

Cahier des spécifications

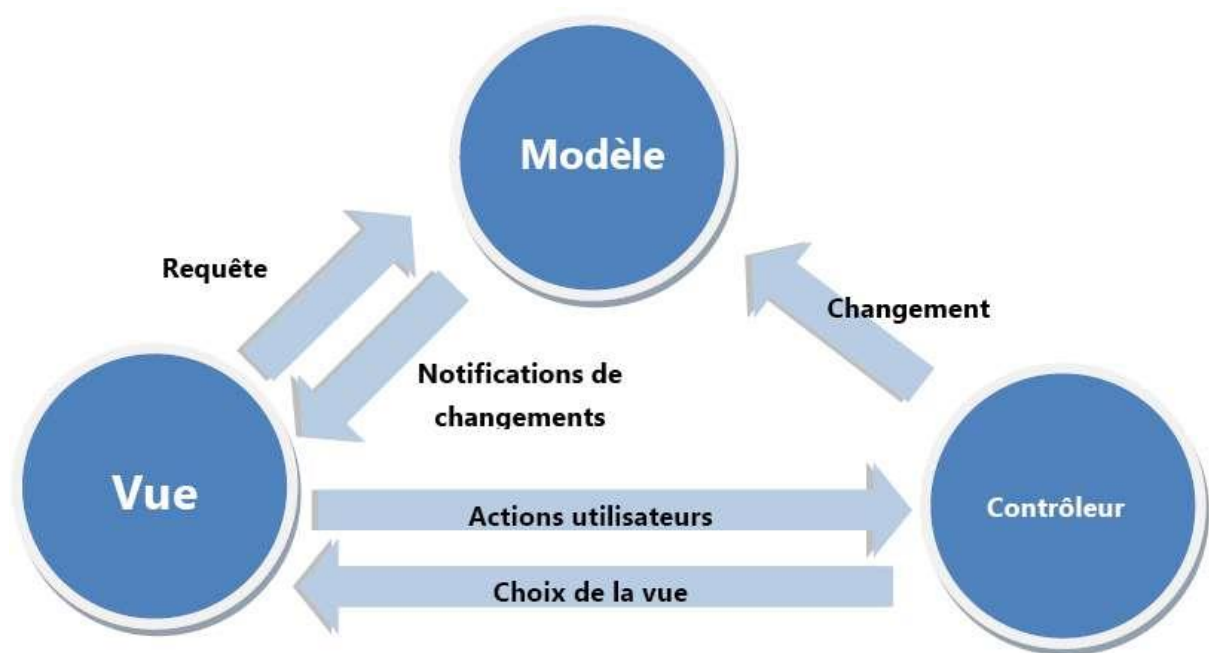
GROUPE 2
DISII 2016-2017
CCI CHATEAUROUX

ARCHITECTURE LOGICIELLE :

Le projet sera fait à l'aide d'un MVC (Modèle Vue Contrôleur), qu'est-ce que c'est ?

Architecture Modèle/Vue/Contrôleur

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface. L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application. Dans l'architecture MVC, les rôles des trois entités sont les suivants: Modèle : données (accès et mise à jour). Vue : interface utilisateur (entrées et sorties) Contrôleur : gestion des événements et synchronisation



Le modèle :

représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est responsable de récupérer les données, de les convertir pour notre application, tels que le traitement, la validation, l'association et beaucoup d'autres tâches concernant la manipulation des données. L'objet Modèle peut être vu comme la première couche d'interaction avec n'importe quelle base de données que l'on pourrait utiliser pour notre application.

La vue :

La Vue retourne une présentation des données venant du modèle. Etant séparée par les Objets Modèle, elle est responsable de l'utilisation des informations dont elle dispose pour produire une interface de présentation de notre application. Par exemple, de la même manière que le modèle retourne un ensemble de données, la Vue utilise ces données pour fournir une page HTML les contenant.

Le controleur :

gère les requêtes des utilisateurs. Il est responsable des réponses avec le Modèle et la Vue. Le Controleur peut être imaginé comme un manager qui a pour mission que toutes les ressources souhaitées pour accomplir une tâche soient faite correctement.

CHARTE GRAPHIQUE :



CHARTE GRAPHIQUE PROJET N°2

Le client est le service vétérinaire d'un parc simiesque, abritant diverses espèces de singes, le site conserve l'identité visuelle du parc familial ; usant ainsi de tons verts et jaunes ; et d'un graphisme relativement simple sans être enfantin.

Couleurs



#6cffb3 / #61e5a1 / #367f59 / 1b402d



#ffc352 / #e5b04a / #7f6229 / #5a451d



#eeffc3 / #d6e5af / #777f61 / #3b4031

Polices

Affichage, Gros-titres

Righteous Taille 12

Righteous Taille 14

Righteous Taille 16

Articles des fiches animaux, envoi de mails

Alfa Slab One Taille 12

Alfa Slab One Taille 14

Alfa Slab One Taille 16

DICTIONNAIRE DE DONNEES :

