

Période : 20/05/2019 au 21/06/2019

# DOSSIER TECHNIQUE

*Programmation de l'application*



Maître de stage : Janet RAHARISON

Enseignant Référent : Jérôme HAVE



Arthur RICHARD

BTS SIO – 1ère année option SLAM

---

## SOMMAIRE

---

- I. Introduction
  - A. Remerciements
  - B. Présentation de l'organisation
  - C. Présentation du projet de stage
- II. Développement de l'appliquatif
  - A. Conception de la base de données
  - B. Le formulaire de réservation
  - C. Le module de visualisation des réservations
  - D. Le module de connexion à l'appliquatif
  - E. Le module de validation des réservations
- III. Documentation du projet
  - A. Guide d'utilisation de l'appliquatif pour les employés de mairie
  - B. Guide d'utilisation de l'appliquatif pour les relations publiques
- IV. Annexes
  - A. Documentations utilisées lors du développement
  - B. Documentations utilisées lors du déploiement
  - C. Organisation personnelle quotidienne

## I. Introduction

### A. Remerciements

Mes remerciements s'adressent en premier lieu à mon maître de stage, Monsieur Janet RAHARISON, Responsable du service des système d'information de la mairie de Sotteville-Lès-Rouen, pour sa confiance accordée et son ouverture d'esprit qui m'a permis de proposer des solutions personnelles dans le projet réalisé.

Je remercie également tous les membres du service des systèmes d'information pour l'équipement mis à disposition assurant le bon fonctionnement de mon stage et leurs précieux conseils.

Je congratulate par la même occasion le personnel d'accueil de la mairie de Sotteville-Lès-Rouen pour leurs avis partagés sur les solutions proposées.

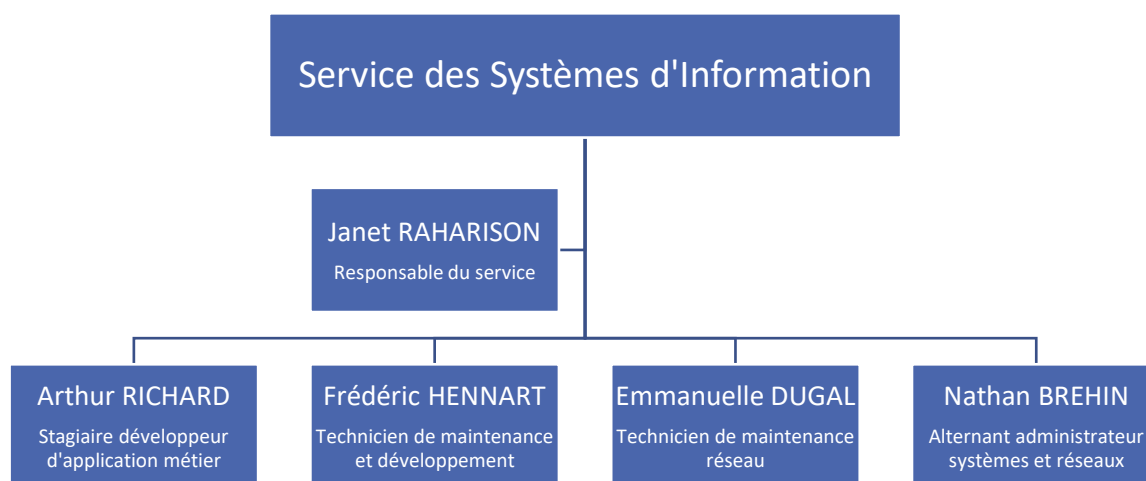
## B. Présentation de l'organisation

L'organisation qui s'occupe de mon stage est la mairie de Sotteville-Lès-Rouen. Le service des systèmes d'information de la mairie de Sotteville-Lès-Rouen est composé de deux techniciens et d'un alternant en DUT Informatique.

Les différentes missions du service sont les suivantes :

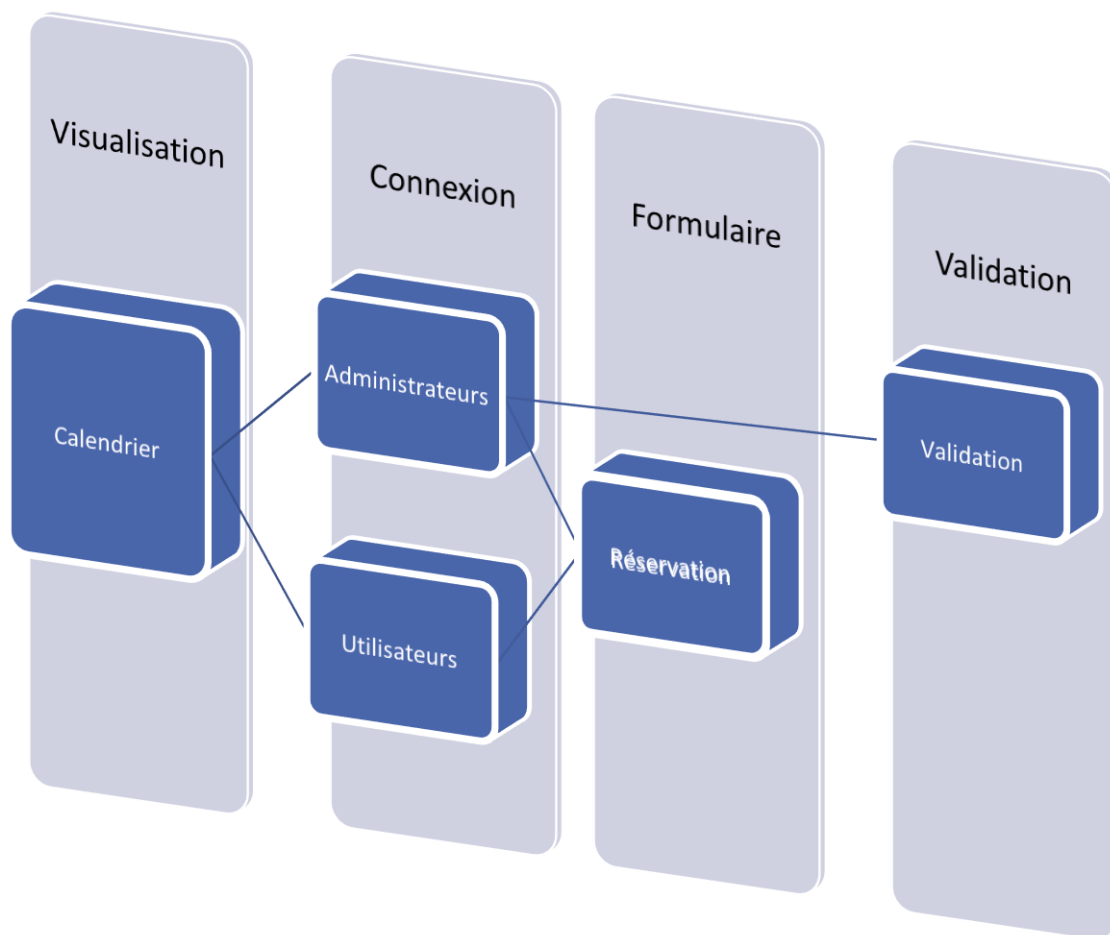
- Assurer l'accès à la technologie dans les différentes infrastructures publiques
- Assister les employés de mairie lors de problèmes techniques
- Rentrer en contact avec des prestataires extérieurs
- Réaliser la maintenance des équipements technologiques appartenant à la mairie
- Faire évoluer technologiquement des tâches professionnelles et applicatifs métiers

Organigramme du service des systèmes d'information :



### C. Présentation du projet de stage

Mon projet de stage consiste à digitaliser un formulaire de réservation de salle et de permettre la visualisation des réservations à tous les employés de mairie. La réservation de salle passe par la validation qui est une page accessible que par les relations publiques. Une fois validée, ces réservations sont affichées sur le module de visualisation. Voici la carte de l'application pour mieux comprendre les interactions :



Les administrateurs sont les employés du service des relations publiques et les utilisateurs sont tous les employés de mairie restant. A présent, nous allons voir quelles ont été les contraintes à respecter lors du projet.

