



L007 - Technologies du Web

Rapport de projet : maNounou.com

LAGORCE Bertille - LALLIER Robin

Semestre P18

TABLE DES MATIÈRES

INTRODUCTION

ANALYSE DES BESOINS/CONTRAINTES

MODÉLISATION DES FONCTIONNALITÉS

CONCEPTION DU PROJET

Organisation du projet

Langages de programmation utilisés

Outils utilisés

Quel(s) langage(s)/outil(s) pour quelle(s) fonctionnalité(s) ?

CONCEPTION DE LA BASE DE DONNÉES

Schéma de la base de données

DIFFICULTÉS RENCONTRÉES

Utilisation de Git

Mise en place des classes php

CONCLUSION

Évolution de nos compétences

ANNEXES

Script SQL de la base de données

INTRODUCTION

Ce rapport retrace les différentes étapes qui ont été réalisées afin de mener à bien ce projet effectué dans le cadre de l'UE L007. Il s'agissait de créer un site internet dédié à la mise en relation de nounous et de parents. Les parents peuvent choisir les nounous en fonction de plusieurs critères tels que la disponibilité, les compétences, les évaluations des prestations précédentes etc... Ainsi, grâce à cette plateforme, des parents peuvent s'inscrire en renseignant leur secteur, nombre d'enfants etc, et rechercher une nounou disponible pour telle ou telle date, un certain type de garde et autres options. De nos jours, ce genre de plateforme se développe de plus en plus, ce pourquoi nous n'avons pas eu de mal à cerner son fonctionnement et imaginer la manière dont le site serait agencé. Nous nous sommes servi des notions abordées en cours et de nos connaissances personnelles pour construire petit à petit *maNounou.com*, *le meilleur site de recherche de nounous* !. La lecture de ce rapport vous apportera les informations nécessaires sur la manière dont nous avons travaillé.

ANALYSE DES BESOINS/CONTRAINTES

Contraintes

En reprenant l'énoncé du sujet de projet nous avons déjà des **contraintes** définies :

- serveur Apache
- pages HTML
- feuilles de style CSS
- scripts PHP
- instructions JavaScript
- base de données relationnelle
- éventuellement éléments de Bootstrap (que nous avons utilisé)

Le cahier des charges contient aussi des **contraintes relatives au site web** :

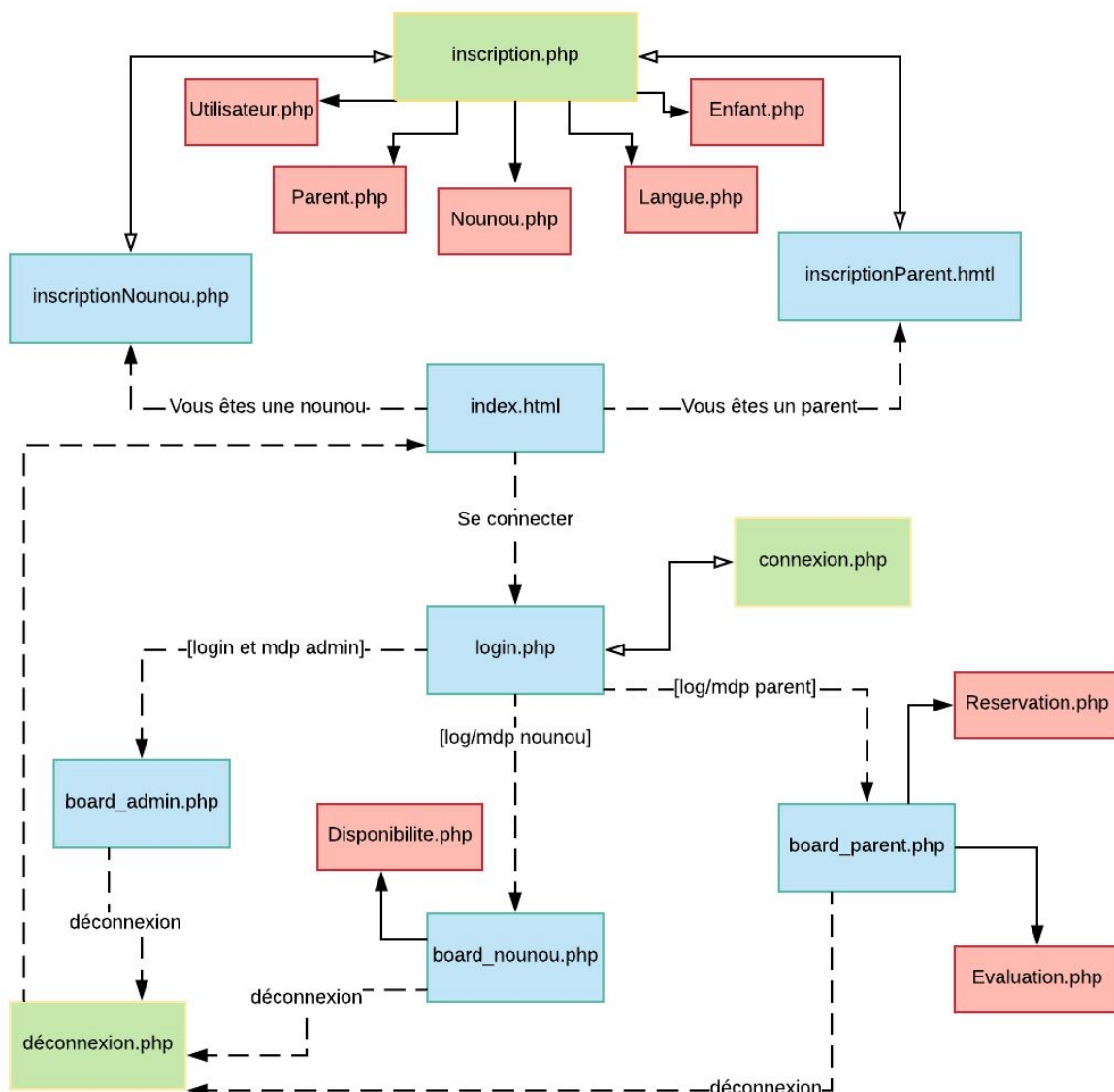
- Possibilité de connexion (admin, parent, nounou)
- Inscriptions sur le site
- Affichage de listes et d'informations, calcul du chiffre d'affaire, modération des nounous
- Définir des disponibilités, effectuer une réservation sur un créneau
- Évaluer une prestation
- Accéder à quelques informations sur le site en tant que visiteur

