

SPECIFICATIONS LOGICIELLES

PROJET EQUIDA

VERSION :

3.0

DATE DE DERNIERE MISE A JOUR :

01/11/2019

AUTEURS :

PEAKY BLINDERS

Table des matières

1.HISTORIQUE DES MODIFICATIONS DU DOCUMENT	3
2.INTRODUCTION	4
2.1.Présentation générale de l'application	4
2.2.Documents de référence	4
2.3.Spécifications générales	4
2.4.Contraintes d'évolution	4
3.SPECIFICATIONS FONCTIONNELLES	4
3.1.Définition des acteurs	4
3.2.Cas d'utilisation	4
4.SPECIFICATIONS D'INTERFACES	4
4.1.Structure des pages	4
4.2.Liste des pages	5
4.3.Cinématique	5
4.4.Interfaces utilisateurs	5
4.4.1.Interface X	5
4.5.Interfaces avec d'autres applications	5
4.6.interfaces avec des fichiers	5
5.SPECIFICATIONS TECHNIQUES	5
5.1.Environnement matériel	5
5.2.Environnement logiciel	5
5.3.Mise en œuvre	6
5.4.Exigences de programmation	6
6.MODELISATION	6
6.1.Modèle relationnel et base de données	6
6.2.Diagramme de classes	6
7.ARCHITECTURE APPLICATIVE	6
8.QUALIFICATION DU LOGICIEL	6
9.SAUVEGARDE ET RESTAURATION DE LA BASE ET DE L'APPLI	6

1. HISTORIQUE DES MODIFICATIONS DU DOCUMENT

Date	Auteur	Version	Sujet de la modification
05/09/16	Zakina Annouche	1.0	Création
02/09/2018	Zakina Annouche	2.0	Dissociation env de production / env de dev correction de fautes d'orthographe
01/09/2019	Zakina Annouche	3.0	Simplification du document
29/10/2019	Quentin Lalin	3.1	Ajout des paragraphes 2.1, 2.3, 4.1
30/10/2019	Thomas Leconte	3.2	Modification des paragraphes 3 et 7
30/10/2019	Metz Julien	3.3	Ajout 2.2, 2.4, 4.3, 4.5, 4.6

 Pensez à l'Environnement avant d'imprimer ce fichier

2. INTRODUCTION

2.1. Présentation générale de l'application

Ce paragraphe présente l'application dans son contexte

La société Equida, spécialisée dans la vente aux enchères de chevaux de course, possède de nombreux clients à travers le monde qui vont du simple propriétaire aux haras. Ayant des correspondant des ces divers pays afin de se rapprocher de sa clientèle étrangère, l'utilisation d'un site web et de diverses applications permettent une proximité plus grande encore. C'est pourquoi le site d'Equida est une importante plu valu pour cette société.

2.2. Documents de référence

Ce paragraphe liste tous les documents applicables au projet (cahier des charges, spécifications, normes de développement (y compris les normes sur le net W3C...), document de livraison, d'exploitation, document de tests, plan d'assurance qualité, etc.) .

Document	Fonction	Emplacement
Cahier des charges	Décrit les besoins des utilisateurs	NAS://PROJETS/EQUIDAPEAKY/CDC
Spécifications logicielles	Décrit fonctionnellement et techniquement l'application	NAS://PROJETS/EQUIDAPEAKY/SPEC
Plan de tests	Liste tous les tests fonctionnels et techniques à réaliser	NAS://PROJETS/EQUIDAPEAKY/PLANTEST
Document de livraison	Décrit la procédure de livraison de l'application	NAS://PROJETS/EQUIDAPEAKY/DOC
Document d'exploitation	Décrit l'exploitation courante de l'application et notamment toutes les procédures de sauvegarde/restauration	NAS://PROJETS/EQUIDAPEAKY/LIVRABLE

2.3. Spécifications générales

*Ce paragraphe décrit sommairement et en quelques lignes :
les fonctionnalités du logiciel. Il s'agit de donner un premier aperçu sans rentrer dans le détail
l'environnement d'utilisation et d'exécution du logiciel.*

Le site web d'Equida permet, en se connectant via un identifiant et un mot de passe, d'avoir la liste des chevaux, des courses et des ventes. De pouvoir consulter ses informations personnelles ainsi que celles de ses chevaux. Il est également possible d'ajouter un cheval à une vente. Ajout qui sera accepté ou non par la suite par un employé d'Equida.

2.4. Contraintes d'évolution

Ce paragraphe précise si le logiciel est développé en prévoyant de le faire évoluer pour le réutiliser dans des applications similaires, ou si des évolutions sont d'ores et déjà prévues.

Le logiciel est développé selon le cahier des charges et les besoins du client Equida mais il est tout à fait capable de réutiliser, modifier et améliorer l'application pour des applications similaires. Des améliorations sont prévues comme lister les éléments archivés et pouvoir les désarchiver.

3. SPECIFICATIONS FONCTIONNELLES

3.1. Définition des acteurs

Les acteurs seront donc les suivants :

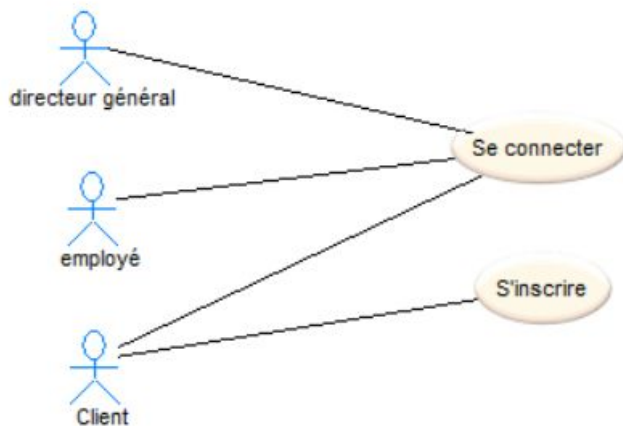
- Le client,
- L'employé Equida (considéré comme un simple employé / directeur commercial),
- Le directeur général (Administrateur Equida).