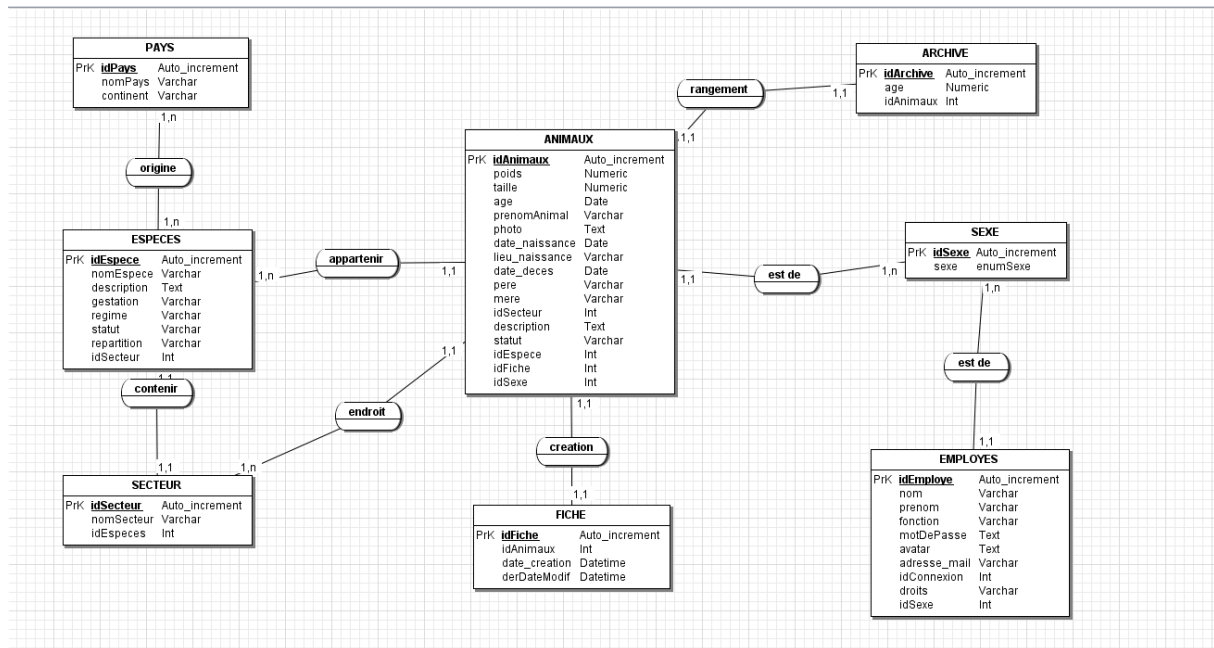
1 : PAYS			
Enregistrements	Entité	Propriété/longueur	
idPays	PAYS	Auto-Incrémenter	11
nomPays	PAYS	Varchar	40
continent	PAYS	Varchar	40
2 : SEXE			
Enregistrements	Entité	Propriété/longueur	
idSexe	SEXE	Auto-Incrémenter	11
sexe	SEXE	EnumSexe	2
3 : EMPLOYES			
Enregistrements	Entité	Propriété/longueur	
idEmploye	EMPLOYES	Auto-Incrémenter	11
idConnexion	EMPLOYES	Varchar	40
nom	EMPLOYES	Varchar	40
prenom	EMPLOYES	Varchar	40
fonction	EMPLOYES	Varchar	40
motDePasse	EMPLOYES	Text	40
adresse_mail	EMPLOYES	Varchar	40
avatar	EMPLOYES	Text	40
droits	EMPLOYES	Int	3
idSexe	EMPLOYES	Int	11

4 : ESPECES			
Enregistrements	Entité	Propriété/longueur	
idEspece	ESPECES	Auto-Incrémenter	11
nomEspece	ESPECES	Varchar	40
description	ESPECES	Text	255
gestation	ESPECES	Varchar	25
regime	ESPECES	Varchar	40
statut	ESPECES	Varchar	25
repartition	ESPECES	Varchar	50
idSecteur	ESPECES	Int	11
5 : SECTEUR			
Enregistrements	Entité	Propriété/longueur	
idSecteur	SECTEUR	Auto-Incrémenter	11
nomSecteur	SECTEUR	Varchar	25
idEspece	SECTEUR	Int	11
6 : ORIGINE			
Enregistrements	Entité	Propriété/longueur	
idPays	ORIGINE	Auto-Incrémenter	11
idEspece	ORIGINE	Auto-Incrémenter	11