Besoins

Face à ses contraintes, voici les besoins importants nécessaires au **bon fonctionnement du site web** et permettant de répondre au mieux à ce qui nous est demandé :

- Création de formulaires (connexion, inscriptions...), dont les données vont être redirigées vers la base de données qui va donc être mise à jour.
- Gestion des login et mot de passe pour permettre à chaque type d'utilisateur d'accéder à son propre espace personnel
- Gestion des erreurs (mauvais mot de passe, login déjà existant, inexistence d'une donnée dans la base)
- Mise à jour des données
- Barres de recherche, boutons poussoir et requêtes SQL adaptées en fonction des informations rentrées par l'utilisateur.
- Affichage des infos de manière dynamique (utilisation de JavaScript de façon à ne pas avoir trop de pages et de ne pas encombrer la page).
- Création de variables de session
- Possibilité de déconnexion

MODÉLISATION DES FONCTIONNALITÉS

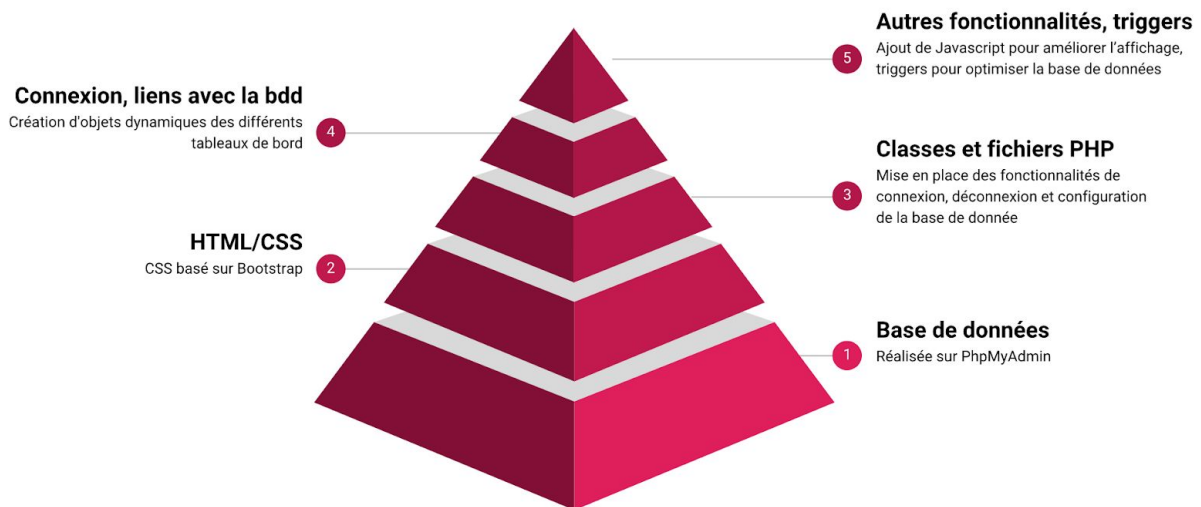


Par soucis de clarté nous n'avons pas précisé les appels au fichier php : [config.php](#) qui permet de faire le lien avec notre base de données.

Les diverses fonctionnalités associées à chaque profil (recherche d'une nounou, affichage du chiffre d'affaire, réservation...) se font directement via les pages [board_*](#), nous n'avons donc pas eu besoin d'implémenter une page pour chaque fonctionnalité. Celles-ci s'affichent/se cachent dynamiquement en fonction de l'action de l'utilisateur.