3.2. Cas d'utilisation

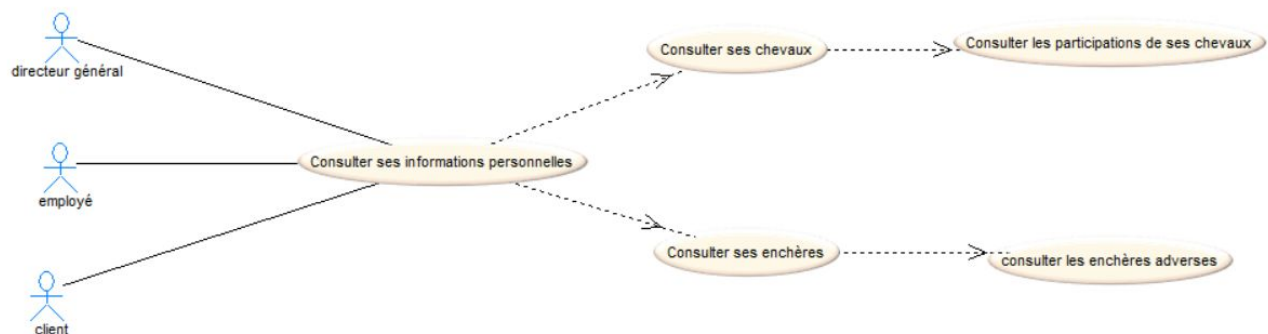
Les cas d'utilisation pourront être organisés par fonctionnalités ou exigences.

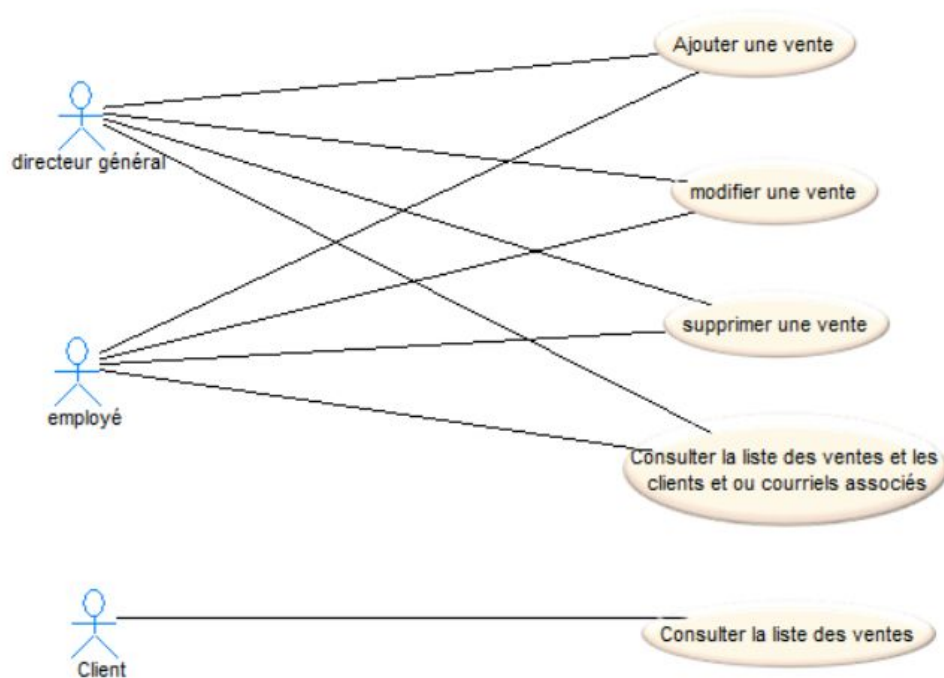
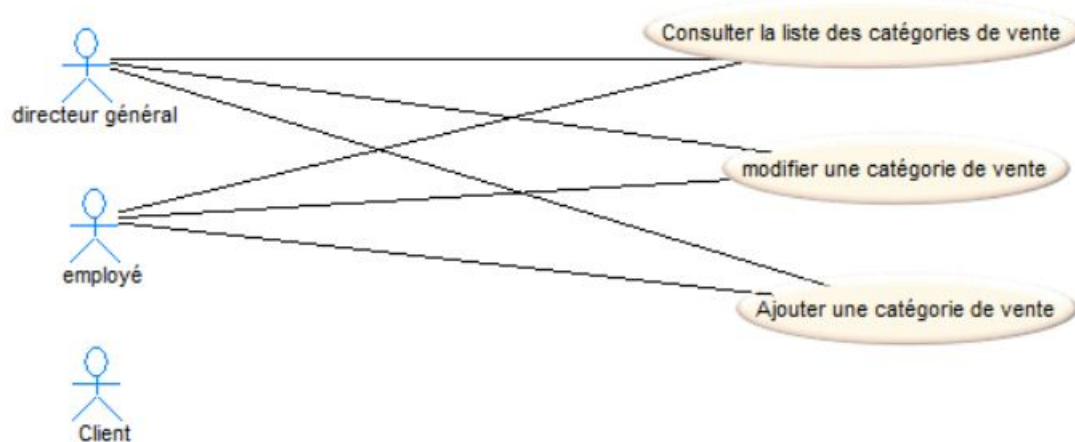
Les cas d'utilisation les plus complexes seront détaillés (scénario et/ou diagramme de séquences).

Se connecter / s'inscrire :

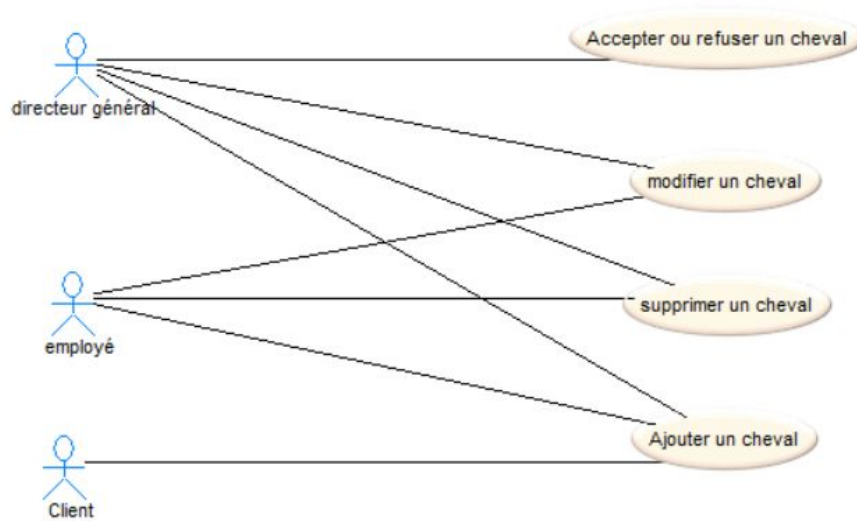


Consulter ses informations personnelles :

