codés avec une balise <nl>, alors qu'en HTML on utilise
. J'ai donc fait appel à la fonction php "nl2br" pour intervertir ces balises avant l'affichage [17].

Le code de la fonction est le suivant :

Code 6 fonction nl2br

```
<?
$texteformaté = nl2br($textebrut);
?>
```

e) Les pages de l'exposé final

Les pages d'affichage du cas sont assez simples : on va extraire les données de différentes tables et on les insère sous forme de texte ou de tableau.

f) Utilisation des feuilles de style

Le principe des feuilles de style est de définir des caractéristiques de mise en forme applicables à des groupes d'éléments.

Les feuilles de styles permettent :

- d'avoir une présentation homogène sur tout un site ;
- de pouvoir changer l'aspect du site entier en ne modifiant que quelques lignes de code, dans un nombre restreint de fichiers ;
- d'obtenir une plus grande lisibilité du code HTML,
- de charger les pages plus rapidement car le code n'est pas surchargé.

C'est un avantage dans l'évolutivité du site [11].

J'ai défini trois feuilles de styles.

Sur les images suivantes, je détaille les zones où interviennent les différents styles définis.

Figure 15 : feuille de style des menus

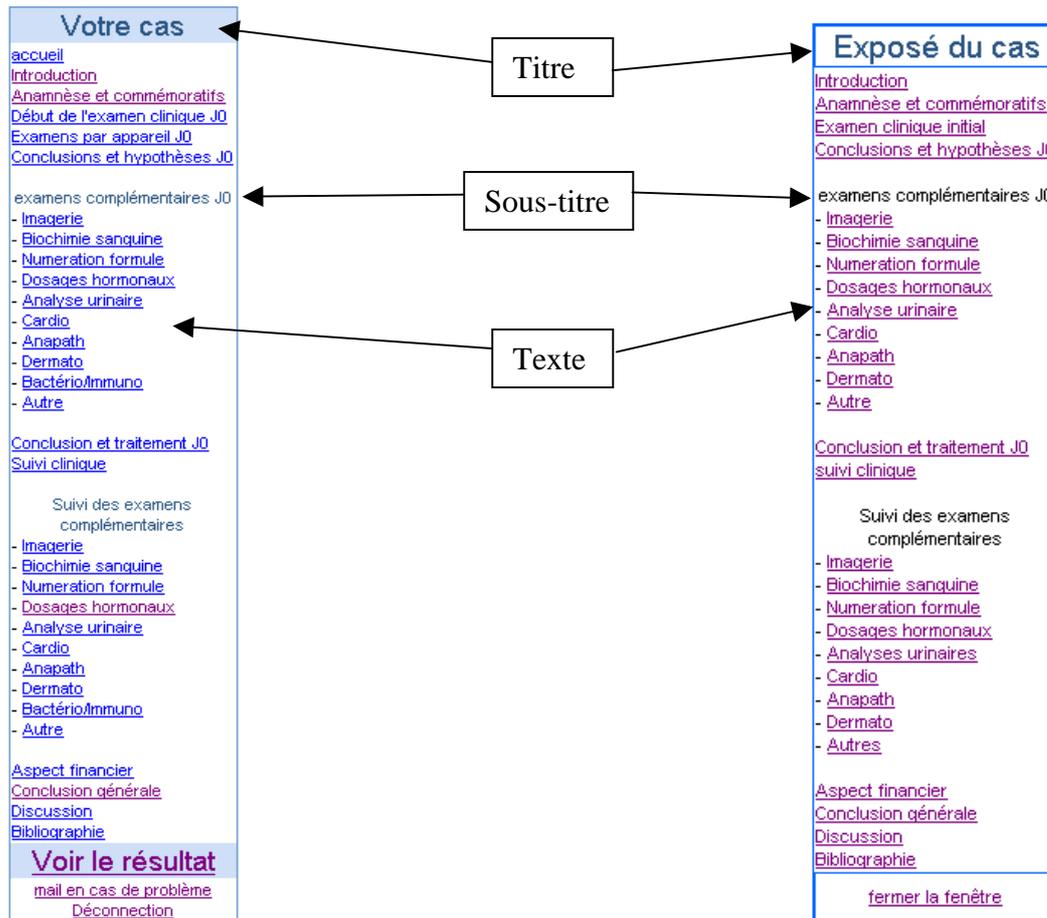
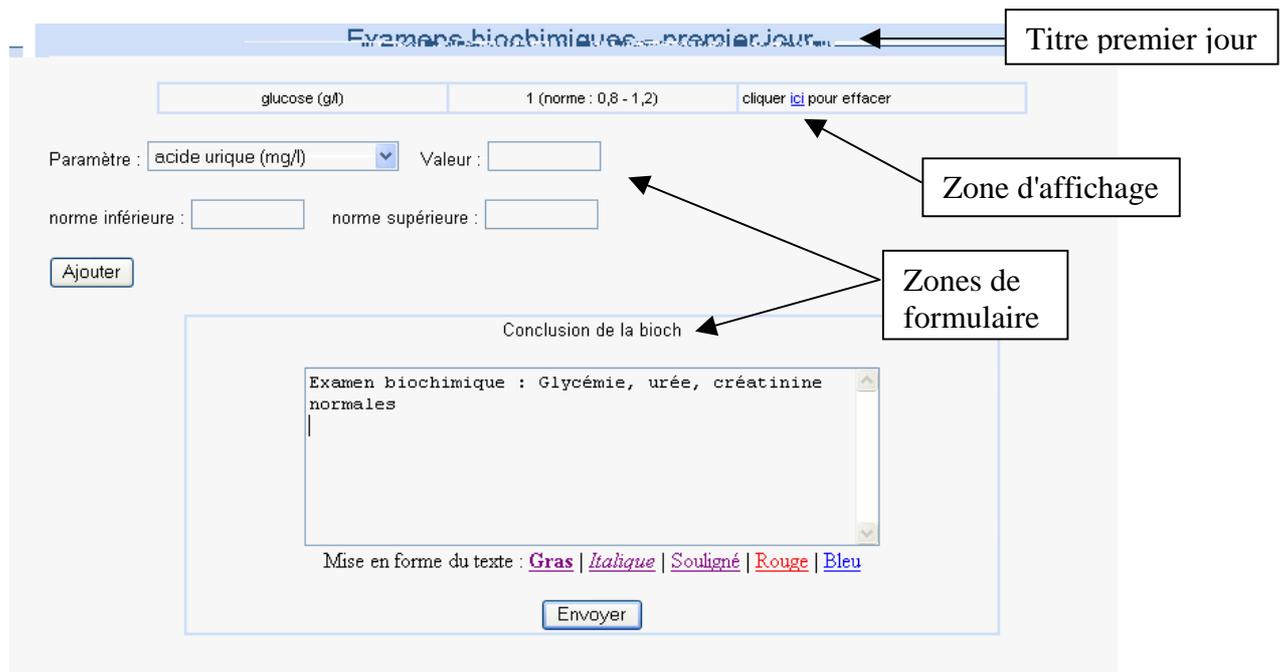
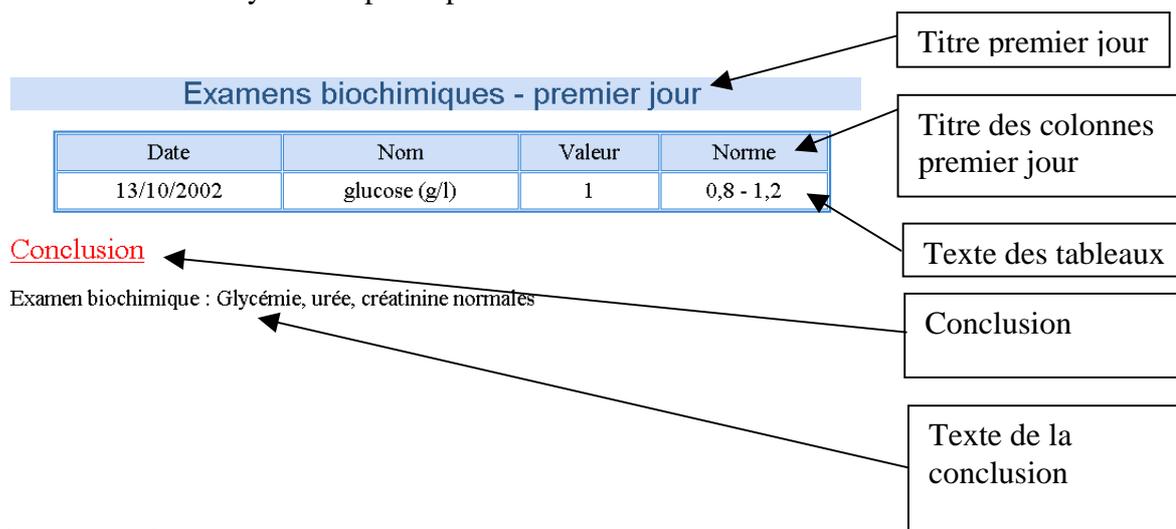


Figure 16 : feuille de style de la partie formulaires



Il existe également un style pour les titres dans le suivi du cas.