CONCEPTION DU PROJET

Organisation du projet



Afin d'optimiser notre travail, nous avons organisé notre travail de la manière suivante :

1. En premier lieu, nous avons réalisé notre base de données, afin d'assurer la persistance des données.
2. Nous avons ensuite créé les pages principales de notre arborescence, à savoir l'accueil et les tableaux de bords de chaque type d'utilisateur.
3. Viennent ensuite la création des scripts php nécessaire à la connexion, à la gestion des variables de session ainsi que les classes faisant le lien vers la base de données
4. La partie principale consistait en la création de requêtes SQL et PHP accédant à la base de données pour insérer ou afficher des tuples, et à la mise en forme de ceux-ci en des éléments visuels.
5. Enfin, Nous avons ajouté des éléments Javascript pour améliorer l'affichage des tableaux et des triggers pour optimiser la gestion de la base de données.

Langages de programmation utilisés



Outils utilisés



PhpStorm



GitHub



Bootstrap

Quel(s) langage(s)/outil(s) pour quelle(s) fonctionnalité(s) ?

Récupération et gestion des données

Nous avons utilisé un système de gestion de bases de données (phpMyAdmin) afin de pouvoir conserver les données une fois rentrées par l'utilisateur. La base de données que nous avons construite sera détaillée dans la partie qui lui est consacrée. Ainsi, les pages du sites php vont être reliées à cette base de données et l'utilisateur va, par le biais de requête préalablement insérées dans le code, effectuer des sélections voire des changements sur cette base lorsqu'il utilisera la plateforme.

Pour récupérer les données, nous avons implémenté des formulaire HTML en utilisant la méthode **POST**, puis des scripts PHP afin de récupérer les données et de formuler quelques actions en fonction de celles-ci grâce à des instructions PHP.

Inscriptions (hachage de mot de passe, classes PHP..)

Concernant les inscriptions, nous avons eu recours aux classes PHP, qui ont d'ailleurs été abordées en cours et vivement conseillées dans l'implémentation du projet. Le principe de récupération des données se fait aussi grâce à la méthode POST, renvoyant vers un script PHP inscription.php. C'est dans ce script que nous hachons tout d'abord le mot de passe rentré à l'inscription pour des questions de sécurité.

Puis, un nouvel utilisateur est créé dans la base, et il en va de même en fonction de si c'est une nounou ou un parent. Nous avons au préalable créé des classes PHP ce qui nous permet de faire directement : `New ____()` en rentrant simplement en paramètres les informations.

```
//Hachage du mot de passe
$password = password_hash($_POST["password"], $algo: PASSWORD_DEFAULT);

$user = new Utilisateur($_POST["nom"], $_POST["prenom"], $_POST["ville"],
    $_POST["email"], $_POST["telephone"], $_POST["login"], $password);
$user->addToDatabase($bdd);

//On Checke si l'inscription est celle d'une nounou ou d'un parent
if (isset($_POST["nounou_experience"]) && ($_POST["nounou_experience"] != "")) {
    $nounou = new Nounou($user->getIdUtilisateur(), $_POST['photo'], $_POST['age'], $_POST['nounou_experience'], $_POST['description']);
    $nounou->addToDatabase($bdd);

    $langues = $_POST['langue'];
    foreach($langues as $langue){
        $j = new Langue($user->getIdUtilisateur(), $langue);
        $j->addToDatabase($bdd);
    }

    header( string: "Location: http://localhost:8888/index.html");
} //On vérifie si l'utilisateur est un parent
elseif (isset($_POST["informations"])) {
    $parent = new Parents($user->getIdUtilisateur(), $_POST["informations"]);
    $parent->addToDatabase($bdd);
}
```

Aller piocher dans la base de données

Suite à une demande de l'utilisateur, il va être parfois nécessaire d'aller chercher des informations dans la base de données. Pour cela, ce sont les **requêtes SQL** qu'il va falloir intégrer dans les scripts PHP.

Voici un exemple tout simple, lorsque l'administrateur veut savoir combien de nounous sont inscrites sur sa plateforme.

```
<section class="my-3">
    <!--Nombre de nounous inscrites et acceptées-->
    <?php
    $requete = "SELECT * FROM NOUNOU WHERE candidature=0";
    $resultat = mysqli_query($bdd, $requete);

    if ($resultat) {
        $nbrnounous = mysqli_num_rows($resultat);
        if ($nbrnounous != 0) {
            echo("<p class='h5'>Vous avez " . $nbrnounous . " nounou(s) inscrite(s) sur la plateforme.<br></p>");
        } else {
            echo("<p class='h5'>Aucune nounou n'est encore inscrite sur la plateforme.<br></p>");
        }
    }
    ?>
</section>
```

Gestion des événements (déclenchement de trigger, erreurs...)

