|  |  |  |  |
| --- | --- | --- | --- |
| **Informatisation du service de location de salles municipales**  **DOSSIER DE CONCEPTION** | | | |
| Ref. Document : |  | Version : | 1 |
| Date : | 25/04/2016 | Nb / pages : | xx |
| Radu-Catalin HOREANU, Gaëtan COUSIN, Adrien LEVEQUE | | Nb/ annexes : | XX |

|  |  |
| --- | --- |
| Résumé d’auteur : Ce projet a pour but la réalisation d’un site WEB permettant la réservation et la gestion des locations des salles. |  |

# Préambule

Aujourd’hui, la location d’une salle implique beaucoup de traitements artisanaux, ce qui entraine un coût élevé de gestion des demandes de location.

L’ensemble des documents entrants et sortants sont gérées manuellement et un archivage physique est mis en place. Cette situation ne facilite pas le travail des employés et implique une efficacité réduite du travail effectué.

De plus, le processus existant ne permet pas de sortir un état fiable sur les réservations enregistrées pour une date donnée. A l’heure actuelle, une synthèse existe, sous forme d’un fichier Excel maintenu par le service de location et alimenté par les données fournies périodiquement par les acteurs du service et les agents municipaux.

Ce fichier s’appuie sur des référentiels hétérogènes. Il n’est pas exhaustif et s’avère difficilement exploitable.

Cette situation implique donc un travail important de maintenance de la part de nombreux acteurs en interne, sur lequel l’entreprise ne capitalise pas et auquel s’ajoute tout le travail chronophage de récupération et centralisation de ces données.

Par ailleurs, une enquête a été réalisée par l’entreprise auprès de ses clients pour recueillir leurs opinions et leurs besoins par rapport à ce service. 90% des clients considèrent que le processus existant est lourd et que la mise en place d’un site WEB facilitera les échanges client-entreprise et simplifiera les formalités nécessaires à la réservation d’une salle.

L’entreprise a décidé de mettre en place un site WEB permettant la visualisation des salles disponibles et leurs réservations en ligne.

## Objectif du projet

L’objectif de ce site WEB est d’informatiser le service de location de salles pour :

* améliorer les conditions de travail des agents ;
* répondre au mieux aux besoins des clients ;
* augmenter l’efficacité du travail effectué ;
* garantir la fiabilité des traitements ;
* réduire le coût de gestion et d’archivage ;
* obtenir des informations cohérentes pour la prise de décision (reporting) et comptabilisation.

## DOCUMENTS DE REFERENCE

Les documents suivants sont à utiliser en référence avec la lecture de ce document :

* Le Cahier Des Charges dans sa version finale : il contient les spécifications initiales des exigences du maître d’ouvrage.

# Environnement et outils de développement

Le matériel de développement utilisé est une machine préparée pour chaque développeur, équipée de Windows 10 et d’une quantité suffisante de mémoire vive (le minimum a été fixé à 1Go pour avoir une qualité de développement acceptable, en partie en raison des nombreux services à exécuter). Les trois membres de l’équipe de développement exécutent les applications du projet sur leurs propres machines. La base de données MySQL est située sur Apache.

## Standards de programmation

L’équipe de développement suit un ensemble de conventions de codage qui permettent une homogénéisation des sources :

* Les conventions de codage SQL sont celles recommandées dans le cours de BD.
* Les conventions approuvées par le W3C sont également appliquées dans le cadre du développement WEB (HTML5 et CSS3).

## Langages utilisés

Voici la liste des différents langages utilisés dans le projet :

* SQL pour les scripts de création de la base de données (création de tables et insertions des données dans la BD).
* PHP, HTML5, CSS3 et Javascript pour la création du site WEB.

# CONCEPTION GENERALE

La construction du modèle conceptuel de données a été réalisée à partir des spécifications du cahier des charges, du modèle métier et des réunions avec le maître d’ouvrage.