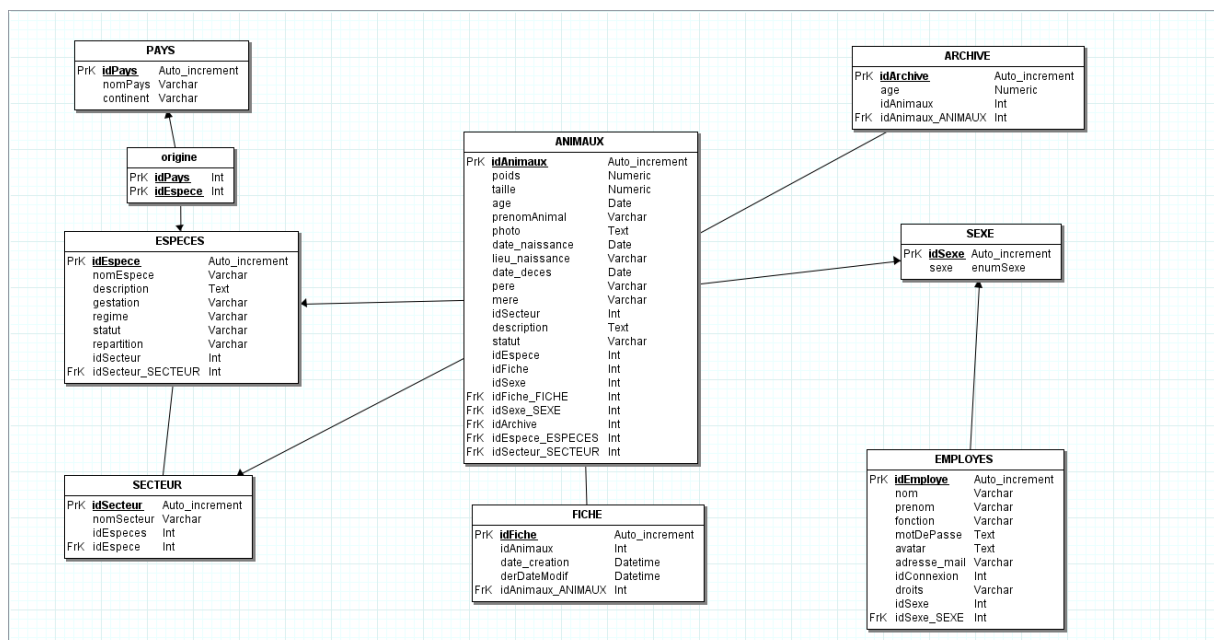
7 : ANIMAUX			
Enregistrements	Entité	Propriété/longueur	
idAnimaux	ANIMAUX	Auto-Incrémenter	11
poids	ANIMAUX	Numeric	5
taille	ANIMAUX	Numeric	5
age	ANIMAUX	Numeric	3
date_naissance	ANIMAUX	Date	10
lieu_naissance	ANIMAUX	Varchar	20
date_deces	ANIMAUX	Date	10
prenomAnimal	ANIMAUX	Varchar	40
pere	ANIMAUX	Varchar	20
mere	ANIMAUX	Varchar	20
idSecteur	ANIMAUX	Int	11
description	ANIMAUX	Text	255
statut	ANIMAUX	Varchar	15
idEspece	ANIMAUX	Int	11
photo	ANIMAUX	Text	25
idFiche	ANIMAUX	Int	11
idSexe	ANIMAUX	Int	11
8 : FICHE			
Enregistrements	Entité	Propriété/longueur	
idFiche	FICHE	Auto-Incrémenter	11
idAnimaux	FICHE	Varchar	11
date_creation	FICHE	Datetime	15
derDateModif	FICHE	Datetime	15

9 : ARCHIVE			
Enregistrements	Entité	Propriété/longueur	
idArchive	ARCHIVE	Auto-Incrémenter	25
age	ARCHIVE	Numeric	30
idAnimaux	ARCHIVE	Int	11

MCD AVEC JMERISE :



MLD AVEC JMERISE :



SQL AVEC JMERISE :

```
#-----  
#      Script MySQL.  
#-----  
  
#-----  
# Table: EMPLOYES  
#-----  
  
CREATE TABLE EMPLOYES(  
  idEmploye      int (11) Auto_increment NOT NULL ,  
  nom            Varchar (40) ,  
  prenom         Varchar (40) ,  
  fonction       Varchar (40) ,  
  motDePasse     Text NOT NULL ,  
  avatar         Text NOT NULL ,  
  adresse_mail   Varchar (40) ,  
  idConnexion    Int NOT NULL ,  
  droits         Varchar (3) NOT NULL ,  
  idSexe         Int NOT NULL ,  
  idSexe_SEXE    Int NOT NULL ,  
  PRIMARY KEY (idEmploye )  
)ENGINE=InnoDB;  
  
#-----  
# Table: ANIMAUX  
#-----  
  
CREATE TABLE ANIMAUX(  
  idAnimaux      int (11) Auto_increment NOT NULL ,  
  poids          Numeric ,  
  taille         Numeric ,  
  age            Date ,  
  prenomAnimal   Varchar (40) ,  
  photo          Text NOT NULL ,  
  date_naissance Date ,  
  lieu_naissance Varchar (20) ,  
  date_deces     Date ,  
  pere           Varchar (20) ,  
  mere           Varchar (20) ,  
  idSecteur      Int NOT NULL ,  
  description    Text ,  
  statut         Varchar (15) ,  
  idEspece       Int NOT NULL ,  
  idFiche        Int NOT NULL ,  
  idSexe         Int NOT NULL ,  
  idFiche_FICHE  Int NOT NULL ,  
  idSexe_SEXE    Int NOT NULL ,  
  )
```

```

        idFiche_FICHE      Int NOT NULL ,
        idSexe_SEXE       Int NOT NULL ,
        idArchive         Int NOT NULL ,
        idEspece_ESPECES  Int NOT NULL ,
        idSecteur_SECTEUR Int NOT NULL ,
        PRIMARY KEY (idAnimaux )
    )ENGINE=InnoDB;

#-----
# Table: ARCHIVE
#-----

CREATE TABLE ARCHIVE(
    idArchive      int (11) Auto_increment NOT NULL ,
    age            Numeric ,
    idAnimaux      Int NOT NULL ,
    idAnimaux_ANIMAUX Int NOT NULL ,
    PRIMARY KEY (idArchive )
)ENGINE=InnoDB;

#-----
# Table: SECTEUR
#-----

CREATE TABLE SECTEUR(
    idSecteur      int (11) Auto_increment NOT NULL ,
    nomSecteur     Varchar (25) ,
    idEspeces      Int NOT NULL ,
    idEspece       Int NOT NULL ,
    PRIMARY KEY (idSecteur )
)ENGINE=InnoDB;

#-----
# Table: ESPECES
#-----

CREATE TABLE ESPECES(
    idEspece       int (11) Auto_increment NOT NULL ,
    nomEspece      Varchar (40) ,
    description     Text ,
    gestation      Varchar (25) ,
    regime         Varchar (40) ,
    statut         Varchar (25) ,
    repartition    Varchar (50) ,
    idSecteur      Int NOT NULL ,
    idSecteur_SECTEUR Int NOT NULL ,