Nous avons choisi de gérer certains événements via des triggers (ou déclencheurs), gérés directement dans la base de données. → Par exemple, à chaque nouvelle réservation le trigger `revenu_after_insert` va se déclencher et faire en sorte que les revenus de la nounou associées soient modifiés en fonction de la réservation.

```
1 DROP TRIGGER IF EXISTS revenu_after_insert;
2
3 DELIMITER //
4
5 CREATE TRIGGER revenu_after_insert
6 AFTER INSERT
7 ON reservations FOR EACH ROW
8 BEGIN
9
10     DECLARE countEnfant int(11);
11
12     SELECT COUNT(*) INTO countEnfant
13     FROM ENFANTS e, PARENTS p
14     WHERE e.idParents = p.idParents AND p.idParents = NEW.idParents;
15
16     IF (countEnfant = 1) THEN
17
18         UPDATE NOUNOU n
19         SET n.revenus = IFNULL(n.revenus,0) + (NEW.nbr_heures*7)
20         WHERE n.idNounou = NEW.idNounou;
21
22     ELSE
23
24         UPDATE NOUNOU n
25         SET n.revenus = IFNULL(n.revenus,0) + (NEW.nbr_heures*7) + (countEnfant*4)
26         WHERE n.idNounou = NEW.idNounou;
27
28     END IF;
29
30 END; //
31
32 DELIMITER ;
```

Certains événements vont être aussi gérés directement dans les scripts PHP, comme par exemple les événements de type erreurs. → Ici, l'exemple de la gestion d'erreur de login et de mot de passe, qui initie en même temps les sessions.

```
<?php
include 'config.php';

//Vérification de la validité des informations entrées
if( isset($_POST["login"]) && isset($_POST["password"])) {

    $login = $_POST["login"];

    //Vérification du mot de passe
    $password = $_POST["password"];
    $result1 = mysqli_query($bdd, "SELECT * FROM UTILISATEURS WHERE login='$login'");

    $resultat = mysqli_fetch_array($result1, resulttype: MYSQLI_ASSOC);
    $isPasswordCorrect = password_verify($password, $resultat['mdp']);

    if( mysqli_num_rows($result1) == 1 && $isPasswordCorrect){
        session_start();
        if($resultat['admin'] == '1') {

            $_SESSION['categorie']='admin';
            $_SESSION['nom']=$_POST["login"];
            header( "string: Location: http://localhost:8888/html/board_admin.php");
        }
        else{

            //Associe un id à la session de l'utilisateur
            $_SESSION['id'] = $resultat["id_utilisateur"];

            $request = mysqli_query($bdd, "SELECT * FROM NOUNOU WHERE idNounou=".$resultat["id_utilisateur"]);
            if(mysqli_num_rows($request)==1){
                $_SESSION['categorie']='nounou';
                header( "string: Location: http://localhost:8888/html/board_nounou.php");
            }
            else{
                $_SESSION['categorie']='parent';
                $_SESSION['ville']= $resultat['ville'];
                header( "string: Location: http://localhost:8888/html/board_parent.php");
            }
        }
    }
    else{
        echo("<h2>Ces identifiants ne sont pas reconnus. Veuillez retourner au formulaire précédent.</h2>");
    }
}
```

Affichage dynamique des infos

Nous avons utilisé du **JavaScript** pour afficher seulement lorsque nécessaire les informations demandées. Pour cela, nous avons défini les classes des div concernées à **hidden** et fait appel aux scripts JavaScript pour appliquer la fonction d'affichage.

```
1 function affiche(nomId) {  
2  
3  
4  
5     let div = document.getElementById(nomId);  
6  
7     let activeDiv = document.getElementsByClassName("visible");  
8     let i = 0;  
9     while (activeDiv.length > i){  
10         activeDiv[i].setAttribute("class", "hidden");  
11         i++;  
12     }  
13  
14     div.setAttribute("class", "visible");  
15 }
```

Interface graphique

Pour l'aspect graphique de la plateforme, nous avons choisi de réaliser un template assez simple qui se répète sur les différentes pages (bandeau supérieur par exemple, contenants gris sur fond blanc...).



Bootstrap nous a permis de ne pas perdre trop de temps sur certains points de mise en forme tout en ayant un bon rendu. Nous l'avons utilisé pour l'aspect des formulaires, des boutons et d'autres fonctionnalités.

[Informations](#)[Se Connecter](#)

CONCEPTION DE LA BASE DE DONNÉES

La base de données est la première chose sur laquelle nous nous sommes concentrés afin de noter et répartir les informations dont nous allons avoir besoin. Nous avons déjà eu l'occasion de manipuler une base de données en suivant l'UE NF16, durant laquelle nous étions d'ailleurs aussi en binôme pour le projet. C'est la raison pour laquelle nous avons utilisé phpMyAdmin.

Nous avons premièrement repris le sujet du projet et les fonctionnalités demandées, puis nous avons imaginé les tables, leurs attributs et les contraintes de clés qui les reliaient.