### *Contrainte du formulaire de réservation :*

En effet, la mairie donne la possibilité à ses services municipaux de réserver des salles dans le but d'organiser une formation ou une réunion propre à un service ou non.

Pour réserver une salle, le demandeur doit en faire la demande sous forme de formulaire, qui doit être ensuite être validée par le service des relations publiques afin de pouvoir être autorisée.

Le formulaire indique le nom de l'intervenant, le nom du demandeur afin de pouvoir retracer la démarche de réservation, la date à laquelle la réservation aura lieu, l'heure de début ainsi que l'heure de fin. Il indique aussi le nom de la salle réservée, la disposition de cette salle ainsi que les prestations supplémentaires émises par le demandeur.

La disposition d'une salle concerne l'agencement du mobilier ainsi que l'équipement nécessaire au bon fonctionnement de la réservation. Parmi les équipements se trouve les ordinateurs, on peut demander au maximum 10 ordinateurs et on doit préciser si la connexion à Internet est voulue.

Le formulaire donne l'opportunité au demandeur de proposer une disposition inexistante ainsi qu'une annonce qui sera projeté sur l'écran d'accueil de la mairie. L'affichage doit être compréhensible et accessible par tous, dans le but de faciliter l'accès à la réservation.

Les prestations supplémentaires ne sont accessibles que pour les réservations de salle comprenant des participants extérieurs à la mairie ou des élus, ainsi les réservations de salle comprenant un nombre d'employés de mairie supérieur à 20 personnes et un créneau horaire situé dans la matinée.

### *Contrainte du module de connexion :*

Le module de connexion doit exploiter les identifiants et les mots de passe déjà existants des employés qui se trouve dans l'Active Directory de la mairie.

Lorsqu'un employé des relations publiques se connectera, il aura accès aux pleins droits sur la gestion des réservations de salle, ainsi qu'une page supplémentaire permettant la modération des validations des réservations de salle.

### *Contrainte de la visualisation :*

Le calendrier doit être visible par tout le monde, c'est-à-dire qu'il n'est pas nécessaire d'être connecté pour le consulter. Néanmoins, être connecté permet à l'utilisateur de voir des réservations personnalisées en fonction du service dans lequel il se trouve.

Le calendrier doit permettre la visualisation par mois, par semaine, et par jour heure par heure. Il doit être actualisé à chaque modification, ajout ou suppression d'une réservation.

*Contrainte du déploiement :*

L'application doit être hébergée sur un des serveurs de la mairie et doit être accessible depuis un nom de domaine choisi.

Le déploiement nécessite un environnement de test en amont afin de tester le bon fonctionnement de l'application à l'intérieur de réseau et pour éviter toutes détérioration de l'infrastructure déjà présente.

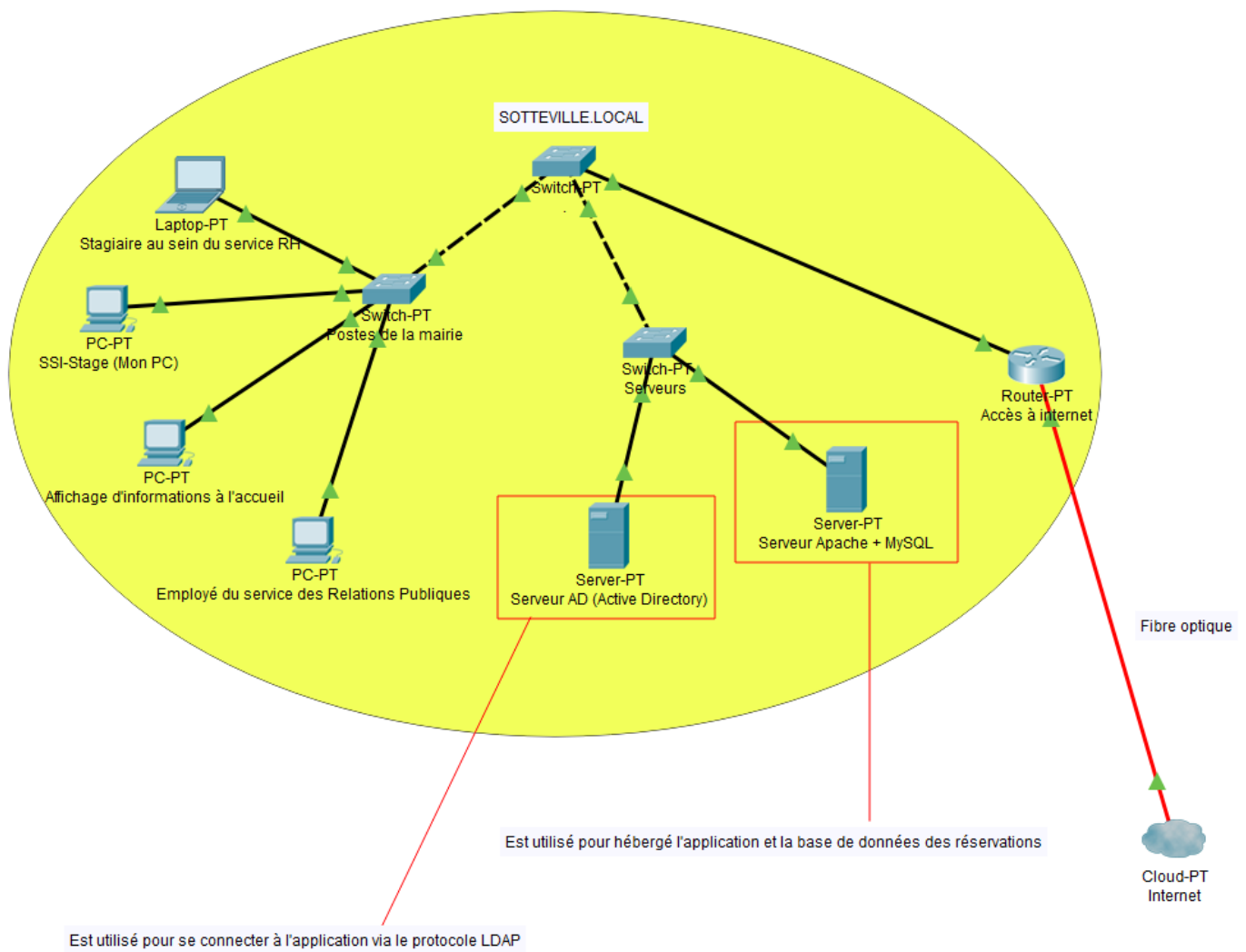
Tâches à réaliser	Méthode choisie	Justification
Développement de l'appliatif	PHP 7.3	<ul style="list-style-type: none"> <li>✓ Utilisé par 80% des sites web existants</li> <li>✓ Documentation variée</li> <li>✓ Permet de communiquer avec l'Active Directory en utilisant l'extension LDAP</li> <li>✓ Permet de communiquer avec MySQL en utilisant l'extension PDO</li> <li>✓ Langage déjà utilisé auparavant</li> </ul>
Stockage des données	MySQL	<ul style="list-style-type: none"> <li>✓ SGBDR Gratuit</li> <li>✓ Très utilisé</li> <li>✓ Documentation variée</li> <li>✓ Système déjà utilisé auparavant</li> </ul>
Hébergement de l'application	Apache 2.4	<ul style="list-style-type: none"> <li>✓ Fournit avec MySQL depuis XAMPP</li> <li>✓ Extension PHP déjà installée avec</li> <li>✓ Documentation variée</li> </ul>