```

```

CREATE TABLE ESPECES(
    idEspece          int (11) Auto_increment NOT NULL ,
    nomEspece         Varchar (40) ,
    description        Text ,
    gestation          Varchar (25) ,
    regime             Varchar (40) ,
    statut             Varchar (25) ,
    repartition        Varchar (50) ,
    idSecteur          Int NOT NULL ,
    idSecteur_SECTEUR Int NOT NULL ,
    PRIMARY KEY (idEspece )
)ENGINE=InnoDB;

```

```

#-----
# Table: SEXE
#-----

```

```

CREATE TABLE SEXE(
    idSexe int (11) Auto_increment NOT NULL ,
    sexe   Enum ("Masculin","Feminin") NOT NULL ,
    PRIMARY KEY (idSexe )
)ENGINE=InnoDB;

```

```

#-----
# Table: FICHE
#-----

```

```

CREATE TABLE FICHE(
    idFiche          int (11) Auto_increment NOT NULL ,
    idAnimaux         Int NOT NULL ,
    date_creation     Datetime ,
    derDateModif      Datetime ,
    idAnimaux_ANIMAUX Int NOT NULL ,
    PRIMARY KEY (idFiche )
)ENGINE=InnoDB;

```

```

#-----
# Table: PAYS
#-----

```

```

CREATE TABLE PAYS(
    idPays          int (11) Auto_increment NOT NULL ,
    nomPays         Varchar (40) ,
    continent        Varchar (40) ,
    PRIMARY KEY (idPays )
)ENGINE=InnoDB;

```

```

#-----
# Table: origine
#-----

```

```

CREATE TABLE origine(
    idPays Int NOT NULL ,
    idEspece Int NOT NULL ,
    PRIMARY KEY (idPays ,idEspece )
)ENGINE=InnoDB;







```

```

ALTER TABLE EMPLOYES ADD CONSTRAINT FK_EMPLOYES_idSexe_SEXE FOREIGN KEY (idSexe_SEXE) REFERENCES SEXE(idSexe);
ALTER TABLE ANIMAUX ADD CONSTRAINT FK_ANIMAUX_idFiche_FICHE FOREIGN KEY (idFiche_FICHE) REFERENCES FICHE(idFiche);
ALTER TABLE ANIMAUX ADD CONSTRAINT FK_ANIMAUX_idSexe_SEXE FOREIGN KEY (idSexe_SEXE) REFERENCES SEXE(idSexe);
ALTER TABLE ANIMAUX ADD CONSTRAINT FK_ANIMAUX_idArchive FOREIGN KEY (idArchive) REFERENCES ARCHIVE(idArchive);
ALTER TABLE ANIMAUX ADD CONSTRAINT FK_ANIMAUX_idEspece_ESPECES FOREIGN KEY (idEspece_ESPECES) REFERENCES ESPECES(idEspece);
ALTER TABLE ANIMAUX ADD CONSTRAINT FK_ANIMAUX_idSecteur_SECTEUR FOREIGN KEY (idSecteur_SECTEUR) REFERENCES SECTEUR(idSecteur);
ALTER TABLE ARCHIVE ADD CONSTRAINT FK_ARCHIVE_idAnimaux_ANIMAUX FOREIGN KEY (idAnimaux_ANIMAUX) REFERENCES ANIMAUX(idAnimaux);
ALTER TABLE SECTEUR ADD CONSTRAINT FK_SECTEUR_idEspece FOREIGN KEY (idEspece) REFERENCES ESPECES(idEspece);
ALTER TABLE ESPECES ADD CONSTRAINT FK_ESPECES_idSecteur_SECTEUR FOREIGN KEY (idSecteur_SECTEUR) REFERENCES SECTEUR(idSecteur);
ALTER TABLE FICHE ADD CONSTRAINT FK_FICHE_idAnimaux_ANIMAUX FOREIGN KEY (idAnimaux_ANIMAUX) REFERENCES ANIMAUX(idAnimaux);
ALTER TABLE origine ADD CONSTRAINT FK_origine_idPays FOREIGN KEY (idPays) REFERENCES PAYS(idPays);
ALTER TABLE origine ADD CONSTRAINT FK_origine_idEspece FOREIGN KEY (idEspece) REFERENCES ESPECES(idEspece);