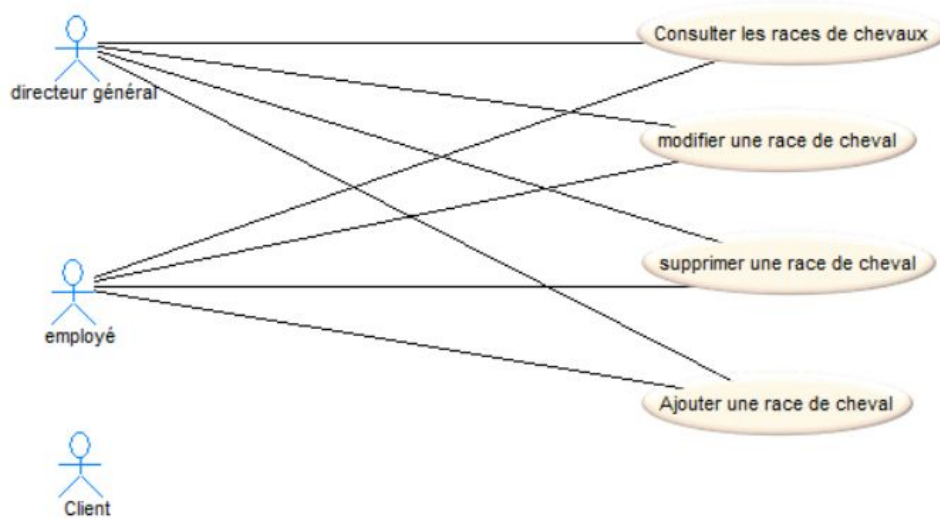


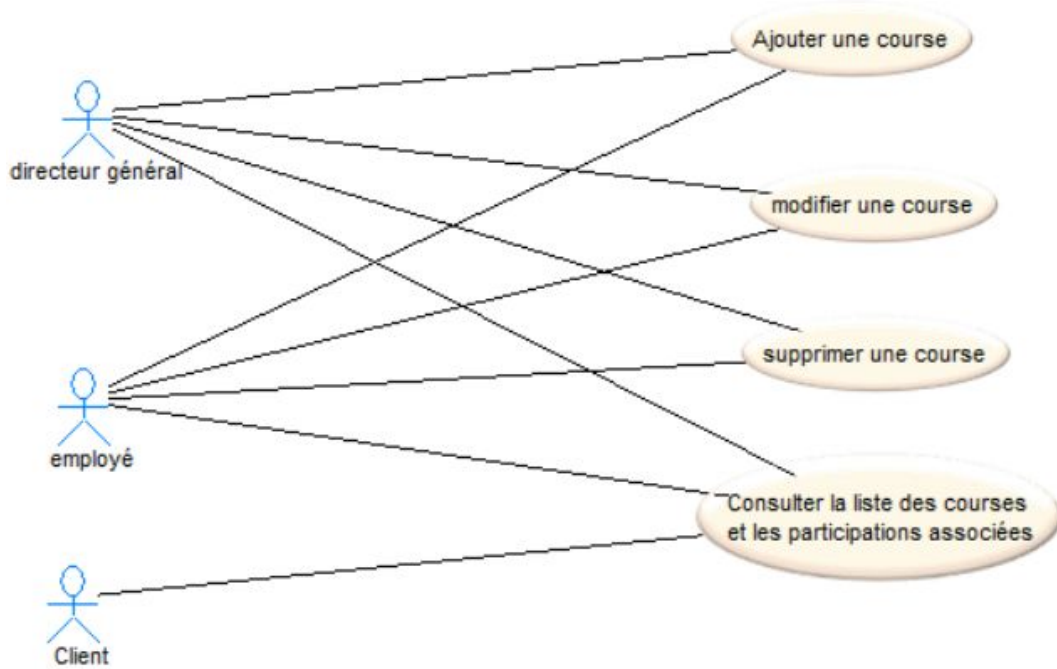
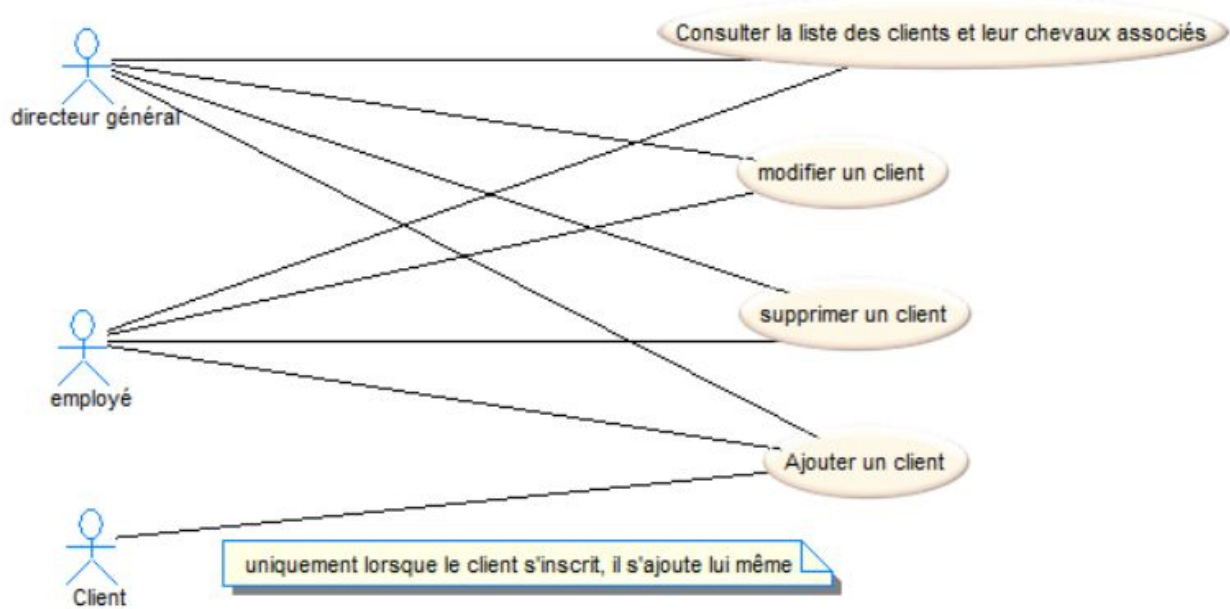
Consulter / ajouter / modifier / supprimer une vente :Consulter / ajouter / modifier une catégorie de vente :

On ne laisse possibilité à quiconque de supprimer une catégorie de vente car il faudrait également supprimer les ventes associées, avec leur lot, enchères, lieux et chevaux associés, ce qui rendrait presque inutilisable l'application de part ce manque de données.