Bien entendu, je vais utiliser **HTML5**, **CSS3** et **JavaScript** pour construire mon application.

Voici la liste des logiciels utilisés lors du projet :

- [Visual Studio Code](#) (Éditeur de code gratuit, léger et accessible)
- [XAMPP](#) (Serveur depuis le lieu de stage)
- [WampServer](#) (Serveur test depuis mon domicile)
- [Google Chrome](#) (Fournit de bons outils de développement web)
- [Mozilla Firefox](#) (Pour observer la différence d'un navigateur à l'autre)
- [Connexion Bureau à Distance](#) (Outil pour visualiser les chemins de l'Active Directory)
- [JMerise](#) (Pour faire la conception de la base de données)
- [Cisco Packet Tracer](#) (Pour faire la schématisation de l'architecture réseau)

Voici à quoi ressemble l'infrastructure du réseau que mon application va utiliser :

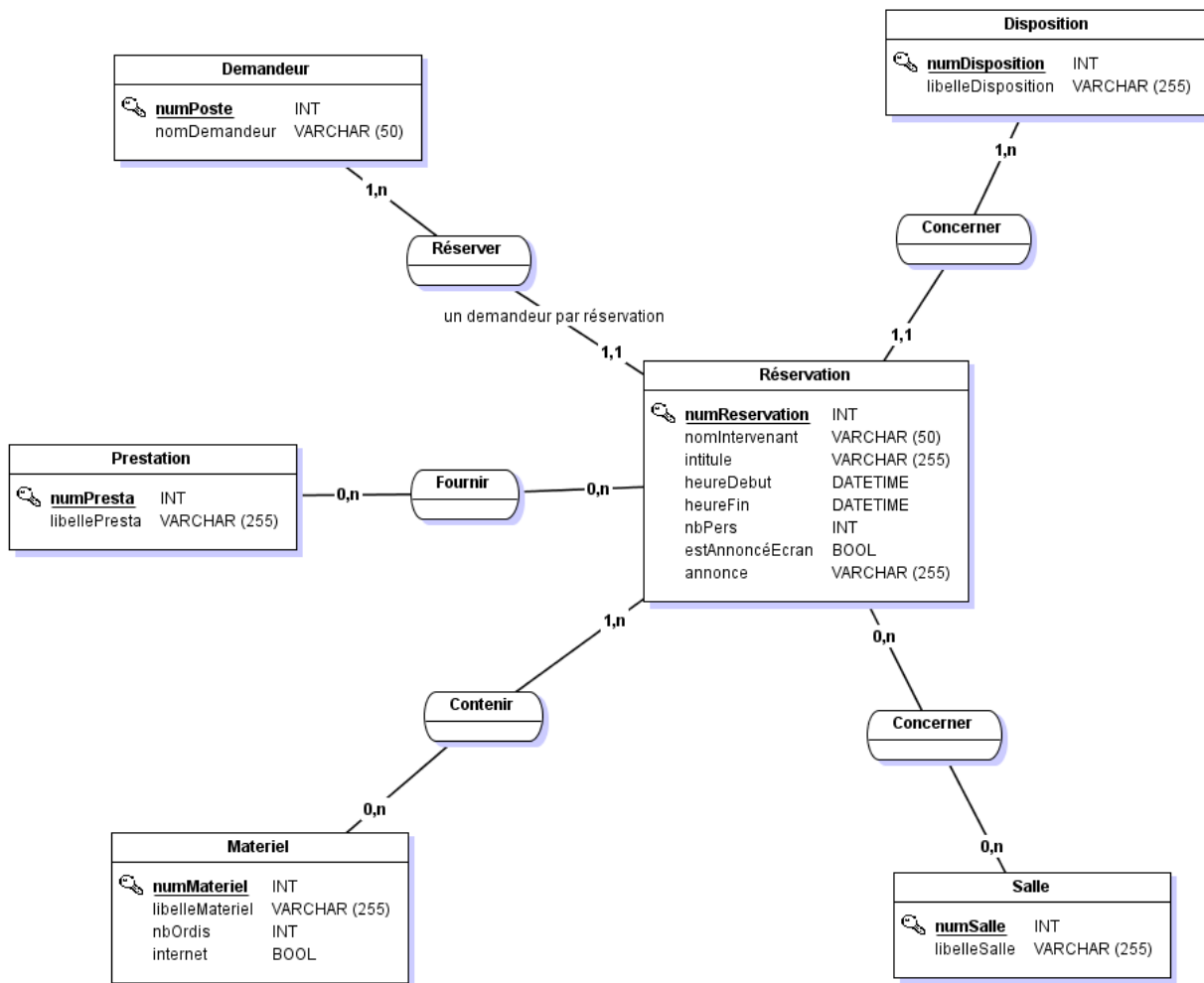




## II. Développement de l'appliatif

### A. Conception de la base de données

Pour concevoir ma base de données, j'ai dû d'abord m'appuyer sur le contexte donné ainsi que les contraintes discutées avec les employés de la mairie. Par la suite j'ai établi un modèle conceptuel de données (MCD) en utilisant la méthode MERISE. Voici ce que j'ai obtenu :



Ensuite, j'ai traduit ce MCD en MLD (Modèle relationnel de données), c'est-à-dire sous forme de script SQL car c'est ce modèle qui est utilisé par le système de gestion de base de données relationnel MySQL.