```

TABLEAU EXCEL SINGE FICHE INDIVIDUEL :

photo	nom de la photo	especies	idAnimaux	poids (kgs)	taille (cms)	age	date_naissance	lieu_naissance	date_deces	prenomAnimal	pere	mere	idSecteur	
	Babouin-male-Ethiopie.jpg	BABOUIN	1	33	102	9 ans	7/5/2007	Zoo Ethiopie		Patio	Simba	Olive	Territoire 1	A e a
	Babouine-Afrique-australe.jpg	BABOUIN	2	12	54	8 ans	4/3/2008	Zoo Afrique Australe		Léonne	Pedor	Sriba	Territoire 1	F e n
	Bebe-babouin.jpg	BABOUIN	3	2	25	6 mois	8/27/2016	Zoo		Franzie	Patio	Léonne	Territoire 1	L v c
	babouin.jpg	BABOUIN	4	37	98	7 ans	10/15/2009	Zoo		Hanon	Heyron	Iris	Territoire 1	A a
	Bonobo-bebe.png	BONOBO	5	1	25	1 mois	12/25/2016	Zoo		Suzy	Arthur	Horine	Territoire 2	P n
	Bonobo_male.jpg	BONOBO	6	42	115	15 ans	9/27/2002	Zoo Apeldoorn		Arthur	—	—	Territoire 2	A n
	Bonobo-femelle.jpg	BONOBO	7	34	111	12 ans	5/3/2005	Zoo		Horine	Philo	Coria	Territoire 2	E n

idSecteur	description	statut	idEspece
Territoire 1	Male de 9 ans ayant un caractère assez fort et voulant toujours faire sa loi, attention avec les petits	preter	10
Territoire 1	Femelle de 8 ans venant de mettre bas, elle est très protectrice avec son petit et serait même agressive avec le male	preter	10
Territoire 1	Le petit est né avec un pelage noir, son visage et ses oreilles étaient bien roses. Il commence à se débrouiller tout seul	vivant	10
Territoire 1	Male très doux et craintif, c'est très bien fait adopté par le clan	vivant	10
Territoire 2	Petite femelle nourrit au biberon car la mère refuse de la nourrir	vivant	11
Territoire 2	Male preter le temps de la reproduction mais il est très docile	preter	11
Territoire 2	Elle n'a pas du tout la fibre maternel et refuse de s'occuper de son bébé	vivante	11

FICHES ESPECES :

idEspece	nomEspece	description	Gestation	Régime	Statut	repartition
1	Babouin	Nom : Papio cynocephalus Famille : Cercopithecidae Identification : Museau de chien, se déplace essentiellement à quatre pattes ; mâles deux fois plus gros que les femelles ; pelage vert-jaune Taille : De 55 à 84 cm (mâles) ; de 36 à 60 cm (femelles) ; queue : de 53 à 66 cm (mâles) de 35 à 56 cm (femelles) Poids : De 20 à 25 kg (mâles), de 8 à 16 kg (femelles) Habitat : Forêts ouvertes, savanes arbustives et arborées, steppes, près des forêts-galeries et des collines rocheuses ; domaine vital moyen de 50 km2 Organisation sociale : Troupe mixte de 10 à 150 animaux Nombre de jeunes par portée : 1 (rarement des jumeaux) Espérance de vie : De 30 à 45 ans Effectifs : Abondants mais inconnus précisément ; stables Statut : Espèce inscrite en Annexe II de la Cites (Convention sur le commerce international des espèces menacées d'extinction).	180	Omnivore	Menacée	Soudan à l'Afrique occidentale
2	Bonobo	Nom : Pan paniscus Famille : Hominidés Identification : Museau de chien, se déplace essentiellement à quatre pattes ; mâles deux fois plus gros que les femelles ; pelage vert-jaune Taille : De 55 à 84 cm (mâles) ; de 36 à 60 cm (femelles) ; queue : de 53 à 66 cm (mâles) de 35 à 56 cm (femelles) Poids : De 20 à 25 kg (mâles), de 8 à 16 kg (femelles) Habitat : Forêts ouvertes, savanes arbustives et arborées, steppes, près des forêts-galeries et des collines rocheuses ; domaine vital moyen de 50 km2 Organisation sociale : Troupe mixte de 10 à 150 animaux Nombre de jeunes par portée : 1 (rarement des jumeaux) Espérance de vie : De 30 à 45 ans	180	Omnivore	Menacée	République démocratique du Congo

3	Capucin	Nom : Cebus capucinus Famille : Cebidae Identification : Le visage est généralement rose et la quantité de fourrure blanche qui le recouvre est variable selon les classes d'âge et de sexe. Taille : entre 33,5 à 45,3 cm et sa queue mesure entre 35,0 et 55,1 cm Poids : Les femelles pèsent en moyenne 2,67 kg alors que les mâles atteignent un poids moyen de 3,87 kg Habitat : Il se trouve dans les forêts tropicales humides et sèches, les forêts inondées, les forêts de mangrove, les forêts-galeries, ainsi que dans les forêts sèches à feuilles caduques où les averse sont absentes 5 à 6 mois de l'année. Répartition : Sa distribution géographique s'étend de l'extrême nord-ouest de l'Équateur jusqu'au Honduras, en passant par l'ouest de la Colombie, le Panama, le Costa Rica ainsi que le Nicaragua. Organisation sociale : La taille moyenne d'un groupe est de 16 individus Nombre de jeunes par portée : 1 (rarement des jumeaux) Espérance de vie : 10 ans Statut : Espèce inscrite en Annexe II de la Cites (Convention sur le commerce international des espèces menacées d'extinction)	160	Omnivore	Menacée	Nord-ouest de l'Equateur Honduras Ouest de la Colombie Panama Costa Rica Nicaragua
4	Chimpanzé	Nom : Pan Famille : Hominidés Identification : Museau de chien, se déplace essentiellement à quatre pattes ; mâles deux fois plus gros que les femelles ; pelage vert-jaune Taille : De 55 à 84 cm (mâles) ; de 36 à 60 cm (femelles) ; queue : de 53 à 66 cm (mâles) de 35 à 56 cm (femelles) Poids : De 20 à 25 kg (mâles), de 8 à 16 kg (femelles) Habitat : Forêts ouvertes, savanes arbustives et arborées, steppes, près des forêts-galeries et des collines rocheuses ; domaine vital moyen de 50 km2 Organisation sociale : Troupe mixte de 10 à 150 animaux	180	Omnivore	Menacée	République démocratique du Congo Afrique de l'Ouest Afrique centrale Afrique de l'Est