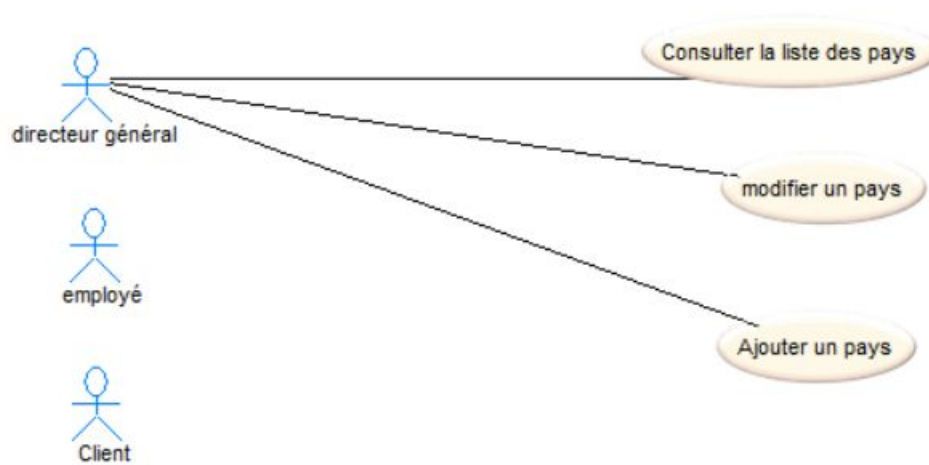
Consulter / ajouter / modifier / supprimer un cheval :

Lors de l'ajout d'un cheval de la part d'un client, d'un employé ou d'un administrateur, l'administrateur lui-même doit valider ou refuser ce cheval, depuis un lien cliquable à l'accueil qui apparaît uniquement lorsqu'un cheval est inscrit.

Consulter / ajouter / modifier / supprimer une race de cheval :

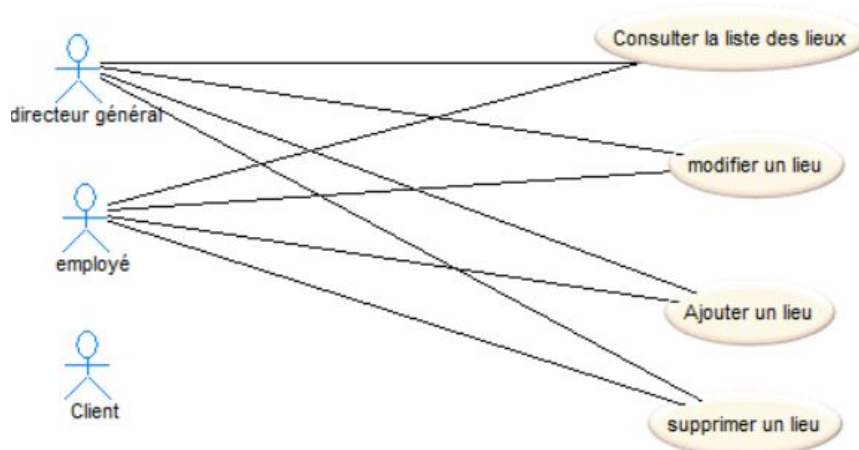
Consulter / ajouter / modifier / supprimer une course :Consulter / ajouter / modifier / supprimer un client :

Consulter / ajouter / modifier un pays :



On ne laisse pas la possibilité de supprimer un pays à qui que ce soit puisque tous les clients sont rattachés dessus, et les supprimer également supprimerait une très grande partie des données de l'application, qui n'aurait plus grand intérêt.

Consulter / ajouter / modifier / supprimer un lieu :



4. SPECIFICATIONS D'INTERFACES

4.1. Structure des pages

Ce paragraphe a pour objet de décrire les éléments communs à toutes les pages (entête, pied-de-page, menu de navigation, etc...).

En fonction du type de compte connecté (client, employé, admin), le menu de navigation sera différent mais restera le même sur toutes les pages auxquelles le compte a accès. L'entête et l'aspect général du site restera également la même partout (excepté la page de connexion).

4.2. Liste des pages

Ce paragraphe liste toutes les pages nécessaires à l'application

Interface	Fichier	Accès	Fonction
Connexion	connexion.jsp	Par le lien s'authentifier ou le lien déconnexion	Connexion à l'application
Accueil	Accueil.jsp
Liste des derniers ouvrages	Index.jsp	Par l'url de l'application	Liste les derniers ouvrages enregistrés. Il s'agit de la page d'accueil

Interface	Fichier	Accès	Fonction
Client	clientConsulter.jsp		consulter la liste des clients intéressés par une vente
client	clientAjouter.jsp		ajouter un client
employé	employé.html		page des ventes propre aux employés
client	client.html		page des ventes propre aux clients
admin	admin.html		page des ventes propre aux admins
course	ajouterCourse.jsp		permet d'ajouter une course
course	listerLesCourses.jsp		affiche la liste des courses
course	listerLesParticipations.jsp		affiche la liste des participants à une course

course	modifierCourse.jsp		Permet de modifier une course
course	supprimerCourse.jsp		Permet de supprimer une course
client	clientConsulter.jsp		affiche la liste de tous les clients
client	infoClient.jsp		Affiche les informations du client sélectionné
client	modifierClient.jsp		permet de modifier un client
client	supprimerClient.jsp		permet de supprimer un client
cheval	AccepterOuRefuserCheval.jsp		Permet aux employés d'accepter ou refuser un cheval pour les ventes
cheval	ajouterCheval.jsp		permet d'ajouter un nouveau cheval
cheval	ajouterTypeCheval.jsp		permet d'ajouter une nouvelle race de cheval
cheval	ajouterUneParticipation.jsp		permet d'ajouter un cheval dans une course
cheval	infosCheval.jsp		permet de consulter les informations d'un cheval

cheval	listerLesChevaux.jsp		affiche la liste des chevaux
cheval	listerLesChevauxProposes.jsp		affiche la liste des chevaux proposés à la vente
cheval	listerLesTypesChevaux.jsp		affiche la liste de toutes les races de chevaux
cheval	modifierCheval.jsp		permet de modifier un cheval
cheval	supprimerCheval.jsp		permet de supprimer un cheval
cheval	modifierTypesCheval.jsp		permet de modifier une race de cheval

cheval	supprimerTypesCheval.jsp		permet de supprimer une race de cheval
--------	--------------------------	--	--

4.3. Cinématique

Ce paragraphe décrit l'enchaînement des pages dans un contexte d'utilisation normale et anormale (messages d'erreurs, etc.).

Dans le cadre d'une utilisation client : Un client arrive sur la page de connexion. Le client ne peut se connecter que s'il rentre ses informations (identifiant/mot de passe) sinon il ne peut se connecter. Grâce à la barre de navigation, une fois connecté, il peut avoir accès à ses informations générales(données du client), les données des ses chevaux et ses enchères. Il a aussi accès à la liste des ventes et à la liste des courses. A partir de la liste des ventes, le client peut accéder à une liste des lots sur une vente choisi. A partir des informations sur les lots d'une vente, le client peut accéder aux enchères de ce lot. A partir de la liste des course, le client peut accéder aux résultats de la course choisi. Sur les résultats on peut cliquer sur le nom d'un cheval, un client aura accès aux informations du cheval ainsi que l'historique des courses.