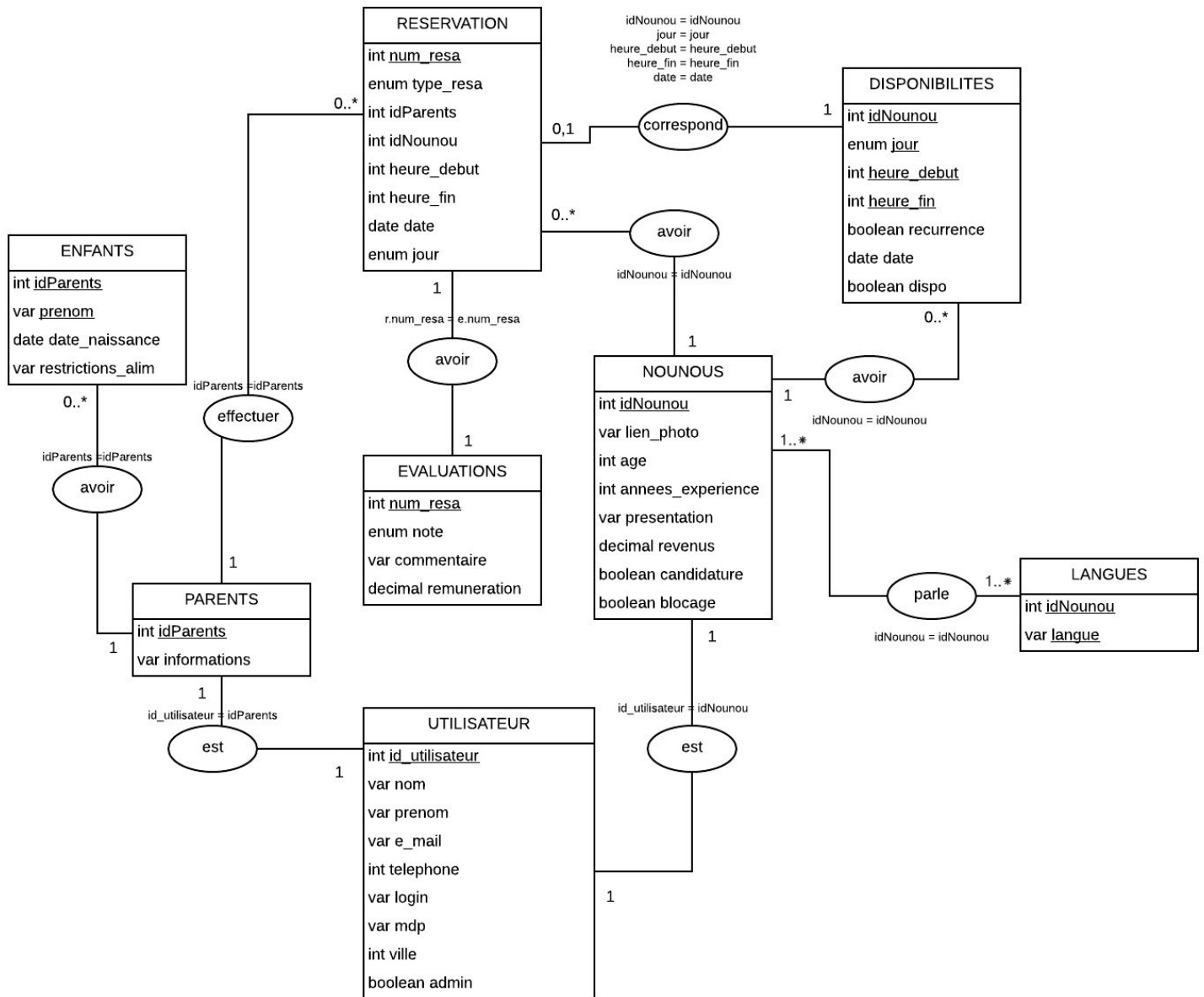
Bien évidemment la base de données à beaucoup évolué depuis cet instant car nous y avons apporté de nombreux changements. En effet, au fil de l'implémentation il a fallu modifier des tables en ajoutant, modifiant ou supprimant des attributs, ajoutant des contraintes de clé, des triggers etc..

Sur la page suivante vous trouverez notre schéma de base de données contenant les huit tables :

- UTILISATEURS
- NOUNOUS
- PARENTS
- ENFANTS
- RESERVATIONS
- EVALUATIONS
- DISPONIBILITES
- LANGUES

Les **clés primaires** de chaque tables sont précisées (elles sont soulignées) et les contraintes de **clés étrangères** sont également spécifiées et sont associées aux liens qui unissent certaines des tables entre elles.

Schéma de la base de données



Voir fichier SQL de la base de données dans ANNEXES.

DIFFICULTÉS RENCONTRÉES

Utilisation de Git

L'utilisation du gestionnaire de version Git nous a posé, à certains moments, des difficultés. Bertille n'ayant jamais utilisé ce logiciel, l'apprentissage des fonctionnalités de merge, de commit et la gestion des conflits a provoqué, à quelques reprises, des ralentissements dans la réalisation.

Nous avons cependant réussi à nous accorder sur notre utilisation du logiciel, et ne rencontrions plus de problèmes à la fin du projet.

Mise en place des classes php

L'utilisation des classes, bien qu'ayant simplifié certaines parties de notre code, a été complexe et n'a pas apporté la simplification attendue. En effet, chaque table de la base de données ayant des nombres d'attributs différents, nous ne sommes pas parvenus à créer une fonction générale permettant l'insertion dans une table d'un tuple, qui aurait ensuite été héritée à chaque table.