Figure 17 : feuille de style de la partie présentation



Il existe également :

- un style pour les titres pour le suivi du cas,
- un style pour les titres des colonnes pour le suivi du cas,
- un style pour les liens,
- un style pour du texte plus petit dans les tableaux.

4. Création d'une aide à l'attention des étudiants

La formation des étudiants vétérinaires aux outils informatiques est très hétérogène et parfois très sommaire. Tous n'utilisent pas Internet couramment, même si cet outil se démocratise à grande vitesse.

Il m'a semblé important de créer une feuille d'aide, qui serait distribuée aux étudiants pendant leur semaine de chenil, afin de leur expliquer le fonctionnement du site.

J'ai décidé de ne pas évoquer ces parties les plus simples, qui concernent la navigation sur le web :

- utilisation du menu pour naviguer sur le site ;
- remplissage des formulaires.

Ce texte explique comment se servir des parties du site les moins intuitives, c'est à dire celles qui à priori pouvaient apporter des problèmes :

- l'envoi d'images à la bonne taille et au bon format ;

-l'existence de menus similaires pour les examens complémentaires du premier jour et des jours suivants.

Cette aide se trouve en annexes.

F. Résultats

1.Script de création de la base de données

La base de données comporte vingt-neuf tables. Leurs structures sont incluses dans le script de création de la base de données. Le script se trouve en annexes.

Celui-ci permet de créer l'ensemble de la base de données sur le serveur d'hébergement.

Il correspond au fichier nommé base.sql, qui se trouve en annexes sur Cd-rom.

2.Fichiers du site

163 fichiers .php et 3 feuilles de style permettent donc de générer 61 pages web dynamiques. L'ensemble des codes se trouve en annexes sur Cd-rom.

J'ai fourni au service informatique de l'ENVA l'ensemble de ces fichiers sur Cd-rom, accompagnés du script de création de la base de données (fichier base.sql).

III. Discussion

A. Finalisation du site

La création des fichiers et de la base de données est maintenant réalisée, mais il reste encore des étapes avant l'utilisation finale du site. Nous allons les exposer ici.

1.Installation des fichiers et de la base de données

L'hébergement se fera sur les serveurs de l'ENVA. J'ai manqué de temps pour effectuer cette phase moi-même. Elle sera réalisée ultérieurement, en collaboration avec le service informatique et multimédia.

Cette étape est simplifiée par l'existence du script de création de la base de données (base.sql) et par la présence de l'ensemble des fichiers sur support CD-Rom.

Le site aura alors une adresse accessible par l'Internet et les étudiants, du type <http://cascliniques-enva.fr>

La base de données comportant des noms d'étudiants, elle devra être déclarée à la CNIL.

Il faudra résoudre le problème de la propriété intellectuelle des cas et des images : appartiennent-ils à l'ENVA ou aux étudiants qui les ont réalisés ? Actuellement, je n'ai pas trouvé de réponse, et je pense que l'aide d'un juriste sera indispensable.

2.Finalisation de la base de données

Comme nous l'avons vu précédemment, certaines tables servent de support aux menus déroulants des formulaires (noms des professeurs, noms et unité des paramètres...).

Il est nécessaire d'implémenter toutes les tables qui le nécessitent.

Cette étape doit être réalisée en collaboration avec les professeurs. Les données ainsi validées ne sont pas définitives, et je pense qu'il sera nécessaire de les mettre à jour régulièrement, en fonction des commentaires des utilisateurs.

3. Tests et déboguages

a) Tests en local de l'affichage

Comme nous l'avons vu, l'installation sur l'ordinateur de travail du programme Easyphp permet de tester le site "en local".

La création de la base de données doit être réalisée dans un premier temps, car elle est utilisée par les tests.

On écrit ensuite les fichiers, que l'on essaye avec EasyPHP.

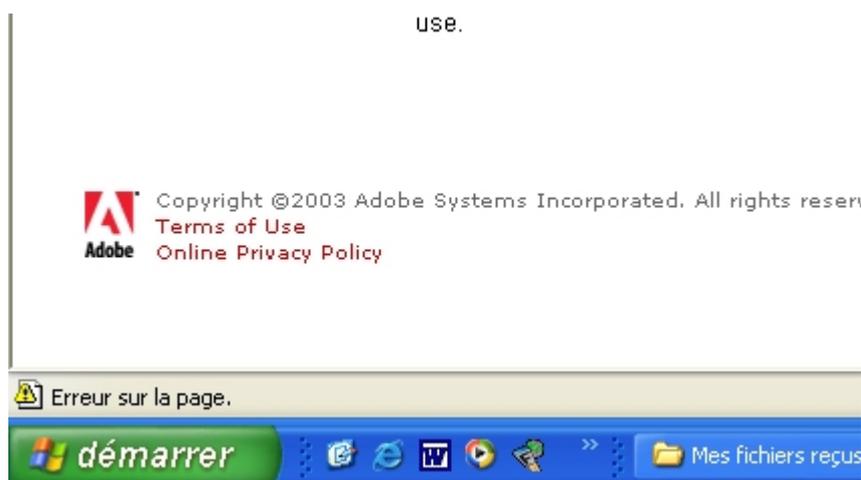
J'ai utilisé un seul navigateur pour ces tests : Internet Explorer 6.0 de Microsoft [2]. C'est le plus répandu actuellement à travers le monde [7]. Ma résolution d'écran était 1024x768 pixels et mon système d'exploitation Windows XP®.

En cas d'erreur :

- soit la page ne s'affiche pas du tout, car elle est refusée par le navigateur ;
- soit la page s'affiche mais avec des erreurs de mise en page, des parties manquantes, etc. ;
- soit la page s'affiche bien mais il existe une erreur mineure qui ne gêne pas le navigateur : celui-ci l'indique grâce à une icône dans la barre d'état (cf. image).

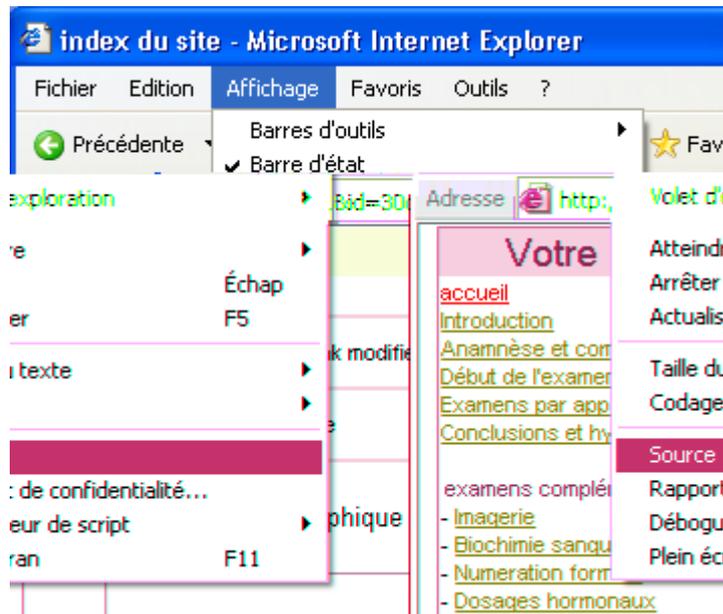
Figure 18 : icône d'erreur sur la page, avec Internet Explorer 6.0

L'icône apparaît à gauche dans la barre d'état (copie d'écran partielle).