## Création de la base de données

1. Règles de gestion

Les règles de gestion décrivent la nature des relations entre les entités d’un système d’information. L’ensemble de ces règles permet de définir un système correspondant à une problématique métier précisément adaptée aux besoins du client. Ces règles de gestion sont utilisées directement dans le modèle métier : chaque règle correspond à une relation entre 2 (ou plusieurs) entités. Les entités métiers sont indiquées en gras.

* R1 : chaque client a un statut, une priorité ;
* R2 : le prix final de location de la salle dépend du statut du client ;
* R3 : une réservation est faite pour une salle donnée ;
* R4 : une réservation peut contenir des matériels ;
* R5 : un client peut effectuer une ou plusieurs réservations. Une réservation correspond à un seul client. Une réservation effective est une location.
* R6 : une réservation est payée avec un mode de règlement particulier (chèque, CB, etc.).

1. Base de données

Les tables de la base de données sont issues de deux catégories de données dans le MCD : les entités (employés, clients, etc.) et les associations (rattachement d’un matériel à une réservation etc.).

Chaque entité possède un libellé lisible qui permet de la distinguer clairement. Ce libellé peut contenir des chiffres, mais ne peut pas contenir d'articles dans le nom (ex : LE\_CLIENT), ni de verbes).

Les associations reliant toujours deux entités au minimum, le libellé des tables correspondantes est une caractéristique des deux entités, ce qui donne par exemple « Ligne\_méteriel » pour la table correspondant à l’association rattachant un matériel à une réservation. Etant donné que plusieurs associations peuvent exister entre 2 tables, on ajoutera alors au libellé de la table le nom caractérisant cette association (Ligne etc.). Les noms des tables ne devront jamais dépassés 30 caractères.

Concernant les champs de la base de données, les noms des colonnes ont les mêmes règles que les libellés des tables.

Un index sur une clé primaire est créé automatiquement et portera le nom de la clé primaire : PK\_nom de la table. Les index sur les clés étrangères seront préfixés de FK. On aura FK\_nom de la référence. La base de données sera créée sur le serveur APACHE en MySQL.

Concernant les requêtes SQL présentes dans les fichiers sources, elles doivent être écrites de la façon suivante : lettres majuscules pour les mots SQL (ex. : SELECT, UPDATE, TO\_CHAR) et lettres minuscules pour les noms des objets sur lequel porte la requête (noms des tables, champs, variables, etc.).

Avant chaque mot SQL, il est souhaitable d'avoir un retour à la ligne et d'aligner les lignes de la requête (avec des espaces et non des tabulations qui ne sont pas portables pour la mise en page). De même, les conditions des sous-requêtes doivent être décalées par rapport à la requête principale.

Les principales tables de la base de données sont :

**Table « Clients »**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description (rôle) | Domaine ou type | Commentaires | Contraintes, règles de calcul |
| id\_client | N° de client | int (entier) | N° séquentiel | Automatique |
| nom\_client | Nom du client | varchar 20 | Nom ou raison sociale, format : tout en majuscules | Obligatoire |
| prenom\_client | Prénom client | varchar 20 |  | Obligatoire |
| adresse\_client | Adresse | varchar 40 |  | Obligatoire |
| telephone\_client | Téléphone | varchar 20 |  | Obligatoire |
| courriel\_client | Courriel | varchar 30 |  | Obligatoire |
| date\_naissance\_client | Date de naissance | Date |  | Obligatoire |
| mdp\_client | Mot de passe | varchar 20 |  | Obligatoire |

**Table « Statut »**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description (rôle) | Domaine ou type | Commentaires | Contraintes, règles de calcul |
| id\_statut | N° Statut | int (entier) | N° séquentiel | Automatique |
| libelle\_statut | Libellé Statut | varchar 20 |  | Obligatoire |
| reduction\_statut | Pourcentage réduction | Pourcentage | Le pourcentage est établi par le client/règles de gestion. | Obligatoire |
| priorite | Priorité | int (entier) | La priorité est établie par le client/règles de gestion. | Obligatoire |
| sans\_recette\_statut | Si recette | Booléen | Oui/Non | Obligatoire |