Dans le cadre d'une utilisation employé : Un employé arrive sur la page de connexion. L'employé ne peut se connecter que s'il rentre ses informations (identifiant/mot de passe) sinon il ne peut se connecter. Grâce à la barre de navigation, une fois connecté, il peut avoir accès aux ventes(Liste des ventes/ Liste des catégories de vente/ Ajouter une vente), aux clients(Lister les clients/Ajouter un client), aux courses(Lister les courses/Ajouter une course) et ses informations.

Dans le cadre d'une utilisation admin: Un admin arrive sur la page de connexion. L'admin ne peut se connecter que s'il rentre ses informations (identifiant/mot de passe) sinon il ne peut se connecter. Grâce à la barre de navigation, une fois connecté, il peut avoir accès aux ventes(Liste des ventes/ Liste des catégories de vente/ Ajouter une vente/ Ajouter une catégorie de vente), aux clients(Lister les clients/Ajouter un client), aux chevaux(Lister les chevaux/Ajouter un cheval/Lister les types de chevaux/Ajouter un type de cheval) aux courses(Lister les courses/Ajouter une course), aux pays(Lister les courses/Ajouter une course), aux pays(Lister les pays/Ajouter un pays), aux lieux(Lister les lieux/ Ajouter un lieu) et ses informations.

4.4. Interfaces utilisateurs

4.4.1. Interface X

Ce paragraphe décrit pour chaque interface :

son contenu : informations affichées, champs de saisie, type du champ attendu, contrôles de saisie, messages d'erreur, etc...

la façon d'y accéder (pages ou liens menant à l'interface)

les interfaces de sortie possible

Interface de connexion :

Accès via la barre de navigation (“Connexion”) ou par la bannière via le bouton “Se connecter”

Informations affichées :

Champs de saisie pour l’identifiant et le mot de passe, bouton pour se connecter, lien pour s’inscrire

Sortie possible : Page d'accueil si identifiant / mot de passe corrects

Interface de recherche :

Accès via la barre de navigation en tant qu’admin dans l’outil de recherche

Informations affichées :

Champs de saisie pour entrer un nom de client / cheval, bouton pour valider la recherche.

Sortie possible : page du résultat de la recherche

Interface de la liste des clients :

Accès via la barre de navigation en tant qu’admin/employé

Informations affichées :

Information de chaque client avec un bouton modifier / supprimer pour chacun d’entre eux.

Sortie possible : page affichant les chevaux associés au client concerné, page de modification du client et page de suppression du client.

Interface d’ajout d’un client :

Accès via la page de connexion ou via la barre de navigation (uniquement pour l’admin)

Informations affichées :

Champs de saisies pour le nom, prénom, adresse, code postal, ville et champs de sélection des catégories de vente et bouton valider.

Sortie possible : page de la liste des clients lorsque l’inscription est bien effectuée

Interface de modification d’un client :

Accès via la barre de navigation (uniquement pour l’admin et l’employé)

Informations affichées :

Champs de saisies pour le nom, prénom, adresse, code postal, ville et champs de sélection des catégories de vente et bouton valider.

Sortie possible : page de la liste des clients lorsque la modification est bien effectuée

Interface de suppression d’un client :

Accès via la barre de navigation (uniquement pour l’admin et l’employé)

Informations affichées :

Champs de sélection parmi deux choix (Supprimer ou archiver), et pop-up indiquant les risques de cette suppression.

Sortie possible : page de la liste des clients lorsque la suppression est bien effectuée

Interface de la liste des chevaux :

Accès via la barre de navigation en tant qu'admin/employé

Informations affichées :

Information de chaque cheval avec un bouton modifier / supprimer pour chacun d'entre eux.

Sortie possible : page de modification du client et page de suppression du client.

Interface d'ajout d'un cheval :

Accès via la page de connexion ou via la barre de navigation (uniquement pour l'admin)

Informations affichées :

Champs de saisies pour le nom, prénom, sexe, siret, et champs de selection pour la race et le père et mère du cheval inscrit.

Sortie possible : page de la liste des chevaux lorsque l'inscription est bien effectuée

Interface de modification d'un cheval :

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de saisies pour le nom, prénom, sexe, siret, et champs de selection pour la race et le père et mère du cheval inscrit.

Sortie possible : page de la liste des chevaux lorsque la modification est bien effectuée

Interface de suppression d'un cheval :

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de sélection parmi deux choix (Supprimer ou archiver), et pop-up indiquant les risques de cette suppression.

Sortie possible : page de la liste des chevaux lorsque la suppression est bien effectuée

Interface de la liste des races de chevaux :

Accès via la barre de navigation en tant qu'admin/employé

Informations affichées :

Information de chaque cheval avec un bouton modifier / supprimer pour chacun d'entre eux.

Sortie possible : page de modification du client et page de suppression du client.

Interface d'ajout d'une race de cheval :

Accès via la page de connexion ou via la barre de navigation (uniquement pour l'admin)

Informations affichées :

Champs de saisies pour le nom, prénom, sexe, siret, et champs de selection pour la race et le père et mère du cheval inscrit.

Sortie possible : page de la liste des chevaux lorsque l'inscription est bien effectuée

Interface de modification d'une race de cheval :

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de saisies pour le nom, prénom, sexe, siret, et champs de sélection pour la race et le père et mère du cheval inscrit.

Sortie possible : page de la liste des chevaux lorsque la modification est bien effectuée

Interface de suppression d'une race de cheval :

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de sélection parmi deux choix (Supprimer ou archiver), et pop-up indiquant les risques de cette suppression.

Sortie possible : page de la liste des chevaux lorsque la suppression est bien effectuée

Interface de la liste des courses:

Accès via la barre de navigation en tant qu'admin/employé

Informations affichées :

Information de chaque course avec un bouton modifier / supprimer pour chacun d'entre eux.

Sortie possible : page de modification du client et page de suppression du client.

Interface d'ajout d'une course:

Accès via la page de connexion ou via la barre de navigation (uniquement pour l'admin)

Informations affichées :

Champs de saisies pour le nom, date de debut, lieu et champ de sélection pour la catégorie de vente.

Sortie possible : page de la liste des courses lorsque l'ajout est bien effectuée

Interface de modification d'une course:

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de saisies pour le nom, date de debut, lieu et champ de sélection pour la catégorie de vente.