Le navigateur permet d'afficher le code source de la page qu'il affiche. Pour Internet Explorer, il faut cliquer sur Affichage → source : voir l'image.

Figure 19 : affichage du code source d'une page web sous Internet Explorer 6.0
(copie d'écran partielle)



En étudiant ce code, on peut comprendre l'erreur et remonter à son origine. Dans mon cas, les erreurs se trouvaient le plus souvent dans les scripts PHP.

Les erreurs les plus fréquentes sont les fautes typographiques (en particulier les inversions de lettres) et les erreurs dans les requêtes. Les erreurs de code sont ainsi décelées et directement corrigées.

b) Tests en local de la fonctionnalité

J'ai récupéré quelques présentations PowerPoint® des cas cliniques réalisés par les étudiants pendant l'année 2002-2003, avec lesquelles j'ai pu tester les fonctionnalités du site et de la base de données. Toutes les présentations ne respectent pas le plan classique, et certaines ont dû être exclues des essais.

Cette étape permet de vérifier que les formulaires et les scripts sont correctement reliés avec la base de données.

c) Test en ligne

A ce jour, je n'ai pas eu le temps de réaliser des tests en ligne.

Cette phase va permettre :

- de corriger d'éventuelles erreurs de code qui m'auraient échappé ;
- d'adapter le site aux différents navigateurs Internet (Internet Explorer, Netscape, Mozilla...) ;
- d'adapter la mise en page aux différentes résolutions d'écran ;
- de vérifier la fonctionnalité du site avec des cas cliniques.

Cette étape est la dernière à réaliser avant la mise en fonction du site pour les étudiants.

Elle est absolument nécessaire. En effet, si on demande à des étudiants de rentrer leurs cas par le site, et que celui-ci ne fonctionne pas ou mal, ils risquent de n'avoir aucune présentation le moment venu.

Les cas cliniques sont souvent réalisés le week-end pour le lundi soir, après une semaine fatigante ; l'outil utilisé doit donc être parfaitement fonctionnel.

B. Réflexion, critiques

1.Choix du sujet

La création de cet outil me tient particulièrement à cœur, car j'ai assisté à de nombreuses présentations de cas dont il ne reste aucune trace aujourd'hui, et qui pourtant étaient de très grande qualité. Depuis les années 70, on a donc assisté à une perte considérable de données pédagogiques.

Actuellement, à ma connaissance, aucun logiciel ni site web ne permet de constituer une base de données de présentations telle que je l'ai conçue. Il existe des sites exposant des cas cliniques, mais aucun n'est interactif ou ne permet de réaliser des recherches thématiques. Ils ne semblent pas comporter de base de données, mais uniquement des cas écrits en HTML.

Même le logiciel Clovis, utilisé par l'école pour informatiser les dossiers des patients, est limité dans cette utilisation. C'est un outil d'archivage de l'ensemble des cas, et il n'existe pas de sélection des dossiers les plus intéressants au niveau pédagogique.

Au début, je voulais réaliser un site complet autour de la base de données : entrée des cas par les étudiants, possibilité de présentation en amphithéâtre, relecture et validation par les professeurs, consultation de la base de données avec recherche par mots clés. Toutefois le temps de développement aurait été considérablement augmenté, ce qui est incompatible avec les limites imposées par la réforme des études vétérinaires (obligation de réaliser sa thèse l'année qui suit la fin de sa T1-Pro). J'ai donc limité mon sujet à un site fonctionnel pour l'exposé, et qui présente déjà l'intérêt de constituer la base de données en vue d'une exploitation future.

Je pense avoir réalisé un outil ayant une vocation pédagogique importante, en particulier si la base de données est exploitée avec un site de consultation des cas. J'ai proposé au service informatique mon aide pour le développer : après avoir passé de longs moments pour réaliser ma thèse, j'ai l'avantage d'avoir une vision d'ensemble de la BDD, et je pense pouvoir concevoir le site avant juin 2004.

2.Réalisation du site, écriture des pages

Lorsque je me suis lancé dans la réalisation d'un site dynamique, basé sur le PHP, je connaissais mes lacunes dans ce domaine. Dans un premier temps, j'ai cherché des logiciels qui permettent le développement WYSIWYG. Malheureusement, ces logiciels sont relativement rares et très onéreux (par exemple WinDev[®], logiciel français coûte 1600 €). De plus leur utilisation nécessite un apprentissage sans doute aussi long que celui que j'ai passé à étudier la programmation en PHP. Ayant abandonné cette voie, je me suis tourné vers Internet et la lecture d'ouvrages traitant de ce langage. A force de lire des scripts et des lignes de code, j'ai réussi à comprendre comment écrire des fichiers et des requêtes. Cette phase a duré environ 6 mois. Elle a considérablement ralenti la réalisation du site.

Ma formation limitée expliquera les imperfections dans le code des fichiers, imperfections qui m'échappent en grande partie.

Il existe toujours de nombreuses façons d'arriver à ses fins lorsqu'on crée un outil informatique, et celle que j'ai utilisé n'est peut être pas la plus efficace, en termes de vitesse de scripts ou de sécurité, mais elle est pleinement fonctionnelle.

3. Bilan du site

a) Bilan de la base de données

Je pense avoir conçu une base de données correctement architecturée. Le critère essentiel est l'absence de redondances des données dans les différentes tables, ce qui est le cas ici. Les données, en dehors des clés étrangères, ne se trouvent écrites que dans une seule table. C'est essentiel dans le cadre d'une BDD relationnelle. Ceci facilite les requêtes et évite les incohérences.

C'est une partie cruciale du site, car l'ensemble des requêtes est basé sur cette architecture. La modification structurelle de tables obligerait à corriger de nombreux fichiers PHP. Par contre, l'ajout de tables ou de données dans les tables est une manipulation plus simple.

Avec l'usage, il sera peut-être nécessaire de changer les propriétés de certaines données dans les tables. Ceci peut être réalisé par l'administration phpMyAdmin, et, dans la majorité des cas, ne modifie pas les données existantes.

Par exemple, dans la table param_anapath, l'entité NOM_ANAPATH est une chaîne de maximum 20 caractères (varchar (20)). Si on a besoin de mettre des noms plus longs, il suffira de modifier cette propriété en précisant la nouvelle valeur (par exemple varchar (30)). Il est à noter que toute manipulation réduisant la longueur de texte ou de chaîne de caractères pourra modifier des données de façon irréversible.

b) Limites pour les présentations

La présence d'une base de données, et du formulaire tel qu'il a été construit oblige à une certaine rigidité dans le plan. Ce site a été conçu pour des cas "classiques". Comme j'ai pu le constater lors des tests, certains types de présentations ne peuvent convenir. Par exemple :

- les comparaisons de suivis pour plusieurs animaux, car le site ne permet le suivi que d'un patient ;
- les suivis d'animaux diabétiques, pour lesquels des courbes de glycémies ont été réalisées. Dans ce cas, il faudra développer des pages supplémentaires, ce qui est concevable car le PHP permet de générer des diagrammes et des courbes.

c) Limites intrinsèques à l'outil

Les données seront présentes sur un serveur de l'ENVA, et accessible par Internet. Lors des présentations cliniques, le lundi soir, les étudiants devront donc pouvoir se connecter au site. Ceci implique que l'ensemble du réseau fonctionne : connexion sortante de l'ENVA, et connexion aux serveurs. Les pannes sont rares mais généralement imprévisibles, et elles entraîneraient l'impossibilité de présenter les cas.

Le fonctionnement de l'ordinateur relié au rétroprojecteur est également indispensable comme il l'est déjà avec les présentations PowerPoint® actuelles.

d) Conditions d'utilisation

Cet outil peut être utilisé de différentes façons.