Voici à quoi ressemble le script SQL :

```
-- phpMyAdmin SQL Dump
-- version 4.8.5
-- https://www.phpmyadmin.net/
--
-- Hôte : localhost
-- Généré le : lun. 17 juin 2019 à 08:44
-- Version du serveur : 10.1.38-MariaDB
-- Version de PHP : 7.3.2

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données : `bddreserv`
--

--
-- Structure de la table `contenir`
--

CREATE TABLE `contenir` (
  `Id` smallint(6) NOT NULL,
  `IdReservation` smallint(6) NOT NULL,
  `IdMateriel` smallint(6) NOT NULL,
  `nbOrdis` int(11) NOT NULL,
  `internet` tinyint(1) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

--
-- Structure de la table `demandeur`
--

CREATE TABLE `demandeur` (
  `IdPoste` smallint(6) NOT NULL,
  `nomDemandeur` varchar(50) COLLATE utf8_bin DEFAULT NULL,
  `IdService` smallint(6) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

--
-- Structure de la table `disposition`
--

CREATE TABLE `disposition` (
  `Id` smallint(6) NOT NULL,
```

```
`Libelle` varchar(50) COLLATE utf8_bin DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

```
--
```

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

```
--
```

```
INSERT INTO `disposition` (`Id`, `Libelle`) VALUES
(1, 'Conférence'),
(2, 'Carré'),
(3, 'Tables rondes'),
(4, 'Buffet + quelques chaises');
```

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

```
--
```

```
-- Structure de la table `fournir`
```

```
--
```

```
CREATE TABLE `fournir` (
  `Id` smallint(6) NOT NULL,
  `IdReservation` smallint(6) NOT NULL,
  `IdPresta` smallint(6) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

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

```
--
```

```
-- Structure de la table `materiel`
```

```
--
```

```
CREATE TABLE `materiel` (
  `Id` smallint(6) NOT NULL,
  `Libelle` varchar(50) COLLATE utf8_bin DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

```
--
```

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

```
--
```

```
INSERT INTO `materiel` (`Id`, `Libelle`) VALUES
(1, 'Paperboard + stylo'),
(2, 'Ordinateur'),
(3, 'Vidéoprojecteur + écran'),
(4, 'Sono'),
(5, 'TV');
```

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

```
--
```

```
-- Structure de la table `prestation`
```

```
--
```

```
CREATE TABLE `prestation` (
  `Id` smallint(6) NOT NULL,
  `Libelle` varchar(50) COLLATE utf8_bin DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

```
--
```

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

```
--
```

```
INSERT INTO `prestation` (`Id`, `Libelle`) VALUES
(1, 'Café'),
(2, 'Thé'),
(3, 'Eau'),
(4, 'Buffet'),
(0, 'Rien');
```

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

```
--
-- Structure de la table `reservation`
--
```

```
CREATE TABLE `reservation` (
  `Id` smallint(6) NOT NULL,
  `nomIntervenant` varchar(50) COLLATE utf8_bin DEFAULT NULL,
  `start` datetime DEFAULT NULL,
  `end` datetime DEFAULT NULL,
  `title` varchar(255) COLLATE utf8_bin DEFAULT NULL,
  `typeReservation` varchar(50) COLLATE utf8_bin DEFAULT NULL,
  `typeParticipants` varchar(50) COLLATE utf8_bin DEFAULT NULL,
  `nbPers` int(11) DEFAULT NULL,
  `estAnnonceEcran` tinyint(1) DEFAULT NULL,
  `annonce` longtext COLLATE utf8_bin,
  `validation` tinyint(1) NOT NULL,
  `IdDemandeur` smallint(6) NOT NULL,
  `IdDisposition` smallint(6) NOT NULL,
  `IdSalle` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

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

```
--
-- Structure de la table `salle`
--
```

```
CREATE TABLE `salle` (
  `Id` smallint(6) NOT NULL,
  `libelle` varchar(50) COLLATE utf8_bin DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

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

```
INSERT INTO `salle` (`Id`, `libelle`) VALUES
(1, 'Salle des mariages (les 3 salles)'),
(2, 'Salon d\'Honneur'),
(3, 'Salle des Commissions'),
(4, 'Salle du Conseil Municipal (les 2 salles)'),
(5, 'Salle du Conseil'),
(6, 'Salle Louise A. Boyd'),
(7, 'Salle des Fêtes'),
(8, 'Salle Rosalind Franklin'),
(9, 'Salle Bleue'),
(10, 'Bureau d\'Accueil');
```