Sortie possible : page de la liste des courses lorsque la modification est bien effectuée

Interface de suppression d'une course:

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de sélection parmi deux choix (Supprimer ou archiver), et pop-up indiquant les risques de cette suppression.

Sortie possible : page de la liste des courses lorsque la suppression est bien effectuée

Interface de la liste des catégories:

Accès via la barre de navigation en tant qu'admin/employé

Informations affichées :

Information de chaque catégorie de vente avec un bouton modifier pour chacun d'entre eux.

Sortie possible : page de modification des catégories.

Interface d'ajout d'une catégorie:

Accès via la page de connexion ou via la barre de navigation (uniquement pour l'admin)

Informations affichées :

Champs de saisies pour le nom et le code.

Sortie possible : page de la liste des catégories de vente lorsque l'ajout est bien effectuée

Interface de modification d'une catégorie:

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de saisies pour le nom et le code.

Sortie possible : page de la liste des catégories lorsque la modification est bien effectuée

Interface de la liste des pays:

Accès via la barre de navigation en tant qu'admin/employé

Informations affichées :

Information de chaque pays avec un bouton modifier / supprimer pour chacun d'entre eux.

Sortie possible : page de modification du pays et page de suppression du pays.

Interface d'ajout d'un pays:

Accès via la page de connexion ou via la barre de navigation (uniquement pour l'admin)

Informations affichées :

Champs de saisies pour le nom et le code du pays.

Sortie possible : page de la liste des pays lorsque l'ajout est bien effectuée

Interface de modification d'un pays:

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de saisies pour le nom et le code du pays.

Sortie possible : page de la liste des pays lorsque la modification est bien effectuée

Interface de suppression d'un pays:

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de sélection parmi deux choix (Supprimer ou archiver), et pop-up indiquant les risques de cette suppression.

Sortie possible : page de la liste des pays lorsque la suppression est bien effectuée

Interface de la liste des lieux:

Accès via la barre de navigation en tant qu'admin/employé

Informations affichées :

Information de chaque lieu avec un bouton modifier / supprimer pour chacun d'entre eux.

Sortie possible : page de modification du lieu et page de suppression du lieu.

Interface d'ajout d'un lieu:

Accès via la page de connexion ou via la barre de navigation (uniquement pour l'admin)

Informations affichées :

Champs de saisies pour le nom de la ville, le nombre de boxes et un commentaire.

Sortie possible : page de la liste des lieux lorsque l'ajout est bien effectuée

Interface de modification d'un lieu:

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de saisies pour le nom de la ville, le nombre de boxes et un commentaire.

Sortie possible : page de la liste des lieux lorsque la modification est bien effectuée

Interface de suppression d'un lieu:

Accès via la barre de navigation (uniquement pour l'admin et l'employé)

Informations affichées :

Champs de sélection parmi deux choix (Supprimer ou archiver), et pop-up indiquant les risques de cette suppression.

Sortie possible : page de la liste des lieux lorsque la suppression est bien effectuée

4.5. Interfaces avec d'autres applications

Ce paragraphe décrit les interfaces (échange d'information, de données, ...) prévues avec d'autres applications.

Notre application échange principalement avec un serveur mySQL qui permet de stocker toutes les données qui transite dans l'application.

4.6. interfaces avec des fichiers

Ce paragraphe décrit les échange de l'application avec des fichiers externes (import/export, batchs, ...).

L'application n'effectue pour l'instant aucun échange avec des fichiers externes.

5. SPECIFICATIONS TECHNIQUES

5.1. Environnement de production

5.1.1. Environnement matériel

Ce paragraphe décrit l'environnement matériel nécessaire pour le fonctionnement de l'application (serveurs, matériels utilisateurs)

Un serveur de production est nécessaire pour le déploiement de l'application web afin qu'il soit utilisé par les utilisateurs. Sur ce serveur de production nous allons y mettre l'application opérationnelle.

*Décrire ici l'environnement de développement, de recette/test livraison et de production
Cette partie servira notamment aux SISR pour mettre en place l'infrastructure*

5.1.2. Environnement logiciel

Ce paragraphe décrit l'architecture logicielle de l'application (serveurs web, serveurs d'application, clients légers/lourds, bases de données, etc...)

Dans le serveur de production, phpmyadmin doit être configuré car nous utilisons le gestionnaire de base de données MySQL.

Configurer Apache Tomcat dans le serveur web.

Cette partie servira notamment aux SISR pour mettre en place l'infrastructure

5.2. Environnement de développement

5.2.1. Environnement matériel

Ce paragraphe décrit l'environnement matériel nécessaire pour le développement de l'application.

Un ordinateur, un serveur de production composé d'un serveur d'application, un serveur de base de données et un serveur web.

5.2.2. Environnement logiciel

Ce paragraphe décrit l'architecture logicielle de l'application (serveurs web, serveurs d'application, clients légers/lourds, bases de données, etc.) nécessaire pour le développement.

MySQL pour la base de données.

L'IDE NetBeans afin de programmer l'application.

Le logiciel de gestion de version Git.

Apache Tomcat pour le serveur web.

JEE (Java Enterprise Edition) pour le serveur d'application.

5.3. Mise en œuvre

Ce paragraphe précise sous quelle forme est livré le logiciel: installation automatique, source et/ou exécutable, éventuellement méthodes de création d'exécutables à partir des sources.

Etape de la mise en production :

Changer le Web.xml pour avoir les bons identifiant de la base de données.

Générer le projet avec la librairie JDK 1.7

Déposer le fichier .war (généré dans le dossier "dist" du projet local) sur le serveur de production.

Importe la base de données sur le serveur de production.

Un document de livraison devra être rédigé par vous.

Il sera appliqué par les SISR.

5.4. Exigences de programmation

Ce paragraphe décrit le langage, les outils et les méthodes de développement, les normes de développement applicables: git, IDE, framework, ...

Le framework JEE est obligatoire pour faire cette application avec l'IDE NetBeans.

Git est utilisé pour les gestions de versions.