Compte tenu de tout ce nous venons de voir, je propose les solutions suivantes :

- laisser les étudiants préparer leurs présentations PowerPoint® ou autre, sans utiliser le site. Le lundi soir, les professeurs choisissent les cas qui seront rentrés dans la base de données par l'auteur lui-même. Celui-ci aura donc un travail supplémentaire à réaliser. C'est une utilisation limitée, mais qui permet d'implémenter la base avec une sélection de cas ;
- laisser le choix aux étudiants de préparer leurs cas avec le site ou l'outil de leur choix. Ceci permet de profiter de l'ensemble des avantages des formulaires : facilité d'emploi, rapidité dans la conception de la présentation. Les cas présentés sans le site et qui présentent un intérêt particulier pourront ensuite être entrés dans la base comme expliqué en premier point.

e) Pérennité

Un outil de ce type, en particulier dans les premiers temps d'utilisation, va être amené à évoluer. Il faut être capable de corriger les erreurs, accéder à la base de données, et également répondre aux questions des utilisateurs. Ceci demande des connaissances minimales en informatique. Si personne ne s'occupe de la maintenance, le site sera sans doute vite amené à disparaître.

Je pense donc qu'il est nécessaire de constituer une équipe chargée de s'occuper :

- de la maintenance et de l'évolution du site ;
- du recueil des idées et commentaires des utilisateurs.

Une étroite collaboration entre le service de médecine et le service informatique et multimédia me semble nécessaire.

Durant quelques mois, je me propose comme volontaire pour intervenir dans ce groupe.

f) Principes concernant l'amélioration du site

Un site web est toujours amené à évoluer. Il est nécessaire pour cela d'être à l'écoute des utilisateurs. C'est pour cette raison qu'on trouve, à la fin des menus, un lien permettant d'envoyer un mail au webmaster. Je suis également en cours de réalisation d'un questionnaire à l'attention des étudiants, pour recueillir leurs impressions et critiques.

Je les interroge sur les points suivants :

- facilité d'utilisation du site ;
- facilité d'envoi d'images ;
- réponse du site à tous leurs besoins pour rentrer leurs cas clinique ;
- comparaison avec l'utilisation de PowerPoint®.

Il leur sera également possible d'écrire un commentaire libre.

Les professeurs qui assistent aux présentations interviendront également dans l'amélioration de l'outil. Ils devront communiquer toutes leurs impressions et volontés au webmaster. Pour cela, l'organisation de réunions régulières (trimestrielles, semestrielles) me semblent intéressante.

Il sera également nécessaire d'améliorer le site en fonction des avancées technologiques. Les langages HTML, PHP, SQL évoluent constamment. Généralement, les anciennes versions restent compatibles, mais il sera sans doute nécessaire de changer quelques lignes de codes dans les années futures.

g) Suggestions d'amélioration du site

Le temps m'a manqué pour développer complètement le site, et je suis en train de concevoir deux fonctionnalités supplémentaires :

- la première concerne la mise en forme du texte dans les formulaires. En utilisant des scripts java, il sera possible d'avoir du texte en gras, en italique, souligné et en couleur ;
- la seconde concerne la mise en place de liens vers les comptes-rendus d'autopsie, fournis par le service d'anatomie-pathologique. Celui-ci dispose en effet d'un outil de création automatique de pages web comportant le rapport d'autopsie, et des photos du cas. Les modalités de gestion pour ces liens restent à définir.

Il serait également intéressant de faire apparaître en couleur les données numériques importantes au sein du cas : valeurs qui sortent de la norme, ou qui permettent de souligner un aspect particulier du cas clinique. Pour cela, je pense qu'il sera nécessaire de modifier les tables dans la base de données, en rajoutant une propriété aux champs des valeurs : "mise en page". Cette propriété pourra prendre plusieurs valeurs, qui conditionneraient l'affichage, par exemple grâce à l'existence de styles différents.

Un site peut être constamment amélioré, mais cela nécessite beaucoup de temps. C'est pour cette raison que je n'ai pas développé ces fonctionnalités.

C. Evolution, développement

1. Réalisation d'un site complet autour des cas cliniques

Dans un premier temps, je voulais réaliser un site complet, avec possibilité de consultation des cas et recherche par mots-clés. Le temps m'a hélas manqué.

Cette partie reste à faire et va permettre de rentabiliser complètement la base de données. Elle permettra d'obtenir un outil complet :

- entrée des cas par les étudiants ;
- présentation clinique ;
- consultation par les étudiants ;

- recherche par mots clés.

Cet outil permettra également de réaliser des présentations de cas en dehors du contexte actuel du lundi soir : lorsque la base de données aura de nombreux cas, il sera possible de réaliser des soirées thématiques, par exemple.

2.Evolution et développement de la base de données

La base de données pourra être complétée par des tables, en fonction des besoins : table de mots clés pour chaque cas, de rubriques, etc.

Compte tenu de la grande diffusion actuelle de l'ensemble PHP MySQL, et de leur développement actif, je pense qu'il ne sera pas nécessaire de changer de SGBD dans les années futures.

3.Evolution et développement du site lui-même

J'ai réalisé la conception des fichiers et de la base en collaboration avec le corps enseignant de l'ENVA, et essentiellement pour cette école.

Il est évident que cet outil est diffusable aux autres écoles vétérinaires francophones désirant développer les cas cliniques, ou à leur tour créer une base de données. Ceci peut être réalisé sans modification des fichiers. Mon écrit et les explications du code leurs sont aussi destinées.

On peut également se servir de la base de données et des fichiers comme base de travail à la création d'un site Internet à l'attention des praticiens. Je suis actuellement en cours de préparation d'un tel site, où les cas seraient rentrés par des vétérinaires en exercice.

On peut aussi envisager un emploi du site en médecine humaine. Je pense que la façon de mener un examen clinique, d'établir un suivi, de réaliser les examens complémentaires ne doit pas trop différer de celle pratiquée en médecine vétérinaire.

Un outil similaire pourrait également être réalisé pour la chirurgie vétérinaire, en particulier à l'ENVA.

Conclusion

Mon objectif initial était de créer un site Internet complet autour des cas cliniques. Il devait permettre l'entrée de cas par les étudiants, leur projection en amphithéâtre et leur consultation par tous les élèves. L'obligation de délais induits par la réforme de l'enseignement vétérinaire m'a contraint à abandonner la réalisation de la dernière fonction.

Après six mois d'autoformation en langage PHP, et six autres mois d'écriture des fichiers, j'ai obtenu un outil entièrement fonctionnel. Les tests que j'ai réalisés sur mon ordinateur, en "local", montrent l'absence de bug, que ce soit dans l'affichage ou dans l'implémentation de la base de données.

Il reste encore une phase importante à réaliser : l'hébergement du site sur les serveurs de l'ENVA et les tests avec des cas réels entrés par des étudiants.

Ce manuel de thèse explique ma démarche dans la réalisation d'un tel site de cas cliniques. Il sera utile à tous ceux qui voudront, à leur tour, s'attaquer à un projet similaire. Il servira également au webmaster du site de l'ENVA.

L'idée de site interactif de cas cliniques, et de création d'une base de données peut être en effet reprise par d'autres écoles, en sciences vétérinaires ou en médecine humaine.

Je pense que ce type de site Internet peut avoir un grand intérêt pédagogique. Et le développement actuel d'Internet tend à rendre cette technologie indispensable.

De mon côté, je travaille à la réalisation d'un site de cas cliniques vétérinaires à l'attention des praticiens. Il devrait voir le jour en 2004.