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

```
--
```

```
-- Structure de la table `service`
--

CREATE TABLE `service` (
  `Id` smallint(6) NOT NULL,
  `libelleService` varchar(50) COLLATE utf8_bin DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

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

INSERT INTO `service` (`Id`, `libelleService`) VALUES
(1, 'SSI'),
(2, 'RH'),
(12, 'CCAS'),
(13, 'POLICE'),
(14, 'CCAS'),
(16, 'RP'),
(17, 'CULTURE'),
(18, 'BIBLIOTHEQUE'),
(19, 'ADMINISTRATION GENERALE'),
(20, 'AAFP'),
(21, 'COMMUNICATION'),
(22, 'DIRECTION GENERALE'),
(23, 'FINANCES'),
(24, 'JEUNESSE'),
(25, 'APSM'),
(26, 'ECOLE DE MUSIQUE'),
(27, 'CABINET'),
(28, 'SYNDICATS'),
(29, 'SPORTS'),
(30, 'URBANISME'),
(31, 'VIVACITE'),
(32, 'SERVICES TECHNIQUES');

--
-- Index pour les tables déchargées
--

--
-- Index pour la table `contenir`
--
ALTER TABLE `contenir`
  ADD PRIMARY KEY (`Id`),
  ADD KEY `IdReservation` (`IdReservation`),
  ADD KEY `IdMateriel` (`IdMateriel`);

--
-- Index pour la table `demandeur`
--
ALTER TABLE `demandeur`
  ADD PRIMARY KEY (`IdPoste`),
  ADD KEY `IdService` (`IdService`);

--
-- Index pour la table `disposition`
--
ALTER TABLE `disposition`
  ADD PRIMARY KEY (`Id`);
```

```
--
-- Index pour la table `fournir`
--
ALTER TABLE `fournir`
  ADD PRIMARY KEY (`Id`),
  ADD KEY `IdReservation` (`IdReservation`),
  ADD KEY `IdPresta` (`IdPresta`);

--
-- Index pour la table `materiel`
--
ALTER TABLE `materiel`
  ADD PRIMARY KEY (`Id`);

--
-- Index pour la table `prestation`
--
ALTER TABLE `prestation`
  ADD PRIMARY KEY (`Id`);

--
-- Index pour la table `reservation`
--
ALTER TABLE `reservation`
  ADD PRIMARY KEY (`Id`),
  ADD KEY `IdDemandeur` (`IdDemandeur`) USING BTREE,
  ADD KEY `IdDisposition` (`IdDisposition`),
  ADD KEY `IdSalle` (`IdSalle`);

--
-- Index pour la table `salle`
--
ALTER TABLE `salle`
  ADD PRIMARY KEY (`Id`);

--
-- Index pour la table `service`
--
ALTER TABLE `service`
  ADD PRIMARY KEY (`Id`);

--
-- AUTO_INCREMENT pour les tables déchargées
--

--
-- AUTO_INCREMENT pour la table `contenir`
--
ALTER TABLE `contenir`
  MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=73;

--
-- AUTO_INCREMENT pour la table `demandeur`
--
ALTER TABLE `demandeur`
  MODIFY `IdPoste` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=23;

--
-- AUTO_INCREMENT pour la table `disposition`
--
ALTER TABLE `disposition`
```

```
MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=30;

--
-- AUTO_INCREMENT pour la table `fournir`
--
ALTER TABLE `fournir`
  MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;

--
-- AUTO_INCREMENT pour la table `materiel`
--
ALTER TABLE `materiel`
  MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT pour la table `prestation`
--
ALTER TABLE `prestation`
  MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

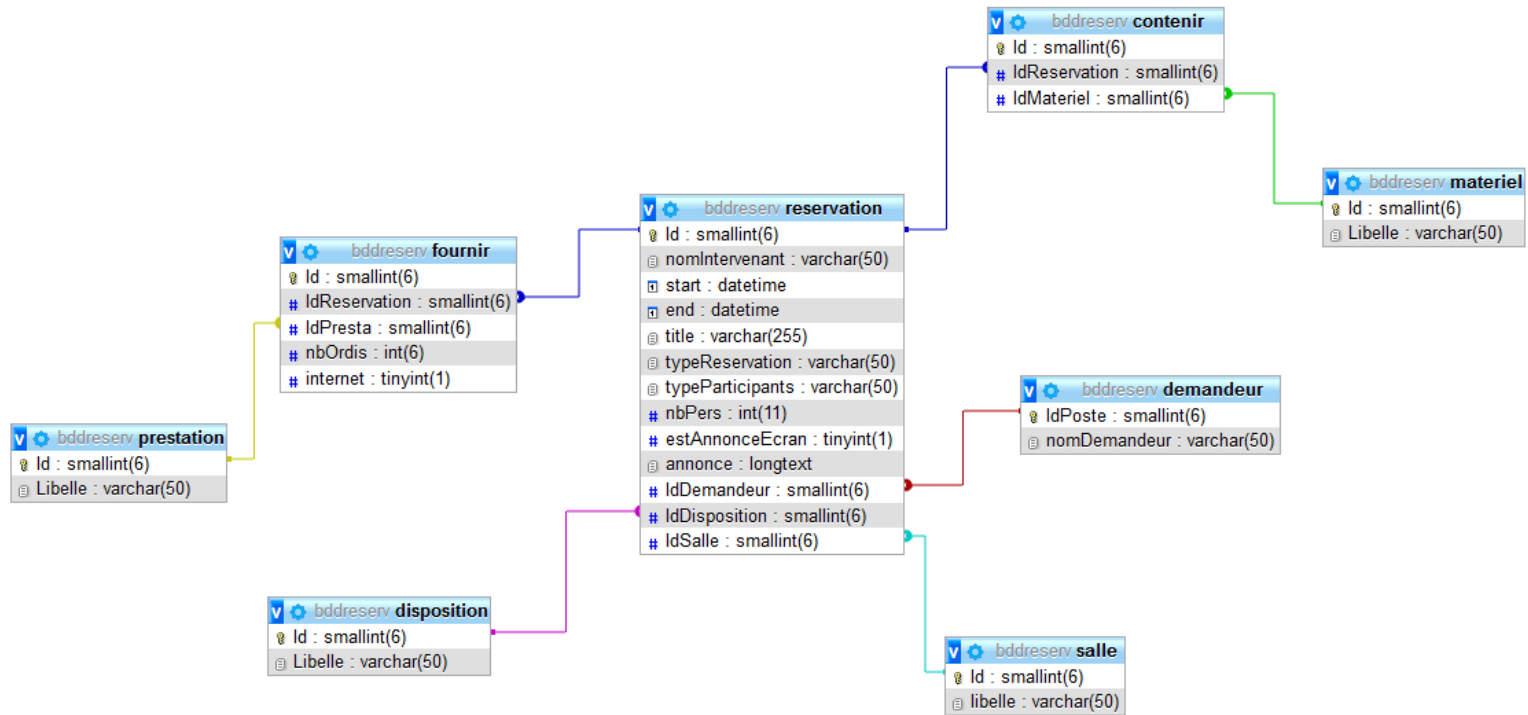
--
-- AUTO_INCREMENT pour la table `reservation`
--
ALTER TABLE `reservation`
  MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=274;

--
-- AUTO_INCREMENT pour la table `salle`
--
ALTER TABLE `salle`
  MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

--
-- AUTO_INCREMENT pour la table `service`
--
ALTER TABLE `service`
  MODIFY `Id` smallint(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=33;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Par la suite, on modifie ce script SQL afin de respecter les normes du modèle relationnel et on l'importe sur le serveur MySQL, et voici ce que l'on obtient lorsqu'on visualise la base de données depuis phpMyAdmin :



La base de données maintenant créée sous le nom de **bddreserv**, nous pouvons débuter le développement de l'application avec PHP7, HTML5, CSS3 et JavaScript.



## B. Le formulaire de réservation

Je débute le développement par la page la plus importante. Le formulaire de réservation existait déjà sous format papier, le reproduire n'est donc pas un problème mais il a fallu que je prenne en compte les différentes contraintes énoncées par les relations publiques et mon tuteur.

Voici à quoi ressemble le formulaire en format papier :

☐ T.C.  
☐ R.S.

Validé le : .... / .... / ....

**FICHE DE RESERVATION : SALLES DE L'HÔTEL DE VILLE**  
 (Réservée aux services Municipaux)

**En cas d'annulation, n'oubliez pas de prévenir l'accueil.**

**Merci de faire une réservation de salle par date**

**Votre demande sera définitivement validée qu'après retour par mail de l'accueil central**

**Demandeur**

Service : ..... Nom du demandeur : ..... N° de poste : .....

☐ Réunion ☐ Formation Nb de personnes : .....

Intitulé : .....

Date : ..... Heure de début : .....h..... Heure de fin : .....h.....

**Salle souhaitée**

☐ Salle des Mariages (les 3 salles)  
☐ Salon d'Honneur  
☐ Salle des Commissions  
☐ Salle des Fêtes  
☐ Bureau d'Accueil

☐ Salle du Conseil Municipal (les 2 salles)  
☐ Salle du Conseil  
☐ Salle Louise A. Boyd  
☐ Salle Rosalind Franklin  
☐ Salle Bleue

**Participants**

☐ Élus ☐ Personnel ☐ Extérieurs

Souhaitez-vous que cette réunion soit annoncée sur l'écran d'informations d'accueil ?

☐ Oui ☐ Non

Si oui, merci de rédiger cette annonce de la façon la plus synthétique possible :

---

**DEMANDES LOGISTIQUE**

**Disposition de la salle :**

☐ Conférence ☐ Carré ☐ Tables rondes ☐ Buffet + quelques chaises

☐ Autre disposition (Précisez) : .....

**Matériel souhaité :**

☐ Paperboard + stylo ☐ Ordinateur ☐ Vidéoprojecteur + écran ☐ Sono ☐ TV

**AUTRES PRESTATIONS**

☐ Café ☐ Thé ☐ Eau ☐ Buffet (si manifestations)

**Si autre demande, la faire par mail auprès du service Relations Publiques.**

En premier lieu, j'ai commencé par reproduire en HTML les entrées du formulaire.

```
<div class="card reservation">
  <form action="reservation.php" method="POST">
    <div class="card-header w3-grey"><b>Intervenant</b></div>
    <div class="card-body">
      Intervenant<span>*</span> : <input type="text" name="intervenant" id="textIntervenant" required>
      <input type="radio" name="typeReservation" value="reunion" id="radioReunion" required> Réunion
      <input type="radio" name="typeReservation" value="formation" id="radioFormation" required> Formation
      Nombre de personnes<span>*</span> : <input class="col-2" type="number" name="nbPers" id="nbPers" required>
      Intitulé : <input class="col-12" type="text" name="intitule" id="intitule" required>
      Date : <input type="date" name="date" id="date" required>
      Heure de début : <input type="time" name="HdeDebut" id="heureDebut" required>
      Heure de fin : <input type="time" name="HdeFin" id="heureFin" required>
    </div>
  </form>
</div>
```

Lorsque le formulaire est envoyé, les données reçues et testées sont directement insérées dans une requête SQL à l'aide d'une fonction PHP.