Nous ne sommes pas parvenus à créer des fonctions d'affichage au sein de ces classes, car chaque affichage nécessitait des paramètres différents, avec des contraintes bien spécifiques, et le système de classes ne nous permettait pas de répondre efficacement à ces problèmes.

Pour parvenir néanmoins à finaliser les fonctionnalités, nous avons donc codé certaines requêtes en dur dans les pages, ce qui alourdit quelque peu notre code.

CONCLUSION

Grâce à notre plateforme de réservation de nounous, plus de galère pour faire garder vos enfants, et même au dernier moment ! Nous avons respecté les contraintes de développement et de fonctionnalités, en se focalisant tout d'abord sur les choses que nous savions faire, puis sur ce qui nous était moins familier. Concernant notre binôme, nous n'avons pas rencontré de problème particulier concernant l'organisation et la répartition des tâches. Nous avons déjà été amenés à travailler ensemble l'année dernière, sur un projet de base de données notamment. La première chose fut de se concentrer sur la base de données (plutôt Bertille) ainsi que l'organisation générale du site et la création des différents fichiers dont nous allions avoir besoin (plutôt Robin).

Nous avons tous deux travaillé sur les scripts PHP, la création des formulaires et récupération de données. Robin s'est occupé de la partie concernant les sessions, ainsi que d'une bonne partie du code, résultant de sa bonne maîtrise des langages informatiques. Bertille s'est occupée des trigger et de nombreuses instructions PHP nécessitant d'insérer des requêtes SQL afin d'afficher les informations demandées. Github nous a permis de travailler également à distance, et donc plus efficacement mais nous nous retrouvions dans l'ensemble assez souvent pour faire le point sur l'avancée de notre travail et ce qu'il nous restait à faire. Dans l'ensemble tout s'est bien déroulé et nous sommes content de pouvoir présenter maNounou.com pleinement fonctionnel.

Évolution de nos compétences



| Niveau estimé (/10) | Avant LO07 | Après LO07 |
|---------------------|------------|------------|
| HTML/CSS | 5 | 6 |
| Bootstrap | 0 | 4 |
| Javascript | 1 | 3 |
| Php | 0 | 3 |
| MySQL | 2 | 5 |



| Niveau estimé (/10) | Avant LO07 | Après LO07 |
|---------------------|------------|------------|
| HTML/CSS | 8 | 8 |
| Bootstrap | 3 | 8 |
| Javascript | 5 | 5 |
| Php | 0 | 5 |
| MySQL | 1 | 4 |

N'ayant aucune connaissance préalable en PHP, L007 nous a transmis des connaissances approfondies dans la réalisation de sites web dynamiques. Il fut intéressant de mettre bout à bout les connaissances informatiques acquises à l'UTT, et de comprendre comment ces langages sont reliés entre eux, l'utilité qu'ils ont dans différentes situations, l'importance de la sécurité et d'une base de données bien réalisée. Ce projet nous a aussi permis de nous familiariser davantage avec les outils tels que phpMyAdmin, Github mais aussi Bootstrap, MAMP et le serveur Apache. Bien que plus de temps aurait été souhaitable pour pouvoir peut-être se pencher plus en profondeur sur l'aspect esthétique de la plateforme (ce qui n'était pas le but du projet), travailler sur ce projet nous aura été utile en terme d'acquisition de compétences et de gestion de travail de groupe. A bientôt sur **maNounou.com** !