Index des figures

Figure 1 : principes d'une page web statique [15].....	16
Figure 2 : principes d'une page web dynamique	18
Figure 3 : copie d'écran montrant l'icône Easyphp et ses fonctions	22
Figure 4 : les différentes étapes de la méthode MERISE.....	23
Figure 5 : bilan des flux.....	24
Figure 6 : exemple de MCD à 2 entités.....	26
Figure 7 : MCD final.....	27
Figure 8 : le MLD.....	29
Figure 9 : choix du nom de la base de données avec phpMyAdmin.....	30
Figure 10 : choix du nom de la table et du nombre de champs avec phpMyAdmin.....	31
Figure 11 : choix du nom des champs et de leurs propriétés avec phpMyAdmin	31
Figure 12 : récupération du script de création de la base de données avec phpMyAdmin	32
Figure 13 : les pseudo-frames de la page index.php	33
Figure 14 : page biochJ0	41
Figure 15 : feuille de style des menus	44
Figure 16 : feuille de style de la partie formulaires.....	44
Figure 17 : feuille de style de la partie présentation	45
Figure 18 : icône d'erreur sur la page, avec Internet Explorer 6.0	48
Figure 19 : affichage du code source d'une page web sous Internet Explorer 6.0	49

Index des fichiers php détaillés et des fonctions

Code 1 : fichier index.php	34
Code 2 : code partiel du fichier centre.php	36
Code 3 : fichier testconnect.php	37
Code 4 : fichier conf.php	38
Code 5 : fonction datefr	42
Code 6 fonction nl2br	42

Bibliographie

- 1 : *Dreamweaver MX*. Paris : Macromedia, 2001.
- 2 : *Internet Explorer 6.0*. Microsoft Corporation, 2003.
- 3 : *Powerpoint*. Microsoft Corporation, 2003.
- 4 : ABBAL L, FAIVRE E, MURAIL T. *Easyphp 1.6.0.0*. [en-ligne], Mise à jour le 26 novembre 2003 [<http://www.easyphp.net>], (consulté le 30 novembre 2003).
- 5 : ACTOO SARL. *PHP France* [en-ligne], Mise à jour le 26 novembre 2003 [<http://www.phpfrance.com>], (consulté le 30 novembre 2003).
- 6 : ADDA L. *PHPScripts : L'Annuaire Francophone des Scripts PHP / MySQL* [en-ligne], Mise à jour le 8 octobre 2003 [<http://www.phpscripts-fr.net>], (consulté le 30 novembre 2003).
- 7 : BENCHMARK GROUP. *Le Journal du Net – Les chiffres clés* [en-ligne], Mise à jour le 10 novembre 2003 [<http://www.journaldunet.com/chiffres-cles.shtml>], (consulté le 12 novembre 2003).
- 8 : BROUARD F. *Analyse des données – Méthode MERISE* [en-ligne], Mise à jour le 29 juin 2000 [<http://perso.club-internet.fr/brouardf/SGBDRmerise.htm>], (consulté le 10 août 2003).
- 9 : HEUTE D, HEUTE T. *Tout est facile !* [en-ligne], Mise à jour le 27 novembre 2003 [<http://www.toutestfacile.com>], (consulté le 30 novembre 2003).
- 10 : MOUEZANT A, POULET A. *Annuaire de scripts* [en-ligne], Mise à jour le 10 novembre 2003 [<http://www.comscripts.com/>], (consulté le 11 novembre 2003).
- 11 : MUNZ S. *SelfHTML* [en-ligne], Mise à jour le 27 janvier 2003 [<http://selfhtml.selfhtml.com.fr/index.htm>] (consulté le 5 août 2003).

- 12 : MYSQL AB. *MySQL Manual* [en-ligne], Mise à jour le 12 juillet 2003
[<http://www.mysql.com/doc/fr/index.html>] (consulté le 17 juillet 2003).
- 13 : Nexen services. *Nexen.net : portail français de PHP et MySQL* [en-ligne], Mise à jour le 4 juillet 2003 [<http://www.nexen.net/index.php>] (consulté le 17 juillet 2003).
- 14 : Phpdébutant. *Bienvenue sur PHP Débutant* [en-ligne], Mise à jour le 25 novembre 2003
[<http://www.phpdebutant.org>], (consulté le 30 novembre 2003).
- 15 : PILOU J-F. *Comment ça marche ?* [en-ligne], Mise à jour le 19 novembre 2003
[<http://www.commentcamarche.net>], (consulté le 30 novembre 2003).
- 16 : RIGAUX P. *Pratique de MySQL et PHP*. Paris : Ed. O'Reilly, 2001, 369p.
- 17 : The PHP Group. *PHP: Hypertext Preprocessor* [en-ligne], Mise à jour le 11 novembre 2003
[<http://www.php.net>], (consulté le 30 novembre 2003).
- 18 : VIDAL M. *Mémo MERISE* [en-ligne], [<http://perso.wanadoo.fr/matthieu.vidal/default.htm>],
(consulté le 5 août 2003).
- 19 : WELLING L, THOMSON L. *PHP et MySQL*. Paris : CampusPress, 2001, 923p.

ANNEXES

Guide d'utilisation

Site de cas cliniques ENVA

Un site Internet a été développé pour vous aider dans la réalisation de vos cas cliniques, et pour constituer une base de données de cas. L'avantage est de pouvoir rentrer les données (résultats d'examens complémentaires, données cliniques) tout au long de la semaine, en remplissant un formulaire, et de ne pas avoir à gérer la mise en page, qui est faite directement par le site.

La présentation orale se fera avec les pages créées, qui sont assez proches des réalisations Power Point.

Les quelques lignes suivantes vous expliquent la marche à suivre. N'hésitez pas à contacter le webmaster en cas de problèmes.

I. Connectez-vous sur le site : *mettre ici l'adresse définitive*

II. Entrez votre pseudonyme et votre mot de passe ou inscrivez-vous (chaque pseudonyme ne peut réaliser qu'un seul cas)

Le menu à gauche présente la liste des formulaires à remplir.

III. Quelques astuces techniques :

- Pour ajouter des images (radios...) la taille maximale autorisée est de 100ko. Je vous recommande de mettre des fichiers de type jpeg, en résolution 72dpi et environ 600x800 pixels

- En cliquant sur "voir le résultat", l'explorateur ouvre une nouvelle fenêtre correspondant à votre présentation en amphithéâtre.

IV. En cas de bugs, ou de problèmes, vous pouvez joindre le webmaster par mail : *mettre ici l'adresse électronique.*

Script de création de la base de données

Fichier base.sql

```
# phpMyAdmin MySQL-Dump
# version 2.2.6
# http://phpwizard.net/phpMyAdmin/
# http://www.phpmyadmin.net/ (download page)
#
# Serveur: localhost
# Généré le : Dimanche 30 Novembre 2003 à 09:40
# Version du serveur: 3.23.49
# Version de PHP: 4.2.0
# Base de données: `these`
# -----

#
# Structure de la table `anapath`
#

CREATE TABLE anapath (
  ID_IMAGE int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  TYPE_IMAGE varchar(20) NOT NULL default "",
  DESCRIPTION_IMAGE text,
  DATE date default NULL,
  NOM varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_IMAGE),
  KEY CONTIENT_IMAGE_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
# Structure de la table `appareils_j0`
#

CREATE TABLE appareils_j0 (
  ID_APPAREIL int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  NOM_APPAREIL text,
  DESCRIPTION_EXAM text,
  PRIMARY KEY (ID_APPAREIL),
  KEY CONTIENT_APPAREIL_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
```

```

# Structure de la table `autre`
#

CREATE TABLE autre (
  ID_AUTRE int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  TYPE_EXAM varchar(20) default NULL,
  TYPE_IMAGE varchar(20) default NULL,
  DESCRIPTION_IMAGE text,
  DATE date default NULL,
  NOM varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_AUTRE),
  KEY CONTIENT_IMAGE_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
# Structure de la table `bibliographie`
#

CREATE TABLE bibliographie (
  ID_BIBLIO int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  REFERENCE text,
  PRIMARY KEY (ID_BIBLIO)
) TYPE=MyISAM;
# -----

#
# Structure de la table `bioch`
#

CREATE TABLE bioch (
  ID_BIOCH int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  DATE_BIOCH date NOT NULL default '0000-00-00',
  PARAMETRE_BIOCH varchar(200) NOT NULL default "",
  VALEUR_BIOCH double NOT NULL default '0',
  UNITE_BIOCH varchar(15) default NULL,
  NORME_INF double default NULL,
  NORME_SUP double default NULL,
  PRIMARY KEY (ID_BIOCH),
  KEY CONTIENT_BIOCH_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
# Structure de la table `cardio`
#

CREATE TABLE cardio (
  ID_IMAGE int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  TYPE_IMAGE varchar(20) NOT NULL default "",
  DESCRIPTION_IMAGE text,
  DATE date default NULL,
  NOM varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_IMAGE),
  KEY CONTIENT_IMAGE_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