6. MODELISATION

6.1. Modèle relationnel et base de données

Ce paragraphe doit présenter la base de données avec le nom logique de toutes les tables et champs, leur signification, le type de champ, les relations entre les tables, les contraintes spécifiques (de volume, de performance, de transactions, temps de réponses attendus, etc...).

client(id, nom, prenom, dateNaissance, adresse, copos, ville, mail, tel, #codePays, archive, active)

Vendeur (#ven_cli_id)

Acheteur (#ach_cli_id)

clientCategVente(#codeClient, #codeCategVente)

Pays (code, nom, archive)

Vente (id, nom, dateDebut, #ven_lieu, #codeCategVente, archive)

CategVente (code, libelle)

Lieu (lie_id, lie_ville, lie_nbBoxes, lie_commentaires, lie_archive)

Courriel (cou_id, cou_dateEnvoiMessage, cou_objetMessage, cou_corpMessage, #cou_vente)

piecesJointes(pie_id, pie_chemin, pie_desc)

piecemail(#pm_idCourriel, #pm_idPieceJointe)

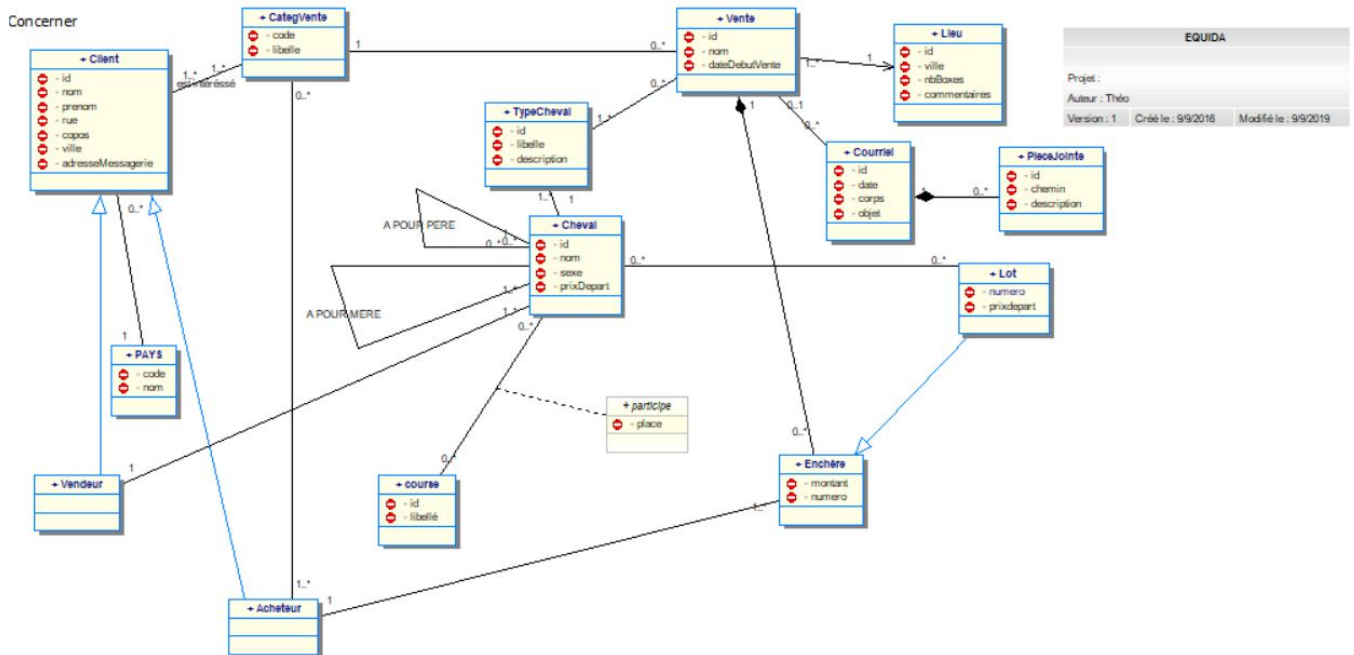
Cheval (che_id, che_nom, che_sexe, che_sire, #che_id_pere, #che_id_mere, #che_client, che_archive, che_active)

Course (id, nom, lieu, date)

Participer (#par_che_id, #par_crs_id, place)

6.2. Diagramme de classes













Ce paragraphe présente le diagramme de classe de la partie métier de l'application (pour un développement objet).



7. ARCHITECTURE APPLICATIVE

Ce paragraphe a pour but de décrire l'architecture technique de l'application, arborescence des dossiers, contrôleurs, images, css, utilisation de bibliothèques d'accès aux données...

L'arborescence de l'application se présente comme ceci :

 .git	Le dossier .git permet d'avoir un suivi de gestion de version, essentiel pendant la période de développement.
 BDD+Script	Le dossier BDD+Script contient le script SQL de la base de données de l'application.
 build	Le dossier build est créé lors du débogage de l'application via l'IDE Netbeans.
 dist	Le dossier dist est également créé par Netbeans lors du build final, nécessaire au déploiement. Il contient le fichier .war, qui est en fait le projet final à déposer sur le serveur de déploiement.
 lib	Le dossier lib contient les différentes librairies nécessaires au développement du projet. Ici, on a juste besoin de MySQLConnector pour interagir avec la BDD.
 Maquettes	Le dossier maquette contient les vues au format powerPoint, nos premières idées quant à celles-ci.
 nbproject	Le dossier nbproject contient les paramètres du projet générés par Netbeans.
 src	Le dossier src contient nos modèles, nos contrôleurs et nos classes pour interagir avec la BDD.
 test	Enfin, le dossier web contient toutes nos différentes vues, accompagnées des dossiers CSS/JS/IMG
 web	
 .gitignore	
 build.xml	

Architecture technique :

Lorsque nous allons sur la page d'accueil, le serveur Java envoie automatiquement la vue « index.jsp », comportement identique sur des serveurs web classiques avec des pages HTML ou PHP. Ensuite, dans cette vue, il y a des liens, et pour illustrer le fonctionnement de l'application nous allons expliquer le fonctionnement de la connexion :

- Premièrement, la page index.jsp contient un lien « connexion » envoyant vers « ../EquidaWebPeaky/ServletConnexion/connexion ». On fait un retour en arrière deux fois car cette page index est composée d'une page « header.jsp » (appelée à chaque vue elle contient la barre de navigation de l'application) qui est stockée dans le dossier /vues/inc/. Etant donné qu'on est à la page index, à la racine du projet, le header doit « pointer » à l'emplacement du fichier ou il est appelé, donc la racine dans ce cas.