**Table « Salles»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description (rôle) | Domaine ou type | Commentaires | Contraintes, règles de calcul |
| id\_salle | N° salle | int (entier) | N° séquentiel | Automatique |
| nom\_salle | Nom de l’employé | varchar 20 | Nom ou raison sociale, format : tout en majuscules | Obligatoire |
| adresse\_salle | Adresse | varchar 30 |  | Obligatoire |
| capacite\_salle | Capacité | int (entier) |  | Obligatoire |
| gardian\_salle | Gardien | Booléen | Oui/Non | Obligatoire |
| caution\_salle | Caution | float 5,2 | Monnaie : € | Obligatoire |
| tarif\_salle | Tarif | float 5,2 | Monnaie : € | Obligatoire |

**Table « Type salle»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description (rôle) | Domaine ou type | Commentaires | Contraintes, règles de calcul |
| id\_type\_salle | N° type salle | int (entier) | N° séquentiel | Automatique |
| libelle\_type\_salle | Nom de la salle à louer | varchar 20 |  | Obligatoire |

**Table « Clef »**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description (rôle) | Domaine ou type | Commentaires | Contraintes, règles de calcul |
| id\_clef | N clefs | int (entier) | N° séquentiel | Automatique |
| emruntee\_clef | Clef donné au client | varchar 20 | Mis à jour par l’employé | Booléen (oui ou non) |

**Table « Matériel »**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description (rôle) | Domaine ou type | Commentaires | Contraintes, règles de calcul |
| id\_materiel | N° Matériel | int (entier) | N° séquentiel | Automatique |
| libelle\_maerial | Nom du matériel | varchar 20 | Nom ou raison sociale, format : tout en majuscules | Obligatoire |
| caution\_materiel | Caution | float | Monnaie : € | Obligatoire |

**Table « Réservation »**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description (rôle) | Domaine ou type | Commentaires | Contraintes, règles de calcul |
| id\_reservation | N° salle | int (entier) | N° séquentiel | Automatique |
| date\_debut | début de la réservation | Date |  | Obligatoire |
| date\_fin | fin de la réservation | Date |  | Obligatoire |
| etat\_lieux\_effectue | état de lieux | Booléen | Oui/Non | Obligatoire |

Les tables seront alimentées par les formulaires WEB. Exemple :

Les coordonnées d’un client seront introduites dans la base de données par l’intermédiaire du formulaire d’inscription ci-dessous :



## Sécurisation des accès aux données

Les données sont accessibles au travers de différents écrans, offrant les différentes fonctionnalités de l’application.

Pour chaque rôle, les fonctionnalités accessibles sont définies dans la base de données.

L’application est développée pour prendre en compte les fonctionnalités accessibles à l’utilisateur courant. Pour cela elle n’affiche que les écrans autorisés à l’utilisateur.

De plus certaines portions de formulaires ou de menus peuvent être masquées. Les droits d’accès sont éditables par l’administrateur qui dispose de tous les droits sur tous les écrans de l’application.

## caractéristiques site web

**Le menu**

Le site présentera en permanence un menu, sur toutes les pages qui permet de voir le plan du site. Ce menu permettra d’accéder à chaque salle, d’accéder à telle ou telle activité, ou de revenir à la page d’accueil.

**Authentification utilisateur**

Cette ’application comportera une interface de gestion destinée à gérer les différents processus de l’entreprise. Cette partie de l’application n’est pas destinée à être accessible par une personne extérieure. L’authentification des employés est donc nécessaire.

Les connexions utilisateurs sont effectuées par identification classique : login et mot de passe. Ces informations sont stockées dans la base de données, leur vérification est donc immédiate lors de l’authentification de l’utilisateur.