```

```

#
# Structure de la table `cas_clinique`
#

CREATE TABLE cas_clinique (
  NUMERO_DE_CAS int(11) NOT NULL auto_increment,
  PSEUDO varchar(15) default NULL,
  NUMERO_DE_DOSSIER varchar(10) NOT NULL default "",
  PROFESSEUR varchar(15) NOT NULL default "",
  DATE_SEMAINE date default NULL,
  NOM_ANIMAL varchar(15) NOT NULL default "",
  ESPECE varchar(15) NOT NULL default "",

  RACE_ANIMAL varchar(20) NOT NULL default "",
  SEXE varchar(10) NOT NULL default "",
  STERILISE varchar(20) NOT NULL default "",
  DATE_NAISSANCE date default NULL,
  NOM_PROPRIO varchar(15) default NULL,
  DATE_PREMIERE_CONSULT date default NULL,
  COMMEMO text NOT NULL,
  VACCINATION text,
  VERMIFUGATION text,
  MOTIF_DE_CONSULTATION text NOT NULL,
  ANAMNESE text NOT NULL,
  POIDS_J0 double default NULL,
  EVOLUTION_POIDS varchar(15) default NULL,
  TEMP_RECTALE double default NULL,
  FREQ_CARDIAQ smallint(6) default NULL,
  MUQ varchar(15) default NULL,
  TRC varchar(15) default NULL,
  DESH2O varchar(15) default NULL,
  CONCLUSION_J0 text,
  CONCLUSION_FINALE text,
  TRAITEMENTS text NOT NULL,
  PRIMARY KEY (NUMERO_DE_CAS),
  KEY FAIT_FK (PSEUDO)
) TYPE=MyISAM;

```

```
# -----
```

```

#
# Structure de la table `conclusion_hypotheses_j0`
#

```

```

CREATE TABLE conclusion_hypotheses_j0 (
  ID_CCL int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  CONCLUSION_HYP text,
  HYPOTHESE_CCL text,
  PRIMARY KEY (ID_CCL),
  KEY CONTIENT_CCL_HYP_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;

```

```
# -----
```

```
#
```

```

# Structure de la table `conclusion_traitement`
#

CREATE TABLE conclusion_traitement (
  ID_CCLTRAIT int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  CONCL_TRAIT text,
  TRAIT_CCL text,
  PRIMARY KEY (ID_CCLTRAIT),
  KEY CONTIENT_CCL_HYP_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
# Structure de la table `conclusion_traitement_j0`
#

CREATE TABLE conclusion_traitement_j0 (
  ID_CCLTRAIT int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  CONCL_TRAIT text,
  TRAIT_CCL text,
  PRIMARY KEY (ID_CCLTRAIT),
  KEY CONTIENT_CCL_HYP_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
# Structure de la table `conclusions`
#

CREATE TABLE conclusions (
  ID_CONCLUSION int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  TYPE_EXAMEN text,
  DESCR_CCL text,
  PRIMARY KEY (ID_CONCLUSION),
  KEY CONTIENT_CONCLUION_EXAM_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
# Structure de la table `dermato`
#

CREATE TABLE dermato (
  ID_IMAGE int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  TYPE_IMAGE varchar(20) NOT NULL default "",
  DESCRIPTION_IMAGE text,
  DATE date default NULL,
  NOM varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_IMAGE),
  KEY CONTIENT_IMAGE_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#

```

```

# Structure de la table `discussion`
#

CREATE TABLE discussion (
  ID_DISCUSSION tinyint(4) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  DISCUSSION text NOT NULL,
  PRIMARY KEY (ID_DISCUSSION)
) TYPE=MyISAM;
# -----

#
# Structure de la table `finances`
#

CREATE TABLE finances (
  ID_FINANCES tinyint(4) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  DISCUSSION text NOT NULL,
  PRIMARY KEY (ID_FINANCES)
) TYPE=MyISAM;
# -----

#
# Structure de la table `hormones`
#

CREATE TABLE hormones (
  ID_HORM int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  DATE_HORM date NOT NULL default '0000-00-00',
  PARAMETRE_HORM varchar(200) NOT NULL default "",
  VALEUR_HORM double NOT NULL default '0',
  UNITE_HORM text NOT NULL,
  NORME_INF double default NULL,
  NORME_SUP double default NULL,
  PRIMARY KEY (ID_HORM),
  KEY CONTIENT_HORM_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#
# Structure de la table `imagerie`
#

CREATE TABLE imagerie (
  ID_IMAGE int(11) NOT NULL auto_increment,
  NUMERO_DE_CAS int(11) NOT NULL default '0',
  TYPE_IMAGE varchar(20) NOT NULL default "",
  DESCRIPTION_IMAGE text,
  DATE date default NULL,
  NOM varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_IMAGE),
  KEY CONTIENT_IMAGE_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;
# -----

#

```

Structure de la table `nf`
#

```
CREATE TABLE nf (  
  ID_NF int(11) NOT NULL auto_increment,  
  NUMERO_DE_CAS int(11) NOT NULL default '0',  
  DATE_NF date NOT NULL default '0000-00-00',  
  LEUCOCYTES double(10,2) NOT NULL default '0.00',  
  HEMATIES double(10,2) NOT NULL default '0.00',  
  HEMOGLOBINE double(10,2) NOT NULL default '0.00',  
  HEMATOCRITE double(10,2) NOT NULL default '0.00',  
  VGM double(10,2) NOT NULL default '0.00',  
  TCMH double(10,2) NOT NULL default '0.00',  
  CCMH double(10,2) NOT NULL default '0.00',  
  RETICULOCYTES double(10,2) NOT NULL default '0.00',  
  PLAQUETTES double(10,2) NOT NULL default '0.00',  
  NEUTRO_NONSEG double(10,2) NOT NULL default '0.00',  
  NEUTRO_SEG double(10,2) NOT NULL default '0.00',  
  EOSINOPHILES double(10,2) NOT NULL default '0.00',  
  BASOPHILES double(10,2) NOT NULL default '0.00',  
  LYMPHOCYTES double(10,2) NOT NULL default '0.00',  
  MONOCYTES double(10,2) NOT NULL default '0.00',  
  PRIMARY KEY (ID_NF),  
  KEY CONTIENT_NF_FK (NUMERO_DE_CAS)  
) TYPE=MyISAM;
```

#

Structure de la table `suivi_clinique`
#

```
CREATE TABLE suivi_clinique (  
  ID_SUIVI int(11) NOT NULL auto_increment,  
  NUMERO_DE_CAS int(11) NOT NULL default '0',  
  DATE_SUIVI date NOT NULL default '0000-00-00',  
  DESCRIPTION_SUIVI text NOT NULL,  
  PRIMARY KEY (ID_SUIVI)  
) TYPE=MyISAM;
```

#

Structure de la table `urine`
#

```
CREATE TABLE urine (  
  ID_URINE int(11) NOT NULL auto_increment,  
  NUMERO_DE_CAS int(11) NOT NULL default '0',  
  DATE_URINE date NOT NULL default '0000-00-00',  
  PARAM_URINE varchar(30) NOT NULL default "",  
  VALEUR_URINE double NOT NULL default '0',  
  UNITE_URINE varchar(30) NOT NULL default "",  
  NORME_INF double default NULL,  
  NORME_SUP double default NULL,  
  PRIMARY KEY (ID_URINE),  
  KEY CONTIENT_URINE_FK (NUMERO_DE_CAS)  
) TYPE=MyISAM;
```