Parlons maintenant de ce lien. Il renvoie 2 fois en arrière, appelle le projet, et ensuite « ServletConnexion/connexion » : « **ServletConnexion** » correspond en fait à un des nombreux contrôleurs de l'application, et ce contrôleur agit différemment en fonction du lien qu'il appelle. Ici on appelle le lien « **connexion** ».

```
String url = request.getRequestURI();
HttpSession session = request.getSession();
session.setAttribute("leClient", null);

if(session.getAttribute("EtatConnexion") == null){
    session.setAttribute("EtatConnexion", false);
}

session.setAttribute("RoleUser", null);

if(url.equals("/EquidaWebPeaky/ServletConnexion/connexion"))
{
    if((session.getAttribute("EtatConnexion")).equals(true)){
        session.setAttribute("EtatConnexion", false);
        this.getServletContext().getRequestDispatcher("/vues/connexion.jsp").forward( request, response );
    }else{
        this.getServletContext().getRequestDispatcher("/vues/connexion.jsp").forward( request, response );
    }
}
}
```

Dans cette Servlet, on vérifie premièrement que l'utilisateur n'est jamais venu sur l'application, avec la variable de session (variable temporaire instanciée uniquement le temps de la visite de l'utilisateur sur l'application) « **EtatConnexion** », qui est en fait un booléen envoyant « **true** » quand connecté et « **false** » dans le cas inverse.

Si l'utilisateur n'est jamais venu, la variable n'existe pas, et donc on l'initialise à false, puisqu'il ne doit pas être connecté dès son arrivée.

Ensuite, on crée une variable de session « RoleUser », qui contiendra son futur rôle après s'être connecté.

Quand on appelle le lien « **connexion** », on vérifie d'abord que l'utilisateur n'est pas déjà connecté. Si c'est le cas, on le déconnecte en passant la variable **EtatConnexion** à false. Ensuite on appelle la vue « connexion.jsp » pour qu'il se connecte à nouveau.

S'il n'est pas encore connecté, on appelle une nouvelle fois cette vue pour qu'il se connecte. Etant donné que cette page est en fait un formulaire, il y a une période de traitement avant d'envoyer la réponse. Pour cela, la Servlet est décomposée en deux parties : La partie affichage (à travers la méthode **doGet()**) et la partie traitement (à travers la méthode **doPost()**).

Quand l'utilisateur va cliquer sur le bouton « se connecter », on va passer à la phase de traitement, composé comme ceci :

```
HttpSession session = request.getSession();

ConnexionForm form = new ConnexionForm();

ArrayList<String> listeChamps = form.verifierChamps(request);
request.setAttribute( "form", form );
if(form.getErreurs().isEmpty()){
    /* Transmission de la paire d'objets request/response à notre JSP */
    Compte unCompte = CompteDAO.verifLoginCompte(connection, listeChamps.get(0), listeChamps.get(1));

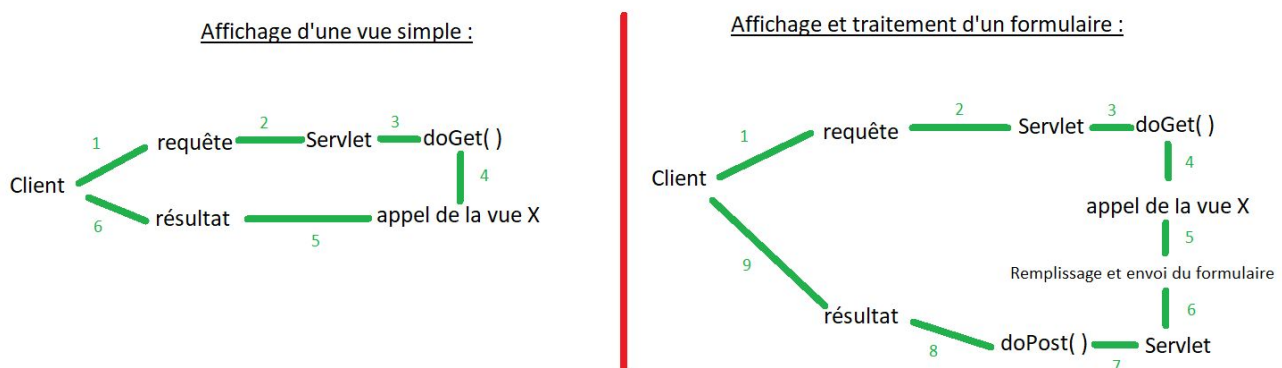
    if(unCompte.isConnexion()){
        session.setAttribute("leClient", ClientDAO.getUnClientParCompte(connection, unCompte.getId()));
        session.setAttribute("EtatConnexion", true);
        session.setAttribute("RoleUser", unCompte.getRole().getLibelle());

        System.out.println("ROLE USER : "+session.getAttribute("RoleUser"));
        response.sendRedirect("/EquidaWebPeaky");
    }else{
        request.setAttribute("pVerifLogin", "PAS OK");
        session.setAttribute("RoleUser", null);
        getServletContext().getRequestDispatcher("/vues/connexion.jsp").forward(request, response);
    }
}else{
    getServletContext().getRequestDispatcher("/vues/connexion.jsp").forward(request, response);
}
```


Premièrement, on récupère toutes les variables de sessions définies précédemment. Ensuite, on appelle une classe de vérification, un sous-contrôleur, pour vérifier l'intégrité des données saisies dans le formulaire. Si ce contrôleur ne retourne aucune erreur, alors on va initialiser un objet **unCompte** (de type **Compte**) et on va l'instancier en appelant une méthode d'une classe DAO (permettant d'interagir avec la BDD), plus précisément **CompteDAO**, avec la méthode « **verifLoginCompte** ». Comme son nom l'indique, elle sert à vérifier les identifiants / mdp d'un compte pour autoriser la connexion ou pas. Si la vérification est correcte, elle instancie l'attribut « **Connexion** » de l'objet **unCompte** à **true**, et **false** dans le cas inverse.

Ensuite, on retourne dans notre méthode **doPost()**, et si l'attribut « **Connexion** » de l'objet **unCompte** est à l'état **true**, alors on initialise une variable de session « **leClient** » en l'instanciant avec la méthode « **getUnClientParCompte** » dans la classe DAO **ClientDAO** qui prend en paramètre l'id du compte récupéré précédemment, et qui retourne le client associé au compte en question. Après, on passe la variable **EtatConnexion** à **true**, et **RoleUser** au rôle récupéré lors de la vérification, et enfin on redirige l'utilisateur vers l'**index.jsp**. Si l'utilisateur n'a pas entré les bons identifiants, on le renvoie vers la page de connexion en lui indiquant que les identifiants sont incorrects.

L'architecture globale fonctionne donc la plupart du temps comme ceci :



8. QUALIFICATION DU LOGICIEL

Ce paragraphe liste les critères de qualité retenus pour déterminer si le logiciel est de qualité. Il décrira notamment les tests à mettre en œuvre pour valider le logiciel. La documentation de tests devra être rédigée.

9. SAUVEGARDE ET RESTAURATION DE LA BASE ET DE L'APPLI

Ce paragraphe doit renvoyer vers un document décrivant le processus de sauvegarde et de restauration du service déployé.
A Réaliser en collaboration avec les SISR.