

Spécifications techniques

M2L – Application de gestion des formations

Table des matières

**I – Introduction**

**II – Cas d’utilisation**

**III – Modèle conceptuel de données**

**IV – Fonctionnalités du service**

**V – Règles de codage et conventions de nommage**

**I – Introduction**

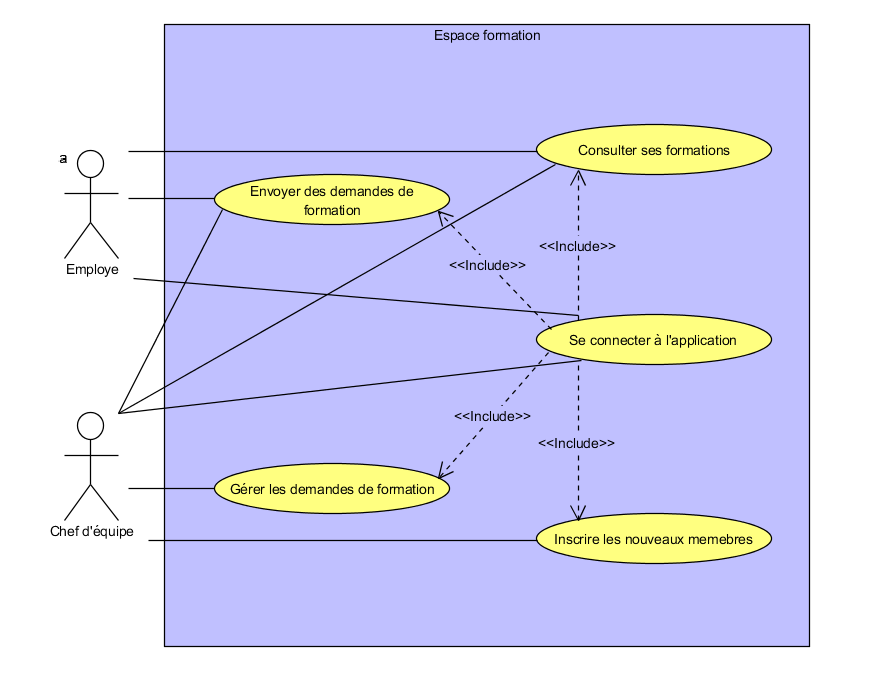
* 1. **– Objet du document**

Ce document a pour but de présenter les fonctionnalités de l’application de gestion des formations. Elle est disponible via l’intranet de la Maison des Ligues et accessible à tous les employés.

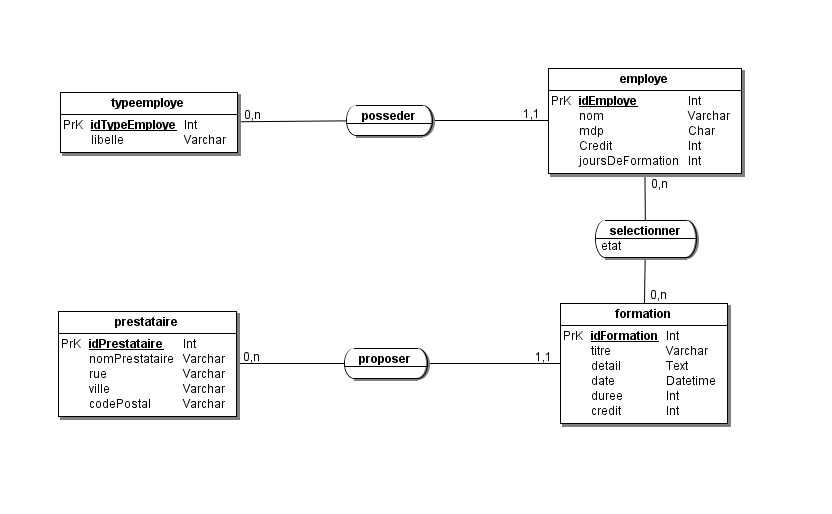
L’application nécessite dans un premier temps la connexion à un compte, l’employé est redirigé vers une page affichant le catalogue de formation proposé par la Maison des Ligues de Lorraine. Sur cette même page il a la possibilité de consulter le détail de chaque formation et de s’inscrire si son crédit lui permet. Chaque formation à un coût, il est défini par le crédit. Chaque employé voit son crédit augmenter selon son ancienneté au sein de la Maison des Ligues. Par l’intermédiaire du menu de navigation, l’utilisateur peut accéder à l’historique des formations validées, refusées et en attente de validation le concernant.

Chaque employé possède un supérieur, lorsqu’une demande de formation est effectuée (si le crédit de l’employé le permet), elle nécessite la validation du supérieur de l’employé pour être définitivement validée. D’un point de vue technique, un employé étant lui-même supérieur et ne possédant pas de supérieur, est en mesure de valider ses propres demandes de formation.

**II – Cas d’utilisation**

****

**III– Modèle conceptuel de données**

****

**IV – Fonctionnalités du service**

Connexion à l’application :

La connexion à l’application nécessite un compte utilisateur créé par un chef d’équipe de la Maison des Ligues de Lorraine. Elle est obligatoire pour accéder aux fonctionnalités de l’Espace Formation. Il est nécessaire de renseigner l’identifiant et le mot de passe qui y est associé. La saisie du mot de passe est sécurisée, cryptée avant son interaction avec la base de données.

Le niveau d’habilitation de l’utilisateur est déterminé au moment de la connexion, si lors de l’interrogation de la base, le résultat reçu correspond au statut ‘Chef d’équipe’, l’utilisateur aura accès aux fonctionnalités ‘Gestion des demandes de formation’ et ‘Inscription au service’ sinon, il n’y aura pas accès.