#

```

# Structure de la table `utilisateur`
#

CREATE TABLE utilisateur (
  NOM varchar(15) NOT NULL default "",
  PRENOM varchar(15) NOT NULL default "",
  ID_MB varchar(20) NOT NULL default '0',
  GROUPE smallint(6) NOT NULL default '0',
  MAIL varchar(30) NOT NULL default "",
  PSEUDO varchar(15) NOT NULL default "",
  NUMERO_DE_CAS int(11) default NULL,
  PASSE varchar(15) NOT NULL default "",
  PRIMARY KEY (PSEUDO),
  KEY FAIT2_FK (NUMERO_DE_CAS)
) TYPE=MyISAM;

# phpMyAdmin MySQL-Dump
# version 2.2.6
# http://phpwizard.net/phpMyAdmin/
# http://www.phpmyadmin.net/ (download page)
#
# Serveur: localhost
# Généré le : Dimanche 30 Novembre 2003 à 09:40
# Version du serveur: 3.23.49
# Version de PHP: 4.2.0
# Base de données: `these`
# -----

#
# Structure de la table `param_anapath`
#

CREATE TABLE param_anapath (
  ID_NOMANAPATH tinyint(4) NOT NULL auto_increment,
  NOM_ANAPATH varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_NOMANAPATH)
) TYPE=MyISAM;

#
# Contenu de la table `param_anapath`
#

INSERT INTO param_anapath VALUES (1, 'biopsie');
INSERT INTO param_anapath VALUES (2, 'pièce d'exérèse');
INSERT INTO param_anapath VALUES (3, 'autopsie');
INSERT INTO param_anapath VALUES (4, 'cytologie');
# -----

#
# Structure de la table `param_appareil`
#

CREATE TABLE param_appareil (
  ID_APPAREIL tinyint(4) NOT NULL auto_increment,
  NOM_APPAREIL varchar(40) NOT NULL default "",
  PRIMARY KEY (ID_APPAREIL)
) TYPE=MyISAM;

#

```

```

# Contenu de la table `param_appareil`
#

INSERT INTO param_appareil VALUES (1, 'appareil respiratoire');
INSERT INTO param_appareil VALUES (2, 'appareil cardio-vasculaire');
INSERT INTO param_appareil VALUES (3, 'ophtalmologie');
INSERT INTO param_appareil VALUES (4, 'neurologie');
INSERT INTO param_appareil VALUES (5, 'S.R.H.');
```

```

#
# Structure de la table `param_bioch`
#
```

```

CREATE TABLE param_bioch (
  ID_NOMBIOCH tinyint(4) NOT NULL auto_increment,
  NOM_BIOCH varchar(20) NOT NULL default "",
  UNIT_BIOCH varchar(6) NOT NULL default "",
  PRIMARY KEY (ID_NOMBIOCH)
) TYPE=MyISAM;
```

```

#
# Contenu de la table `param_bioch`
#
```

```

INSERT INTO param_bioch VALUES (1, 'urée', 'g/l');
INSERT INTO param_bioch VALUES (2, 'créatinine', 'mg/l');
INSERT INTO param_bioch VALUES (3, 'glucose', 'g/l');
INSERT INTO param_bioch VALUES (4, 'PAL', 'UI/l');
INSERT INTO param_bioch VALUES (5, 'ALAT', 'UI/l');
INSERT INTO param_bioch VALUES (6, 'lipase', 'UI/l');
INSERT INTO param_bioch VALUES (7, 'cholesterol', 'g/l');
INSERT INTO param_bioch VALUES (8, 'protéines totales', 'g/l');
INSERT INTO param_bioch VALUES (9, 'acides biliaires', 'µmol/l');
INSERT INTO param_bioch VALUES (10, 'triglycérides', 'g/l');
INSERT INTO param_bioch VALUES (11, 'amylase', 'UI/l');
INSERT INTO param_bioch VALUES (12, 'hémoglobine', 'g/l');
INSERT INTO param_bioch VALUES (13, 'bilirubine totale', 'mg/l');
INSERT INTO param_bioch VALUES (14, 'bilirubine conjuguée', 'mg/l');
INSERT INTO param_bioch VALUES (15, 'acide urique', 'mg/l');
INSERT INTO param_bioch VALUES (16, 'amoniaque', 'µmol/l');
INSERT INTO param_bioch VALUES (17, 'albumines', 'g/l');
INSERT INTO param_bioch VALUES (18, 'globulines', 'g/l');
INSERT INTO param_bioch VALUES (19, 'calcium', 'mg/l');
INSERT INTO param_bioch VALUES (20, 'phosphates', 'mg/l');
INSERT INTO param_bioch VALUES (21, 'trou anionique', 'mEq/l');
INSERT INTO param_bioch VALUES (22, 'ions bicarbonates', 'mEq/l');
INSERT INTO param_bioch VALUES (23, 'magnésium', 'mg/l');
INSERT INTO param_bioch VALUES (24, 'sodium', 'mmol/l');
INSERT INTO param_bioch VALUES (25, 'potassium', 'mmol/l');
INSERT INTO param_bioch VALUES (26, 'chlore', 'mmol/l');
INSERT INTO param_bioch VALUES (27, 'pH sanguin', '');
INSERT INTO param_bioch VALUES (28, 'CPK', 'UI/l');
INSERT INTO param_bioch VALUES (29, 'GGT', 'UI/l');
```

```

#
```

```

# Structure de la table `param_cardio`
#

CREATE TABLE param_cardio (
  ID_NOMCARDIO tinyint(4) NOT NULL auto_increment,
  NOM_CARDIO varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_NOMCARDIO)
) TYPE=MyISAM;

#
# Contenu de la table `param_cardio`
#

INSERT INTO param_cardio VALUES (1, 'radio');
INSERT INTO param_cardio VALUES (2, 'échographie');
INSERT INTO param_cardio VALUES (3, 'ECG');
# -----

#
# Structure de la table `param_dermato`
#

CREATE TABLE param_dermato (
  ID_NOMDERMATO tinyint(4) NOT NULL auto_increment,
  NOM_DERMATO varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_NOMDERMATO)
) TYPE=MyISAM;

#
# Contenu de la table `param_dermato`
#

INSERT INTO param_dermato VALUES (1, 'frottis');
INSERT INTO param_dermato VALUES (2, 'calque');
INSERT INTO param_dermato VALUES (3, 'examen direct');
INSERT INTO param_dermato VALUES (4, 'scotch test');
INSERT INTO param_dermato VALUES (5, 'raclage');
INSERT INTO param_dermato VALUES (6, 'culture');
# -----

#
# Structure de la table `param_horm`
#

CREATE TABLE param_horm (
  ID_NOMHORM tinyint(4) NOT NULL auto_increment,
  NOM_HORM varchar(200) NOT NULL default "",
  UNIT_HORM varchar(6) NOT NULL default "",
  PRIMARY KEY (ID_NOMHORM)
) TYPE=MyISAM;

#