**Scénario administrateur**

Un compte administrateur sera présent dans la base de données, afin d’effectuer des tests sur l’utilisation du site.

De plus, nous prévoyons de développer une page d’administration protégée par mot de passe, qui nous permettra de voir la liste des utilisateurs enregistrés.

**Sessions utilisateurs**

L’authentification d’un utilisateur conduit à l’ouverture d’une session qui permettra à celui-ci de naviguer sur tout le site sans devoir s’identifier à chaque page. Les droits alloués à cet utilisateur doivent être conservés. On conserve donc les droits associés au type d’utilisateur auquel appartient l’employé. Lors de la connexion de l’utilisateur, une entité contenant les droits de celui-ci sera donc initialisée pour la session ouverte.

**Gestion interne de l’authentification**

Lors de la soumission du formulaire d’identification par l’utilisateur, une première vérification est effectuée, permettant de s’assurer que ni le champ de mot de passe, ni le champ de login ne sont vides.

Ensuite, les données sont comparées avec les données stockées en base :

- vérifier que le mot de passe correspond bien à l’utilisateur ;

- dans le cas où le mot de passe est erroné, rediriger vers le formulaire avec une notification d’erreur ;

- dans le cas où le mot de passe est correct, charger les droits de l’utilisateur et rediriger vers la page d’accueil de l’application.

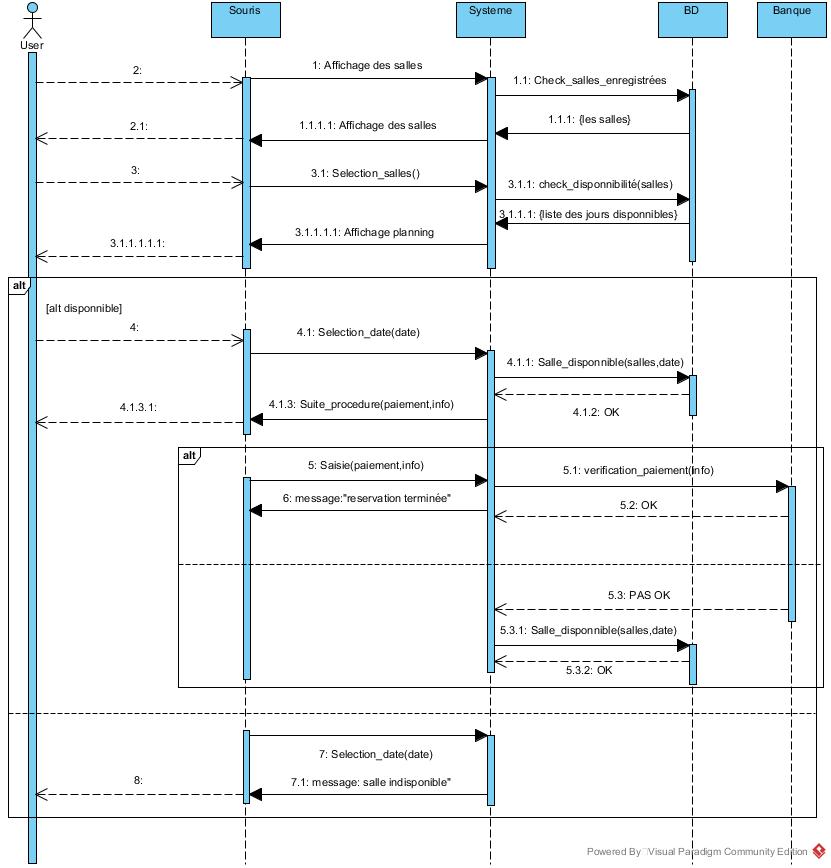
**Expiration de session et reconnexion**