Consulter le catalogue de formation :

Depuis la page de catalogue de formations, l’utilisateur à la possibilité d’envoyer des demandes de formation. A l’envoi de la demande, le code PHP vérifie par l’intermédiaire de plusieurs fonctions que l’utilisateur a en sa possession assez de crédit et de jours de formation pour effectuer sa demande. Si c’est le cas on vérifie que la demande n’existe pas déjà dans la base. Si toutes les conditions sont réunies, la demande est créée et stockée dans la base de données dans la table ‘selectionner’.

Consulter son historique de formation :

L’utilisateur a la possibilité de consulter l’historique de ses demandes de formations, dès l’arrivée sur la page, la base de données est interrogée et le résultat est directement renvoyé à l’utilisateur.

Gestion des demandes de formation :

Si l’utilisateur est un Chef d’équipe il peut accéder à cette fonctionnalité, il lui est alors possible de consulter, refuser et accepter les demandes de formation envoyées par chaque utilisateur, lui compris.

Si l’une des demandes est refusée, le statut de la demande est modifié dans la base de données, le crédit de l’utilisateur ayant effectué la demande lui est aussitôt restitué. Sur la page la demande disparaît et un message confirmant l’action effectuée apparaît au-dessus de la liste des demandes.

Si l’une des demandes est acceptée, le statut de la demande est modifié dans la base de données. Par l’intermédiaire d’un trigger, le nombre de jours de formation disponibles de l’employé concerné est décrémenter à hauteur de la durée en jours de la formation acceptée. Sur la page la demande disparaît et un message confirmant l’action effectuée apparaît au-dessus de la liste des demandes.

Inscription au service :

L’utilisateur par l’intermédiaire d’un formulaire à la possibilité de créer un nouveau profil destiné à un nouvel arrivant dans la Maison des Ligues de Lorraine. Il renseigne : l’identifiant, deux champs ‘Mot de passe’ qui seront cryptés au moment de l’envoi, le statut du salarié (‘Employé’ ou ‘Chef d’équipe’), le crédit et le nombre de jours de formation disponibles du salarié. Tous les champs sont obligatoires (fonctionnalité html).

**V – Règles de codage et conventions de nommage**

Base de données :

Les noms de tables ne peuvent contenir que des caractères alphanumériques. Les tirets bas ("\_") ne sont pas permis. Les nombres sont autorisés mais déconseillés. Les noms de tables doivent toujours commencer avec une lettre en minuscule. Quand un nom de fonction est composé de plus d'un seul mot, la première lettre de chaque mot doit être mise en majuscule. C'est ce que l'on appelle communément la "notationCamel".

Inclusion des scripts :

L'inclusion de scripts peut être réalisée par plusieurs instructions prédéfinies en PHP : include, require, include\_once, require\_once. Toutes ont pour objectif de provoquer leur propre remplacement par le fichier spécifié. Les instructions include et require sont identiques, hormis dans leur gestion des erreurs. include produit une alerte ( warning ) tandis que require génère une erreur fatale . En d'autres termes, lorsque le fichier à inclure n'existe pas, le script est interrompu. include ne se comporte pas de cette façon, et le script continuera son exécution. La différence entre require et require\_once (idem entre include et include\_once) est qu'avec require\_once(), on est assuré que le code du fichier inclus ne sera ajouté qu'une seule fois, évitant de ce fait les redéfinitions de variables ou de fonctions, génératrices d'alertes.

Tag PHP :

Toujours utiliser <?php ?> ou <?= ?> pour délimiter du code PHP, et non la version abrégée <? ?>. C'est la méthode la plus portable pour inclure du code PHP sur les différents systèmes d'exploitation et les différentes configurations.

Indentation et longueur des lignes :

Le pas d’indentation doit être fixe et correspondre à 4 caractères. Ce pas d'indentation doit être paramétré dans l'éditeur de l'environnement de développement. L’indentation est stockée dans le fichier sous forme de 4 caractères espace (sans tabulation réelle). Lorsqu’une ligne d’instruction est trop longue, elle doit être coupée après une virgule ou avant un opérateur. On alignera la nouvelle ligne avec le début de l’expression de même niveau de la ligne précédente.

Appel de fonction et méthodes :

Les fonctions doivent être appelées sans espace entre le nom de la fonction, la parenthèse ouvrante, et le premier paramètre ; avec un espace entre la virgule et chaque paramètre et aucun espace entre le dernier paramètre, la parenthèse fermante et le point-virgule. Il doit y avoir un espace de chaque côté du signe égal utilisé pour affecter la valeur de retour de la fonction à une variable. Dans le cas d'un bloc d'instructions similaires, des espaces supplémentaires peuvent être ajoutés pour améliorer la lisibilité.

Nommage des fonctions :

L'identificateur d’une fonction est un verbe, ou groupe verbal. Les noms de fonctions ne peuvent contenir que des caractères alphanumériques. Les tirets bas ("\_") ne sont pas permis. Les nombres sont autorisés mais déconseillés. Les noms de fonctions doivent toujours commencer avec une lettre en minuscule. Quand un nom de fonction est composé de plus d'un seul mot, la première lettre de chaque mot doit être mise en majuscule. C'est ce que l'on appelle communément la "notationCamel".

Fonctions (Algorithmique) :

Le codage doit être réalisé en recherchant le plus possible la modularité : - chaque fonction doit réaliser un et un seul traitement, - chaque fonction doit être construite de manière à posséder la plus forte cohésion et la plus grande indépendance possible par rapport à son environnement.