```
if (isset($_POST)) {
    $demandeur = $_SESSION['user'];
    $intervenant = $_POST['intervenant'] ?? null;
    $typeReservation = $_POST['typeReservation'] ?? null;
    $nbPers = $_POST['nbPers'] ?? null;
    $intitule = $_POST['intitule'] ?? null;
    $date = $_POST['date'] ?? null;
    $HdeDebut = $_POST['HdeDebut'] ?? null;
    $HdeFin = $_POST['HdeFin'] ?? null;
    $salle = $_POST['salle'] ?? null;
    $typeParticipant = $_POST['typeParticipant'] ?? null;
    if ($typeParticipant == "ext") {
        $presta = $_POST['presta'] ?? null;
    } else {
        $presta = [];
    }
    if (isset($_POST['matériel'])) {
        $matériel = $_POST['matériel'] ?? null;
    } else {
        $matériel = array();
    }
    for ($i = 0; $i < sizeof($matériel); $i++) {
        if ($matériel[$i] == "pc") {
            $nbOrdre = $_POST['need'] ?? null;
            $internet = $_POST['internet'] ?? null;
        }
    }
    $ecran = $_POST['ecran'] ?? null;
    if ($ecran == null || $ecran == "non") {
        $boolEcran = 0;
        $annonce = "pas de message";
    } else {
        $boolEcran = 1;
        $annonce = $_POST['annonce'] ?? null;
    }
    $disposition = $_POST['disposition'] ?? null;
    if ($disposition == "autre") {
        $autreText = $_POST['autreText'] ?? null;
        InsertDisposition($autreText);
        $IdDispo = getDispositionId($autreText);
    } else {
        $autreText = null;
        $IdDispo = getDispositionId($disposition);
    }
    if (isset($disposition)) {
        $combinedD1 = date('Y-m-d H:i:s', strtotime("$date $HdeDebut"));
        $combinedD2 = date('Y-m-d H:i:s', strtotime("$date $HdeFin"));
        Reservation($intervenant, $typeReservation, $combinedD1, $combinedD2, $intitule, $typeParticipant, $nbPers, $boolEcran, $annonce, $salle, 1, getDispositionId($disposition));
        $status = "done";
        InsertMatériel(getReservationId($date, $intitule), $matériel, $nbOrdre, $internet);
        InsertPresta(getReservationId($date, $intitule), $_POST['presta']);
    } else {
        $status = "failed";
    }
}
```

Voici un aperçu des fonctions PHP utilisées pour l'insertion dans la base de données :

```
function Reservation($intervenant,$typeReservation,$HdeDebut,$HdeFin,$intitule,$typeParticipant,$nbPers,$boolEcran,$annonce,$salle,$demandeur,$IdDispo){
    $bdd = getDatabase();
    $IdDispo=1;
    $sqlReservation = "INSERT INTO `reservation`(`nomIntervenant`, `start`, `end`, `title`, `typeReservation`, `typeParticipants`, `nbPers`,
    `estAnnonceEcran`, `annonce`, `IdDemandeur`, `IdDisposition`, `IdSalle`) VALUES
    (:intervenant,:hdedebut,:hdefin,:intitule,:typereservation,
    :typeparticipant,:nbpers,:boolecran,:annonce,:demandeur,:dispo,:salle)";
    $reqReservation = $bdd->prepare($sqlReservation);
    $reqReservation->bindParam(":intervenant",$intervenant);
    $reqReservation->bindParam(":typereservation",$typeReservation);
    $reqReservation->bindParam(":hdedebut",$HdeDebut);
    $reqReservation->bindParam(":intitule",$intitule);
    $reqReservation->bindParam(":typeparticipant",$typeParticipant);
    $reqReservation->bindParam(":nbpers",$nbPers);
    $reqReservation->bindParam(":hdefin",$HdeFin);
    $reqReservation->bindParam(":boolecran",$boolEcran);
    $reqReservation->bindParam(":annonce",$annonce);
    $reqReservation->bindParam(":salle",$salle);
    $reqReservation->bindParam(":demandeur",$demandeur);
    $reqReservation->bindParam(":dispo",$IdDispo);
    $reqReservation->execute();
    $reqReservation->closeCursor();
}

function InsertDisposition($autreText){
    $bdd = getDatabase();
    $sqlDisposition = "INSERT INTO `disposition`(`Libelle`) VALUES (:autretext)";
    $reqDisposition = $bdd->prepare($sqlDisposition);
    $reqDisposition->bindParam(":autretext",$autreText); // Ajoute la nouvelle disposition dans la BdD
    $reqDisposition->execute();
    $reqDisposition->closeCursor();
}

function InsertMateriel($IdReservation,$arrayMateriel,$nbordis,$internet){
    $bdd = getDatabase();
    foreach ($arrayMateriel as $idMateriel) {
        if ($idMateriel == 2) {
            $sqlMateriel = "INSERT INTO `contenir`(`IdReservation`, `IdMateriel`, `nbOrdis`, `internet`) VALUES (:reservation,:materiel,:nbordis,:internet)";
            $reqMateriel = $bdd->prepare($sqlMateriel);
            $reqMateriel->bindParam(":reservation",$IdReservation);
            $reqMateriel->bindParam(":materiel",$idMateriel);
            $reqMateriel->bindParam(":nbordis",$nbordis);
            $reqMateriel->bindParam(":internet",$internet);
            $reqMateriel->execute();
            $reqMateriel->closeCursor();
        }
        else {
            $sqlMateriel = "INSERT INTO `contenir`(`IdReservation`, `IdMateriel`) VALUES (:reservation,:materiel)";
            $reqMateriel = $bdd->prepare($sqlMateriel);
            $reqMateriel->bindParam(":reservation",$IdReservation);
            $reqMateriel->bindParam(":materiel",$idMateriel);
            $reqMateriel->execute();
            $reqMateriel->closeCursor();
        }
    }
}

function InsertPresta($IdReservation,$arrayPresta){
    $bdd = getDatabase();
    foreach ($arrayPresta as $presta) {
        $sqlPresta = "INSERT INTO `fournir`(`IdReservation`, `IdPresta`) VALUES (:reservation,:presta)";
        $reqPresta = $bdd->prepare($sqlPresta);
        $reqPresta->bindParam(":reservation",$IdReservation);
        $reqPresta->bindParam(":presta",$presta);
        $reqPresta->execute();
        $reqPresta->closeCursor();
    }
}
```

Ces trois fonctions permettent d'insérer des données dans la base de données **bddreserv**. Voici la fonction qui permet d'appeler la BdD :