```

Contenu de la table `param_horm`

#

```
INSERT INTO param_horm VALUES (11, 'TSH canine (cTSH)', 'ng/ml');
INSERT INTO param_horm VALUES (10, 'thyroninémie libre (T4l)', 'pmol/l');
INSERT INTO param_horm VALUES (9, 'thyroxinémie totale (T4t)', 'nmol/l');
INSERT INTO param_horm VALUES (12, 'thyroxinémie - stimulation par la TRH - T0', 'nmol/l');
INSERT INTO param_horm VALUES (13, 'thyroxinémie - stimulation par la TRH - après stim', 'nmol/l');
INSERT INTO param_horm VALUES (14, 'thyroxinémie - freinage par un analogue de la triiodothyronine
- J0', 'nmol/l');
INSERT INTO param_horm VALUES (15, 'thyroxinémie - freinage par un analogue de la triiodothyronine
- J0+3j', 'nmol/l');
INSERT INTO param_horm VALUES (16, 'anticorps anti-thyroglobuline (AATg)', '');
INSERT INTO param_horm VALUES (17, 'cortisolémie - stimulation par l'ACTH - T0', 'nmol/l');
INSERT INTO param_horm VALUES (18, 'cortisolémie - stimulation par l'ACTH - T0+30', 'nmol/l');
INSERT INTO param_horm VALUES (19, 'cortisolémie - stimulation par l'ACTH - T0+1h', 'nmol/l');
INSERT INTO param_horm VALUES (20, 'cortisolémie - stimulation par l'ACTH - T0+1h30', 'nmol/l');
INSERT INTO param_horm VALUES (21, 'cortisolémie - freinage par la dexaméthasone à dose faible -
T0', 'nmol/l');
INSERT INTO param_horm VALUES (22, 'cortisolémie - freinage par la dexaméthasone à dose faible -
T0+4h', 'nmol/l');
INSERT INTO param_horm VALUES (23, 'cortisolémie - freinage par la dexaméthasone à dose faible -
T0+8h', 'nmol/l');
INSERT INTO param_horm VALUES (24, 'rapport cortisol / créinine urinaires (RCCU)', '');
INSERT INTO param_horm VALUES (25, 'cortisolémie - freinage par la dexaméthasone à dose forte -
T0', 'nmol/l');
INSERT INTO param_horm VALUES (26, 'cortisolémie - freinage par la dexaméthasone à dose forte -
T0+4h', 'nmol/l');
INSERT INTO param_horm VALUES (27, 'cortisolémie - freinage par la dexaméthasone à dose forte -
T0+8h', 'nmol/l');
INSERT INTO param_horm VALUES (28, 'ACTH', 'pg/ml');
INSERT INTO param_horm VALUES (29, 'aldostéronémie - stimulation par l'ACTH - T0', 'pg/ml');
INSERT INTO param_horm VALUES (30, 'aldostéronémie - stimulation par l'ACTH - après stimulation',
'pg/ml');
INSERT INTO param_horm VALUES (31, 'aldostéronémie basale', 'pg/ml');
INSERT INTO param_horm VALUES (32, 'activité rénine plasmatique basale', '');
INSERT INTO param_horm VALUES (33, '17-hydroxyprogestéronémie - stimulation par l'ACTH - T0', '');
INSERT INTO param_horm VALUES (34, '17-hydroxyprogestéronémie - stimulation par l'ACTH - après
stimulation', '');
INSERT INTO param_horm VALUES (35, 'fructosamines sériques', 'µmol/l');
INSERT INTO param_horm VALUES (36, 'hémoglobine glycosylée sérique', '');
INSERT INTO param_horm VALUES (37, 'IGF-1 plasmatique', 'ng/ml');
INSERT INTO param_horm VALUES (38, 'progestéronémie basale', 'nmol/l');
INSERT INTO param_horm VALUES (39, 'estradiolémie basale', 'pmol/l');
INSERT INTO param_horm VALUES (40, 'prolactinémie basale', 'ng/ml');
INSERT INTO param_horm VALUES (41, 'LH plasmatique', '');
INSERT INTO param_horm VALUES (42, 'relaxinémie basale', '');
INSERT INTO param_horm VALUES (43, 'testotéronémie basale', 'nmol/l');
INSERT INTO param_horm VALUES (44, 'testotéronémie - stimulation par la gonadotropine chorionique
(hCG) +', '');
INSERT INTO param_horm VALUES (45, 'testotéronémie - stimulation par la gonadotropine chorionique
(hCG) - T0', 'nmol/l');
INSERT INTO param_horm VALUES (46, 'testotéronémie - stimulation par la gonadotropine chorionique
(hCG) - T0+24h', 'nmol/l');
INSERT INTO param_horm VALUES (47, 'progestéronémie - stimulation par l'hCG - T0', '');
INSERT INTO param_horm VALUES (48, 'progestéronémie - stimulation par l'hCG - T0+1h', '');
INSERT INTO param_horm VALUES (49, 'estradiolémie - stimulation par l'hCG - T0', '');
INSERT INTO param_horm VALUES (50, 'estradiolémie - stimulation par l'hCG - T0+1h', '');
```

#

```

# Structure de la table `param_image`
#

CREATE TABLE param_image (
  ID_NOMIMAGE tinyint(4) NOT NULL auto_increment,
  NOM_IMAGE varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_NOMIMAGE)
) TYPE=MyISAM;

#
# Contenu de la table `param_image`
#

INSERT INTO param_image VALUES (1, 'radio');
INSERT INTO param_image VALUES (2, 'échographie');
INSERT INTO param_image VALUES (3, 'IRM');
INSERT INTO param_image VALUES (4, 'scanner');
# -----

#
# Structure de la table `param_prof`
#

CREATE TABLE param_prof (
  ID_NOMPROF tinyint(4) NOT NULL auto_increment,
  NOM_PROF varchar(20) NOT NULL default "",
  PRIMARY KEY (ID_NOMPROF)
) TYPE=MyISAM;

#
# Contenu de la table `param_prof`
#

INSERT INTO param_prof VALUES (1, 'Pr. Pouchelon');
INSERT INTO param_prof VALUES (2, 'Dr. Rosenberg');
INSERT INTO param_prof VALUES (3, 'Dr. Blot');
INSERT INTO param_prof VALUES (4, 'Dr. Chetboul');
INSERT INTO param_prof VALUES (5, 'Pr. Cotard');
# -----

#
# Structure de la table `param_urine`
#

CREATE TABLE param_urine (
  ID_NOMURINE tinyint(4) NOT NULL auto_increment,
  NOM_URINE varchar(20) NOT NULL default "",
  UNIT_URINE varchar(6) NOT NULL default "",
  PRIMARY KEY (ID_NOMURINE)
) TYPE=MyISAM;

#
# Contenu de la table `param_urine`
#

INSERT INTO param_urine VALUES (1, 'proteines', 'croix');
INSERT INTO param_urine VALUES (2, 'cortisol', "");
INSERT INTO param_urine VALUES (3, 'créatinine', "");
INSERT INTO param_urine VALUES (4, 'glucose', 'g/l');

```

Création d'un outil informatique d'archivage et d'aide à la réalisation des cas cliniques à l'ENVA

NOM et Prénom : GAVIGLIO Vincent

Résumé

Les étudiants vétérinaires de l'ENVA exposent oralement des cas cliniques de médecine, chaque semaine depuis près de 30 ans. L'intérêt pédagogique de ces présentations est considérable, mais aucun archivage n'a jamais été réalisé. L'auteur a donc voulu développer un outil informatique répondant à ce manque, sous la forme d'un site web relié à une base de données.

Dans un premier temps, il décrit la situation actuelle des exposés, les intervenants et leurs méthodes de travail. La seconde partie présente la trame que doivent respecter les étudiants, sous la forme d'un modèle polyvalent de cas cliniques. Celui-ci sert de point de départ à la création de la base de données, à l'aide de la méthode MERISE, et à l'écriture des pages web, en langage PHP. L'auteur explique ses choix techniques concernant le site. Enfin, la dernière partie dresse un bilan de l'outil, ses fonctions et ses limites. L'auteur y aborde les améliorations possibles, les perspectives d'évolution et d'utilisation dans d'autres domaines.

Mots clés : outil informatique, base de données, Internet, langage PHP, cas cliniques, médecine

Jury :

Président : Pr.

Directeur : Pr. POUCHELON

Assesseur : Pr. FONTAINE

Adresse de l'auteur :

M. GAVIGLIO Vincent

18bis, avenue de l'Abbé Cabrol

13260 Cassis

Conception of a data-processing tool made to create and file clinical cases in National Veterinary School of Alfort

NAME : GAVIGLIO

Surname : Vincent

Summary

The students of the National Veterinary School of Alfort orally expose clinical cases of medicine, each week since 30 years. The teaching interest of these presentations is considerable, but no filing was ever realised. The author thus wanted to develop a data-processing tool answering to this lack, creating a Web site connected to a data base.

Initially, he describes the current situation of the talks, the speakers and their working methods. The second part presents the scheme that the students must respect, in the form of a general-purpose model of clinical cases. This one is used as starting point with the creation of the data base, using the MERISE method, and with the writing of the Web pages, in PHP language. The author explains his technical choices concerning the website. The last part draws up an assessment of the tool, its functions and its limits. The author approaches here the possible improvements, the prospects of evolution and use in other domains.

Key words : data-processing tool, database, Internet, PHP language, clinical cases, medicine

Jury :

Président : Pr.

Director : Pr. Pouchelon

Assessor : Pr. Fontaine

Author's address :

M. GAVIGLIO Vincent
18bis, avenue de l'Abbé Cabrol
13260 Cassis