ANNEXES

Script SQL de la base de données

```
-- phpMyAdmin SQL Dump
-- version 4.7.3
-- https://www.phpmyadmin.net/
```

```
--
```

```
-- Hôte : localhost:8889
-- Généré le : mar. 26 juin 2018 à 16:33
-- Version du serveur : 5.6.35
-- Version de PHP : 7.1.8
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
```

```
--
```

```
-- Base de données : `nounou`
```

```
--
```

```
-- -----
```

```
--
```

```
-- Structure de la table `DISPONIBILITES`
```

```
--
```

```
CREATE TABLE `DISPONIBILITES` (
  `idNounou` int(15) NOT NULL,
  `jour` enum('lundi','mardi','mercredi','jeudi','vendredi','samedi','dimanche') NOT NULL DEFAULT 'lundi',
  `heure_debut` int(11) NOT NULL,
  `heure_fin` int(11) NOT NULL,
```

```
`recurrence` tinyint(1) DEFAULT '0',  
`date` date DEFAULT NULL,  
`dispo` tinyint(1) NOT NULL DEFAULT '1'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- -----
```

```
--
```

```
-- Structure de la table `ENFANTS`
```

```
--
```

```
CREATE TABLE `ENFANTS` (  
  `idParents` int(15) NOT NULL,  
  `prenom` varchar(30) NOT NULL,  
  `date_naissance` date NOT NULL,  
  `restrictions_alim` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- -----
```

```
--
```

```
-- Structure de la table `EVALUATION`
```

```
--
```

```
CREATE TABLE `EVALUATION` (  
  `num_resa` int(11) NOT NULL,  
  `note` enum('0','1','2','3','4','5') NOT NULL,  
  `commentaire` varchar(300) NOT NULL,  
  `rémunération` decimal(65,0) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- -----  
  
--  
-- Structure de la table `LANGUES`  
--  
  
CREATE TABLE `LANGUES` (  
  `idNounou` int(15) NOT NULL,  
  `langue` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Déchargement des données de la table `LANGUES`  
--  
  
INSERT INTO `LANGUES` (`idNounou`, `langue`) VALUES  
(7632, 'arabe'),  
(7632, 'espagnol');  
  
-- -----  
  
--  
-- Structure de la table `NOUNOU`  
--  
  
CREATE TABLE `NOUNOU` (  
  `idNounou` int(15) NOT NULL,  
  `lien_photo` varchar(200) DEFAULT NULL,  
  `age` int(2) NOT NULL,  
  `annees_experience` int(2) NOT NULL,  
  `presentation` varchar(300) NOT NULL,
```

```
`revenus` decimal(65,0) DEFAULT NULL,  
`candidature` tinyint(1) NOT NULL DEFAULT '1',  
`blocage` tinyint(1) NOT NULL DEFAULT '0'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Liste des nounous';  
  
--  
-- Déchargement des données de la table `NOUNOU`  
--  
  
INSERT INTO `NOUNOU` (`idNounou`, `lien_photo`, `age`, `annees_experience`, `presentation`, `revenus`,  
`candidature`, `blocage`) VALUES  
(7632, 'famille.png', 18, 2, 'Je parle très bien italien, et j adore les enfants!', NULL, 0, 0);  
  
-----  
  
--  
-- Structure de la table `PARENTS`  
--  
  
CREATE TABLE `PARENTS` (  
  `idParents` int(15) NOT NULL,  
  `informations` varchar(300) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Déchargement des données de la table `PARENTS`  
--  
  
INSERT INTO `PARENTS` (`idParents`, `informations`) VALUES  
(7631, 'Nous sommes une famille agréable!');
```

```
--  
-- Structure de la table `RESERVATIONS`  
--  
  
CREATE TABLE `RESERVATIONS` (  
  `num_resa` int(15) NOT NULL,  
  `type_resa` enum('ponctuelle','etrangere','recu') NOT NULL,  
  `idParents` int(15) NOT NULL,  
  `idNounou` int(15) NOT NULL,  
  `heure_debut` int(11) NOT NULL,  
  `heure_fin` int(20) NOT NULL,  
  `date` date DEFAULT NULL,  
  `jour` enum('lundi','mardi','mercredi','jeudi','vendredi') DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Déchargement des données de la table `RESERVATIONS`  
--  
  
INSERT INTO `RESERVATIONS` (`num_resa`, `type_resa`, `idParents`, `idNounou`, `heure_debut`, `heure_fin`, `date`,  
`jour`) VALUES  
(5, 'ponctuelle', 7631, 7632, 12, 14, '2018-06-22', NULL),  
(6, 'recu', 7631, 7632, 9, 11, NULL, 'mardi'),  
(9, 'etrangere', 7631, 7632, 8, 23, '0000-00-00', NULL),  
(10, 'ponctuelle', 7631, 7632, 12, 14, '2018-06-22', 'lundi');  
  
--  
-- Déclencheurs `RESERVATIONS`  
--  
  
DELIMITER $$  
  
CREATE TRIGGER `dispo_after_insert` AFTER INSERT ON `RESERVATIONS` FOR EACH ROW BEGIN
```

```
UPDATE DISPONIBILITES d
SET d.dispo = 0
WHERE d.idNounou = NEW.idNounou AND d.date = NEW.date AND d.jour = NEW.jour AND d.heure_debut =
NEW.heure_debut;

END
$$
DELIMITER ;

-----

--
-- Structure de la table `UTILISATEURS`
--

CREATE TABLE `UTILISATEURS` (
  `id_utilisateur` int(15) NOT NULL,
  `nom` varchar(40) NOT NULL,
  `prenom` varchar(40) NOT NULL,
  `ville` int(6) NOT NULL,
  `email` varchar(40) NOT NULL,
  `telephone` int(12) NOT NULL,
  `login` varchar(40) NOT NULL,
  `mdp` varchar(200) NOT NULL,
  `admin` tinyint(1) DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Déchargement des données de la table `UTILISATEURS`
```


--

```
INSERT INTO `UTILISATEURS` (`id_utilisateur`, `nom`, `prenom`, `ville`, `email`, `telephone`, `login`, `mdp`, `admin`)
VALUES
```

```
(7630, 'Lallier', 'Robin', 10000, 'robin.lallier29@gmail.com', 662509816, 'jujube',
'$2y$10$if9CLGu45TVoTfcKm80Q4.JHkwVysZV/4OUvzJi8n7EobEiPkmw/.', 1),
```

```
(7631, 'Roger', 'Rolland', 10000, 'robin.lallier29@gmail.com', 662509816, 'litchi',
'$2y$10$0syc3ETby3LSDL.3YYGCwuymZ22YqO.rJP9kzVyUwTKXpGbfv2LE2', 0),
```

```
(7632, 'Julie', 'Lescault', 10000, 'robin.lallier@utt.fr', 662509816, 'mangue',
'$2y$10$T5K52li/5X7nYkAAKeb/teZzMGdPK50N4/n4YibA8vbjJAuSySAC', 0),
```

```
(7638, 'Mathilde', 'Jimenez', 10000, 'robin.lallier29@gmail.com', 662509816, 'poire',
'$2y$10$DB9yzBOQoYKPt09whrxxONHcrr0n41NXA3ADRFtM27Vb.M98uo9e', 0);
```