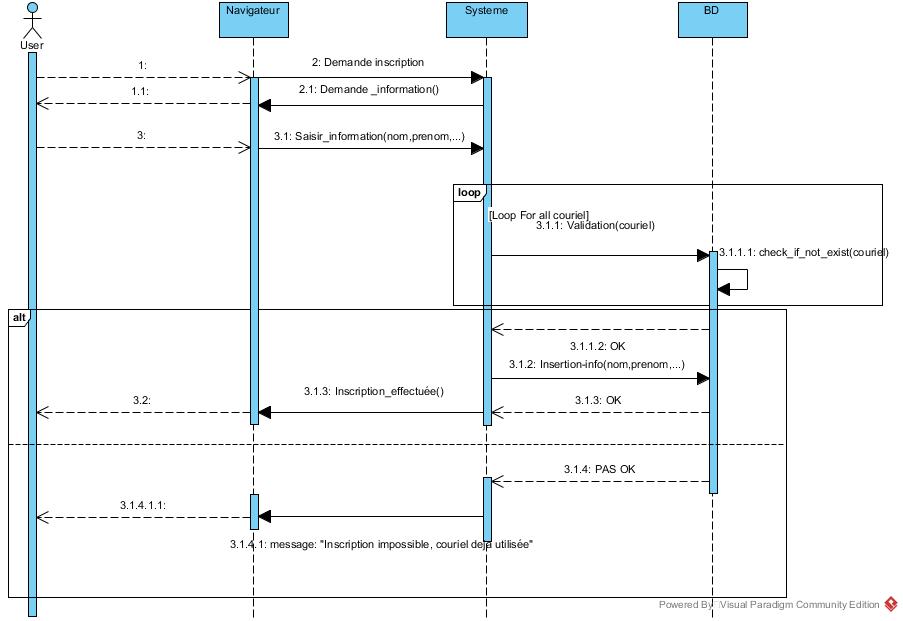
Toute session de connexion possède un « time out », qui correspond à la durée pendant laquelle une session sans activité reste connectée. Au-delà de ce temps, l’utilisateur doit saisir ses informations de connexion à nouveau.