```
function getDatabase($user="root", $pass=""){ // à changer à l'avenir lors de l'installation du serveur MySQL
    $hostname="localhost";
    $port=3306;
    $dbname="BdDReserv";
    try {
        $pdo = new PDO("mysql:host=$hostname;port=$port;dbname=$dbname;charset=utf8", $user, $pass);
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        return $pdo;
    } catch (Exception $e) {
        $msg = 'ERREUR PDO dans ' . $e->getFile() . ' L.' . $e->getLine() . ' : ' . $e->getMessage();
        die($msg);
    }
}
```

Voilà à quoi ressemble le formulaire de réservation après l'habillage de la page choisie :

The screenshot shows a web application interface for room reservations. At the top, there's a navigation bar with 'Visualiser le planning' and a user status 'Bonjour user\_ssi ! Déconnexion'. The main form is titled 'FICHE DE RÉSERVATION - Salles de l'Hôtel de ville' and is noted as '(Réservée aux services Municipaux)'. It includes instructions on cancellation and a note to reserve by date. The form is divided into several sections: 'Intervenant' (with fields for Intervenant, Demandeur, type of event, number of people, service, and date/time), 'Salle souhaitée' (a list of rooms with radio buttons), 'Participants' (with radio buttons for participant type and a question about announcements), and 'Demandes logistique' (with checkboxes for equipment like paperboard, computer, and projector). The background of the form is a blurred image of a modern office interior with large windows and a potted plant.

**Visualiser le planning** Bonjour user\_ssi ! [Déconnexion](#)

### FICHE DE RÉSERVATION - Salles de l'Hôtel de ville

(Réservée aux services Municipaux)

En cas d'annulation, n'oubliez pas de prévenir l'accueil.  
[Merci de faire une réservation de salle par date](#)  
 \* : champs obligatoires

---

**Intervenant**

Intervenant\* : Janet RAHARISON      Demandeur\* (administratif) : Arthur RICHARD  
☐ Réunion    ☒ Formation  
 Nombre de personnes\* : 200      Service : SSI  
 Intitulé :  
 Comment avoir des mots de passe sécurisés ?  
 Date : 13/06/2019    Heure de début : 14:30    Heure de fin : 16:30

---

**Salle souhaitée**

☐ Salle des mariages (les 3 salles)  
☒ Salon d'Honneur  
☐ Salle des Commissions  
☐ Salle du Conseil Municipal (les 2 salles)  
☐ Salle du Conseil  
☐ Salle Louise A. Boyd  
☐ Salle des Fêtes  
☐ Salle Rosalind Franklin  
☐ Salle Bleue  
☐ Bureau d'Accueil

---

**Participants**

☐ Élus    ☒ Personnel    ☐ Extérieurs  
 Souhaitez-vous que cette réunion soit annoncée sur l'écran d'informations d'accueil ?  
☐ Oui    ☒ Non  
 Si oui, merci de rédiger cette annonce de la façon la plus synthétique possible :

---

**Demandes logistique**

Disposition de la salle\* :  
☒ Conférence    ☐ Carré    ☐ Tables rondes    ☐ Buffet + quelques chaises  
☐ Autre disposition (Précisez) :  
 Matériel souhaité :  
☐ Paperboard + stylo  
☐ Ordinateur  
 Nombre d'ordinateurs nécessaires : (Maximum 10)   
 Usage d'internet : ☐ Oui ☐ Non  
☒ Vidéoprojecteur + écran  
☐ Son

Le formulaire est désormais fonctionnel et permet d'insérer des données, il faut maintenant créer des contraintes d'affichage afin de rendre le formulaire cohérent avec le contexte.

**Première contrainte :** Le formulaire doit être évolutif en fonction des choix de l'utilisateur, c'est-à-dire que certaines parties du formulaire s'afficheront en fonction des réponses précédemment données.

**Solution apportée :** Grâce à JavaScript et aux nouveaux attributs d'HTML5, on peut cacher des balises html en leur attribuant `hidden`, puis avec JavaScript, je décide ou non de ne plus attribuer `hidden` en fonction des valeurs remplies comme dans l'exemple présent :

```
function Affiche() {
// Fonction qui permet l'affichage progressif du formulaire afin de simplifier son
if (radioReunion.checked||radioFormation.checked) {
// les inputs radio/checkbox et les inputs text/number sont séparés --> Sinon valeur de vérité altérée
if (service.value!="&&textIntervenant.value!="&&textDemandeur.value!="&&nbPers!="&&intitule.value!="&&heureDebut.value!="&&heureFin.va
CH_salle.hidden = false;
CB_salle.hidden = false;
if (radioSalle1.checked||radioSalle2.checked||radioSalle3.checked||radioSalle4.checked||radioSalle5.checked||radioS
CH_participant.hidden = false;
CB_participant.hidden = false;
if (CbElu.checked||CbExt.checked) {
if (radioNonEcran.checked) {
CH_logistique.hidden = false;
CB_logistique.hidden = false;
if (cbConf.checked||cbCarre.checked||cbRonde.checked||cbBuffetDispo.checked) {
// Les prestations sont accessibles pour les élus et les participants extérieurs à n'importe quelle heure de la journée
CH_presta.hidden = false;
CB_presta.hidden = false;
if (CbCafe.checked||CbTea.checked||CbEau.checked||CbBuffet.checked) {
// Prestations(s) choisie(s), fin du formulaire
btnSubmit.type = "submit";
btnSubmit.class = "btn btn-success";
btnSubmit.value = "Envoyer le formulaire";
btnSubmit.onclick = Confirm();
}
}
}
```

Par la suite, on attribue au bouton « continuer » cette fonction, ce bouton pourra être cliquable à chaque fin de remplissage du formulaire, et lorsque toutes les valeurs possible du formulaire sont remplis, le type du bouton change et devient un bouton de type « submit » qui envoie le formulaire.

```
<input class="btn btn-secondary" onclick="Affiche()" id="btnForm" type="button" value='Continuer'>
```

La nouvelle fonction attribuée n'est donc plus 'Affiche()' mais 'Confirm()' qui envoie une boîte de dialogue avec comme choix « OK » ou « Annuler » afin de fournir à l'utilisateur un moyen de revenir en arrière si les champs remplis ne conviennent pas.

**Deuxième contrainte :** la saisie du formulaire n'est accessible que par les employés de la mairie qui sont connecté afin de les identifier

**Solution apportée :** J'ai réalisé une fonction permettant de vérifier si une session est en cours avec PHP, si une session n'est pas en cours, l'utilisateur est redirigé vers le module de connexion avec une alerte JavaScript spécifiant qu'il doit être authentifié pour remplir le formulaire. Voici à quoi ressemble la fonction :

```
function isAuth(){
    return isset($_SESSION['user']);
}
function needAuth(){
    if (!isAuth()){
        $_SESSION['error_bypass']="Vous devez être connecté pour accéder à cette page !";
        header('Location: /appReservation/index.php'); // redirige vers une la page de connexion
        die();
    }
}
```

Un fichier errors.php permet de créer des alertes JavaScript en fonction du type d'erreur retournée

```
if (isset($_SESSION['error_bypass'])) {
    ?>
    <script>
        alert("Vous devez être connecté pour accéder à cette page !");
    </script>
    <?php
        session_destroy();
}
```

Ce fichier est importé entre le footer et la balise de fermeture html de la page de connexion de l'application pour faire apparaître un script

```
</footer>
<?php require "../appReservation/includes/errors.php"?>
</html>
```

On importe les fonctions dans le formulaire de réservation pour importer les fonctions requises.