--

-- Index pour les tables déchargées

--

--

-- Index pour la table `DISPONIBILITES`

--

```
ALTER TABLE `DISPONIBILITES`
```

```
ADD PRIMARY KEY (`idNounou`, `jour`, `heure_debut`, `heure_fin`);
```

--

-- Index pour la table `ENFANTS`

--

```
ALTER TABLE `ENFANTS`
```

```
ADD PRIMARY KEY (`idParents`, `prenom`);
```

--

-- Index pour la table `EVALUATION`

--

```
ALTER TABLE `EVALUATION`
```

```
ADD PRIMARY KEY (`num_resa`);
```

```
--  
-- Index pour la table `LANGUES`  
--  
ALTER TABLE `LANGUES`  
  ADD PRIMARY KEY (`idNounou`,`langue`);  
  
--  
-- Index pour la table `NOUNOU`  
--  
ALTER TABLE `NOUNOU`  
  ADD PRIMARY KEY (`idNounou`);  
  
--  
-- Index pour la table `PARENTS`  
--  
ALTER TABLE `PARENTS`  
  ADD PRIMARY KEY (`idParents`);  
  
--  
-- Index pour la table `RESERVATIONS`  
--  
ALTER TABLE `RESERVATIONS`  
  ADD PRIMARY KEY (`num_resa`),  
  ADD KEY `fk_resaparent` (`idParents`),  
  ADD KEY `fk_resanounou` (`idNounou`);  
  
--  
-- Index pour la table `UTILISATEURS`  
--  
ALTER TABLE `UTILISATEURS`
```

```
ADD PRIMARY KEY (`id_utilisateur`);
```

```
--
```

```
-- AUTO_INCREMENT pour les tables déchargées
```

```
--
```

```
--
```

```
-- AUTO_INCREMENT pour la table `RESERVATIONS`
```

```
--
```

```
ALTER TABLE `RESERVATIONS`
```

```
MODIFY `num_resa` int(15) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

```
--
```

```
-- AUTO_INCREMENT pour la table `UTILISATEURS`
```

```
--
```

```
ALTER TABLE `UTILISATEURS`
```

```
MODIFY `id_utilisateur` int(15) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7639;
```

```
--
```

```
-- Contraintes pour les tables déchargées
```

```
--
```

```
--
```

```
-- Contraintes pour la table `DISPONIBILITES`
```

```
--
```

```
ALTER TABLE `DISPONIBILITES`
```

```
ADD CONSTRAINT `fk_disponounou` FOREIGN KEY (`idNounou`) REFERENCES `NOUNOU` (`idNounou`);
```

```
--
```

```
-- Contraintes pour la table `ENFANTS`
```

```
--
```

```
ALTER TABLE `ENFANTS`
```

```
ADD CONSTRAINT `fk_parentenfants` FOREIGN KEY (`idParents`) REFERENCES `PARENTS` (`idParents`);
```

```
--  
  
-- Contraintes pour la table `EVALUATION`  
  
--  
  
ALTER TABLE `EVALUATION`  
  ADD CONSTRAINT `fk_eval` FOREIGN KEY (`num_resa`) REFERENCES `RESERVATIONS` (`num_resa`);  
  
--  
  
-- Contraintes pour la table `LANGUES`  
  
--  
  
ALTER TABLE `LANGUES`  
  ADD CONSTRAINT `fk_nounou` FOREIGN KEY (`idNounou`) REFERENCES `NOUNOU` (`idNounou`);  
  
--  
  
-- Contraintes pour la table `NOUNOU`  
  
--  
  
ALTER TABLE `NOUNOU`  
  ADD CONSTRAINT `fk_idnounou` FOREIGN KEY (`idNounou`) REFERENCES `UTILISATEURS` (`id_utilisateur`);  
  
--  
  
-- Contraintes pour la table `PARENTS`  
  
--  
  
ALTER TABLE `PARENTS`  
  ADD CONSTRAINT `fk_idparent` FOREIGN KEY (`idParents`) REFERENCES `UTILISATEURS` (`id_utilisateur`);  
  
--  
  
-- Contraintes pour la table `RESERVATIONS`  
  
--  
  
ALTER TABLE `RESERVATIONS`  
  ADD CONSTRAINT `fk_resanounou` FOREIGN KEY (`idNounou`) REFERENCES `NOUNOU` (`idNounou`),  
  ADD CONSTRAINT `fk_resaparent` FOREIGN KEY (`idParents`) REFERENCES `PARENTS` (`idParents`);
```