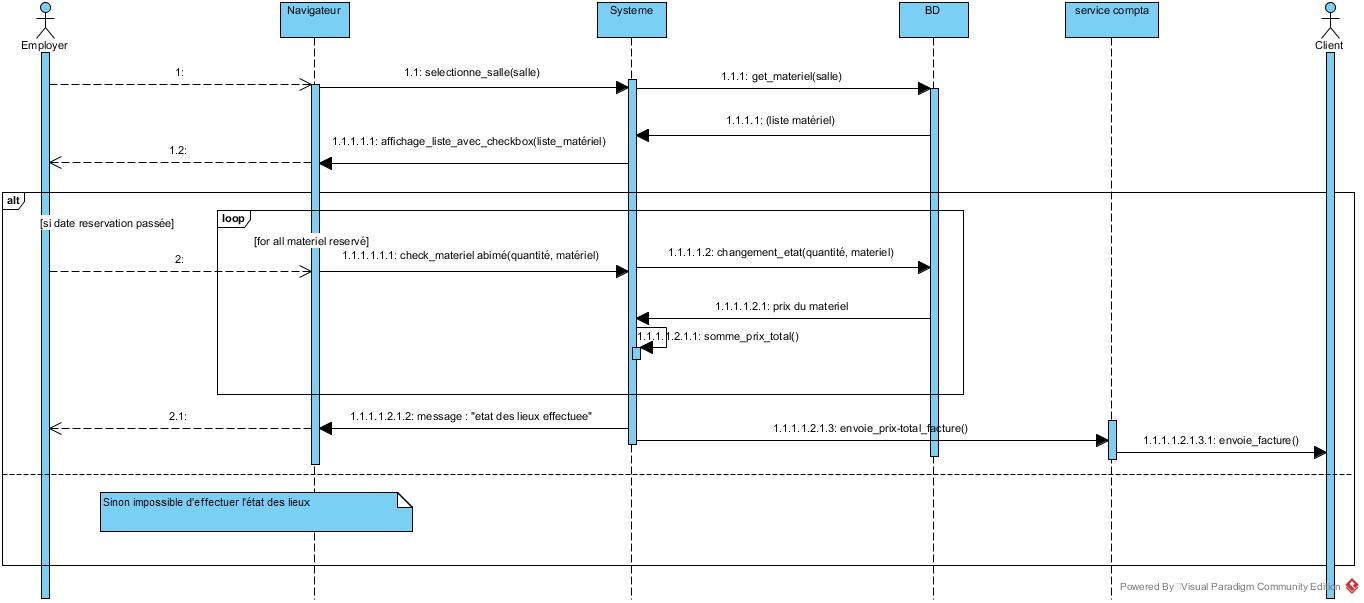
**Le ton et style de texte**

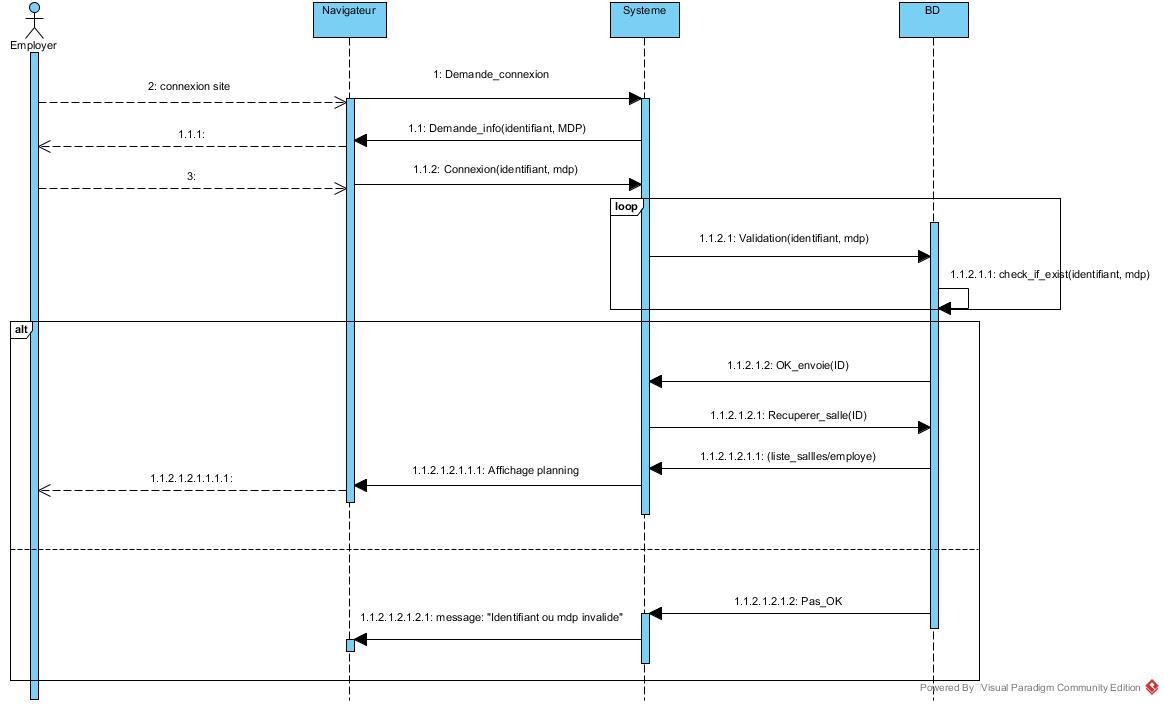
Le genre éditorial sera un langage simple et clair, pour que notre cible comprenne facilement de quoi il est question. Nous utiliserons des mots techniques, mais notre objectif est de rester clair dans nos propos. Le ton sera globalement sérieux.

### Diagrammes de séquence

****

****

****

****