REGLES DE CODAGE ET DE NOMMAGE :

Veille technologique

Nous avons utilisé plusieurs langage au cours de notre projet qui sont le HTML, CSS, PHP et le SQL :



- HTML : Ces origines remonte aux années 1989-1992, depuis 1997 avec HTML 4 son évolution est ralentie mais 10 ans plus tard HTML 4 reste toujours utilisé. En 2008 HTML 5 est à l'étude. En 2011 le html living standard a pour but d'inclure le html 5 et aussi de se développer sans cesse.



- CSS : Dave RAGGET et Hakon LIE modifie en 1995 le navigateur Arena et font une démonstration afin de tester leur langage. Bery BOS qui avait son propre projet SSP (Stream Based Style Sheet Proposal). Ce dernier se lie à Hakon LIE, SSP ayant été développé avec l'idée d'être applicable non seulement à HTML, mais aussi à d'autre langage de balisage. Bert BOS et Hakon LIE reprennent cet objectif et en font une des caractéristiques clés de ce qui devient alors CSS à proprement parlé.



- PHP : Le PHP a été créé en 1994 par Rasmus LERDORF pour son site web (bibliothèque de logiciel en C) qui servir à conserver une trace des visiteurs qui venaient consulter son cv. Avec le temps il y ajoute des fonctionnalités. Rasmus a transformé la bibliothèque en une implémentation capable de communiquer avec des bases de données et de créer des applications dynamiques et simples pour le [Web](#). En 1995 il publie son code qui s'appelait encore PHP/FI (Personal Home Page Tools/Form Interpreter) pendant un an 2 étudiant travaillent sur le cœur de PHP/FI, depuis la version 3 le nom change pour PHP.



- SQL : Créé en 1974 mais normalisé en 1986 (Structured Query Language) en français langage de requête structurée) le langage est reconnu par la grande majorité des systèmes de gestion de bases de données relationnelles (SGBDR). Langage servant à exploiter les SGBDR. Les différentes requêtes permettent d'ajouter, rechercher, modifier ou de supprimer les données de la base.



- Java script : Langage créé en 10 jours par Brendan Eich en mai 1995 pour la Netscape communications corporation, langage inspiré par de nombreux langage dont JAVA tout en simplifiant son écriture mais sa sortie est annoncé en décembre 1995 par SUN et Netscape. JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de NODE.J.S La base de ce langage c'est que les principales interfaces sont fournies par des objets.



Smarty est un moteur de template (modèle) open-source destiné à faciliter et même accélérer le travail du développeur.

Il permet de séparer l'application (logique) et la présentation (interface) ; ainsi il facilite le travail à plusieurs, et aide à améliorer la clarté des scripts, et nous gratifie de plus de fonctionnalités puissantes, contenues dans sa bibliothèque.

Exemple, ici un formulaire html classique :

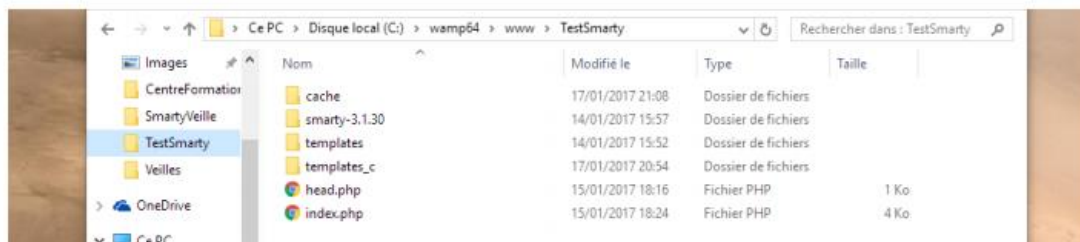
```
<select name="Date_Month">
<option value="01" selected="selected">January</option>
Février, Mars ... (12 balises)
<option value="12">December</option>
</select>
<select name="Date_Day">
<option value="1">01</option>
2, 3, 4, 5 ... (31 balises)
<option value="31">31</option>
</select>
<select name="Date_Year">
<option value="2017" selected="selected">2017</option>
</select>
```

La même chose, en utilisant une classe Smarty :

```
{html_select_date}
```

Installation de base :

- Télécharger Smarty (compressé)
- L'extraire dans localhost (../wamp64/www/)
- En plus du dossier Smarty, on crée obligatoirement à la racine :
 - o Un dossier « cache »
 - o Un dossier « template » qui contient des docs dont l'extension sera **.tpl** (Vue)
Le nom « template » est le nom par défaut
 - o Un dossier « template_c » qui contient les fichiers compilés
Le nom « template_c » est également le nom par défaut
- Nouveau document texte « **head.php** »
- Nouveau document texte « **index.php** »



Une fois le répertoire prêt, il nous faut « enclencher » la machine Smarty, pour cela, dans le « head » (cerveau de Smarty), on fait appel à **Smarty.class.php**, contenu dans le sous-dossier « libs » du dossier « smarty », on crée une nouvelle instance Smarty à laquelle on va donner un nom, puis assigner les deux dossiers « template » cités dans l'installation de base.

```
head.php  x  index.php  x
1 <?php
2 require_once('smarty-3.1.30/libs/Smarty.class.php');
3
4 //Génère une nouvelle instance "objet" Smarty
5 $smarty = new Smarty();
6 //Repertoire des vues
7 $smarty->template_dir = 'templates';
8 //Repertoire des compilations
9 $smarty->compile_dir = 'templates_c';
10 ?>
```

Il est possible d'ajouter d'autres dossiers et paramètres, ces 4 lignes sont cependant impératives.

On peut ensuite commencer à utiliser le moteur, et donc à travailler séparément le traitement et le rendu visuel grâce au nouvel objet **\$smarty** généré dans **head.php**.

On peut ainsi créer un fichier index.php (par exemple), qui fera appel à son tout début au « cerveau ». (**head.php**) et contiendra les variables Smarty que l'on pourra utiliser dans nos templates en leur assignant des variables PHP préalablement déclarées et traitées.

Voici une application du principe de Smarty des plus simplistes, contenant une simple variable « titre » :

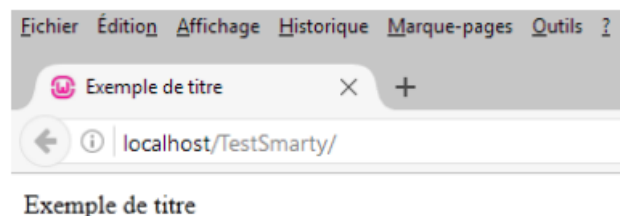
```
1 <?php
2 //Liaison vers le "cerveau" (head.php) de Smarty
3 require('head.php');
4
5 //Variable titre
6 $titre = "Exemple de titre";
7
8 /*Assignment de la variable PHP $titre à la variable smarty "titre"
9  dans 'index.tpl', on fait appel à cette variable comme suit : {$titre}
10  pour afficher son contenu*/
11 $smarty->assign('titre',$titre);
12
13 // Pour afficher les objets assignés dans un modèle : ->display('nomDuTemplate')
14 $smarty->display('index.tpl');
15 ?>
```

Dans le dossier « **templates** », le fichier **index.tpl** que l'on veut afficher
commande `$varPhp->display('page.tpl')` :



```
1 <html>
2 <head>
3   <title>{$titre}</title>
4 </head>
5 <body>
6   {$titre}
7 </body>
8 </html>
9
```

Le résultat saisissant :



Notez la parfaite compatibilité avec Mozilla Firefox ^^

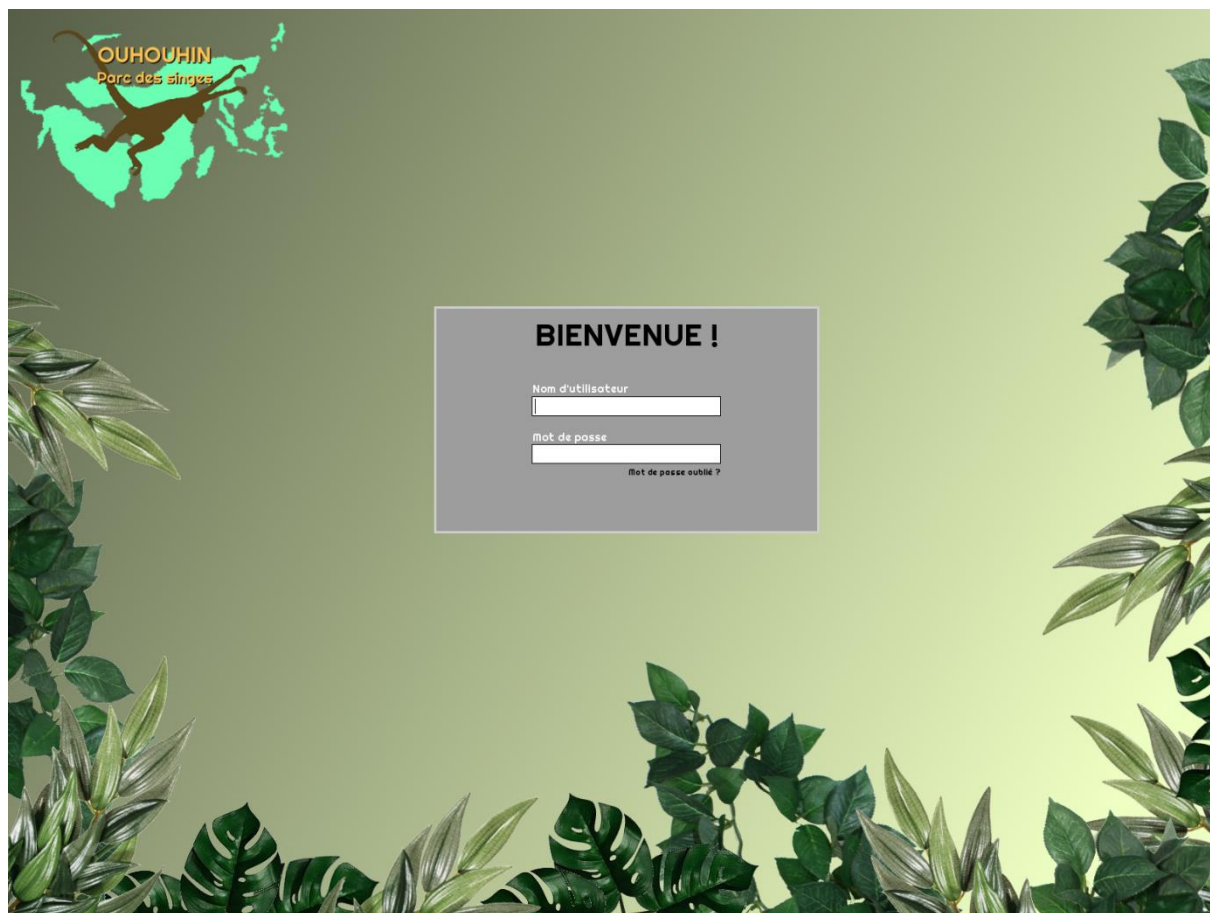
FEUILLAGE POUR FAIRE LA MAQUETTE :



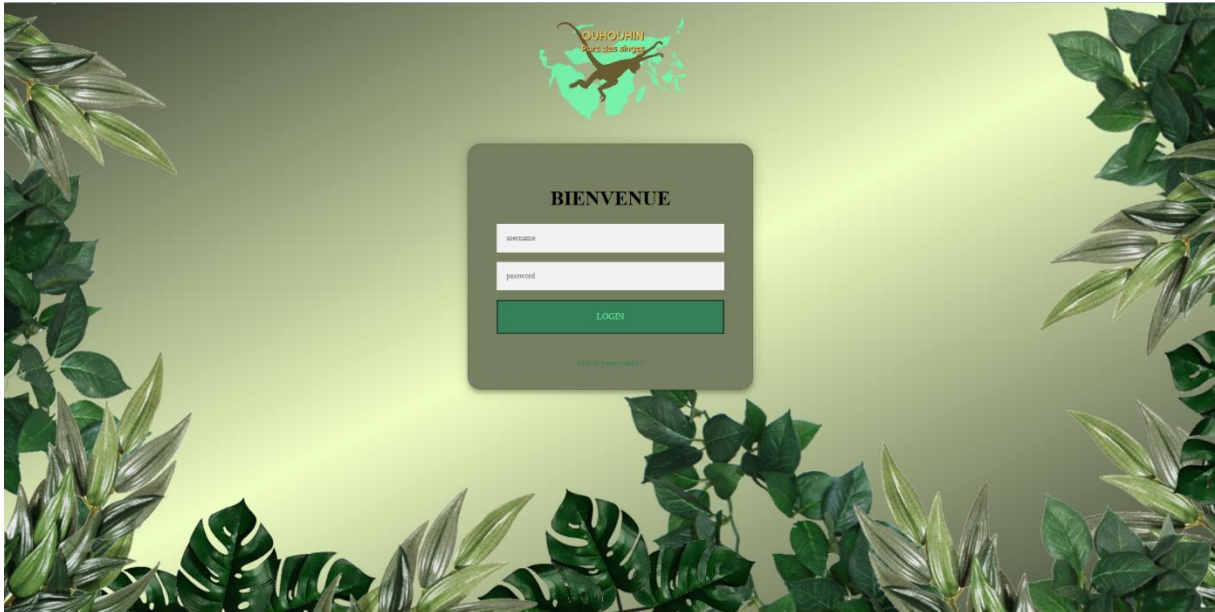
LOGOSINGE POUR LE PROJET :



MAQUETTE DE LA PAGE D'ACCUEIL :





PAGE DE LOGIN :



DEBUT DE TABLE EMPLOYES :

























































idEmploye	idConnexion <small>identifiant de connexion</small>	nom	prenom	fonction	motDePasse	avatar	adresse_mail	idSexe
1	vlacroix	Lacroix	Valérie	Administrateur	rbtqe6qV	userlogof.png	vlacroix@p2zoo.fr	2
2	ecourtemanche	Courtemanche	Eugène	Chef Animalier	qKJfWoB	userlogom.jpg	ecourtemanche@p2zoo.fr	1


DEBUT EMPLOYES VUE WEB :

	Non	Prenom	Fonction	Action		
	Lacroix	Valérie	Administrateur	<input type="button" value="C"/>	<input type="button" value="M"/>	<input type="button" value="S"/>
	Courtemanche	Eugène	Chef Animalier	<input type="button" value="C"/>	<input type="button" value="M"/>	<input type="button" value="S"/>

LISTE DES EMPLOYES AVEC AVATAR :

Liste des utilisateurs

	Nom	Prénom	Fonction	Action		
	Lacroix	Valérie	Admin			
	Courtemanche	Eugène	Chef Animalier			
	Poisson	Tyson	Vétérinaire			
	Bier	Théodore	Vétérinaire			
	Béland	Océane	Stagiaire			
	Avare	Clément	Stagiaire			
	Poirier	Charles	Stagiaire			
	Gaulin	Arianne	Stagiaire			
	Théberge	Suzette	Stagiaire			
	Desjardins	Marjolaine	Stagiaire			
	Duhamel	Mirabelle	Stagiaire			
	Lampron	Jérôme	Stagiaire			
	Couet	Angélette	Stagiaire			
	Couttes	Angélettess	Stagiaire			

Ajout d'un employé 

Ajout d'un animal 

MODELE DE FICHE JOURNALIERE :

Projet2 : Compte rendu du (date)

Participant :

Temps passé :

Tache :

- Tache 1

Numéro de la tache	Intitulé	Bilan

Information complémentaire