```
reservation.php ▸ ...
<?php
require "../includes/functions.php";
startSession();
needAuth();
```

## C. Le module de visualisation des réservations

La solution retenue pour visualiser les réservations est un calendrier. Il existe déjà de nombreux calendriers sur Internet, nous n'allons donc pas tout réinventer afin d'optimiser le temps du stage.

Le calendrier retenu est fullcalendar.js, C'est un calendrier dynamique codé en JQuery ou JavaScript au choix, il est responsive et facile à intégrer. Vous pouvez obtenir plus d'informations sur <https://fullcalendar.io/>.

La méthode choisie a été d'importer les réservations depuis la base de données qui ont été intégrés au calendrier sous forme d'événements à l'aide de PHP et de JQuery. Je vais aussi utiliser Moment.js qui est une librairie permettant la traduction des horaires générés par fullcalendar.js d'être traduit en français, mais aussi de reformater des dates selon le besoin.

```
function event()
{
    // fonction permettant d'importer les données concernant les réservations dans le calendrier
    $bdd = getDatabase();
    $query = "SELECT validation, nomDemandeur, reservation.Id, disposition.Libelle, typeParticipants, typeReserv";
    $resultat = $bdd->query($query);
    $infos = $resultat->fetchAll();
    $resultat->closeCursor();
    return $infos;
}
```

Voici comment j'intègre le calendrier avec ma base de données et PHP :

```
document.addEventListener('DOMContentLoaded', function() {
    var initialLocaleCode = 'fr';
    var localeSelectorEl = document.getElementById('locale-selector');
    var calendarEl = document.getElementById('calendar');
    var calendar;

    initThemeChooser({
        init: function(themeSystem) {
            calendar = new FullCalendar.Calendar(calendarEl, {
                plugins: ['bootstrap', 'interaction', 'dayGrid', 'timeGrid', 'list', 'googleCalendar'],
                themeSystem: themeSystem,
                header: {
                    left: 'prev,next today',
                    center: 'title',
                    right: 'dayGridMonth,timeGridWeek,timeGridDay,listMonth'
                },
                defaultDate: Date.now(),
                defaultView: 'timeGridWeek',
                locale: initialLocaleCode,
                weekNumbers: true,
                minTime: "07:00:00",
                navLinks: true,
                <?php if (isset($_SESSION['user']) && giveRights($_SESSION['user'])) {
                    echo 'editable: true;';
                    echo 'droppable: true;';
                } else {
                    echo 'editable: false;';
                }
            });
            eventLimit: true,
            events: [
                <?php foreach (event() as $infos) {
                    $stuff = getMateriel($infos['Id']);
                    $prestations = getPresta($infos['Id']);
                    if ($infos['validation'] == 0) {
                        >> {
                            title: "NON VALIDÉ" + "\n" + "<?php echo $infos['typeReservation'] ?>" + "\n" + "<?php echo $infos['title'] ?>" + "\n" + "\n" + "<?php echo $infos['nonIntervenant'] ?>" + "\n" + "<?php ec";
                            start: "<?php echo $infos['start'] ?>";
                            end: "<?php echo $infos['end'] ?>";
                            color: 'lightgrey';
                            textColor: 'black';
                            borderColor: 'red';
                        },
                    <?php } else { >> {
                        title: "<?php echo $infos['typeReservation'] ?>" + "\n" + "<?php echo $infos['title'] ?>" + "\n" + "\n" + "<?php echo $infos['nonIntervenant'] ?>" + "\n" + "<?php echo $infos['libelle']";
                        start: "<?php echo $infos['start'] ?>";
                        end: "<?php echo $infos['end'] ?>";
                        <?php if ($infos['libelle'] == "Salle des mariages (les 3 salles)") { >>

```



```

borderColor: 'red'
},
<?php } else { ?> {
    title: "<?php echo $infos['typeReservation'] ?>" + "\n" + "<?php echo $infos['title'] ?>" + "\n" + "\n" + "<?php echo $infos['nomIntervenant'] ?>" + "\n" + "<?php echo $infos['libelle'] ?>"
    start: "<?php echo $infos['start'] ?>",
    end: "<?php echo $infos['end'] ?>",
    <?php if ($infos['libelle'] == "Salle des mariages (les 3 salles)") { ?>
        color: "pink",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salon d'Honneur") { ?>
        color: "gold",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle des Commissions") { ?>
        color: "forestgreen",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle du Conseil Municipal (les 2 salles)") { ?>
        color: "red",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle du Conseil") { ?>
        color: "purple",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle Louise A. Boyd") { ?>
        color: "lightblue",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle des Fêtes") { ?>
        color: "orange",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle Rosalind Franklin") { ?>
        color: "teal",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle Bleue") { ?>
        color: "blue",
    <?php } ?>
    <?php if ($infos['libelle'] == "Bureau d'Accueil") { ?>
        color: "wheat",
    <?php } ?>
    },
    <?php }
} ?>
},
1,
eventClick: function(info) {
    var debut = moment(info.event.start).format('LLLL');
    var fin = moment(info.event.end).format('LLLL'); // pas de risques pour la prochaine mise à jour de moment.js car le format de la date est YYYY-MM-DD HH:mm
    moment(fin).locale('fr');
    moment(debut).locale('fr');
    var str = info.event.title + "\n" + "Horaire de début : " + debut + "\n" + "Horaire de fin : " + fin;
    alert(str);
},

```

```

eventMouseEnter: function(info) {
    <?php
    if (isset($_SESSION['user']) && giveRights($_SESSION['user'])) {
        ?>
        var contextmenu = new ContextMenu();
        var LinkModif = document.getElementById("hrefModif");
        var LinkSuppr = document.getElementById("hrefSuppr");
        if (LinkModif.clicked) {
            LinkModif.href = "./pages/modification.php?Id="+info.el.id;
        }
        if (LinkSuppr.clicked) {
            LinkSuppr.href = "./pages/suppression.php?Id="+info.el.id;
        }
        <?php } ?>
    },
    eventRender: function(info) {
        <?php foreach (event() as $value) {
            $Id = $value['Id'];
            echo "info.el.id = $Id";
        }
        ?>
    }
});

calendar.render();
},

```



Par la suite, je me suis entretenu avec les relations publiques et mon tuteur pour intégrer un module de modification de la base de données depuis l'application, ce module étant accessible qu'aux superutilisateurs permettra une gestion des réservations coté client.

La gestion des réservations coté client inclue la possibilité de supprimer une réservation et de modifier son contenu.

Les données textuelles seront modifiées depuis un formulaire qui s'ouvrira quand un utilisateur ayant les droits fera un clic droit sur une réservation et les horaires seront directement modifiables en drag and drop depuis le calendrier uniquement pour les utilisateurs ayant les droits aussi.

Le formulaire de modification est en tout point identique à celui de réservation sauf pour les horaires car la modification des horaires ne passe donc pas par un formulaire.

Pour que le client interagisse avec le serveur sans passer par un formulaire, nous devons utiliser Ajax de JQuery, cela permet de communiqué avec la base de données coté client en toute sécurité pour éviter une faille XSS.

```

eventDrop: function(info) {
    var dow = info.event.daysOfWeek;
    var start = info.event.start;
    var end = info.event.end;
    var allDay = info.event.allDay;
    start = moment(start).toISOString();
    end = moment(end).toISOString();
    moment(start).locale('fr');
    moment(end).locale('fr');
    var heure = start.getHours();
    var start = start.setHours(heure + 2);
    if (allDay==true) {
        allDay=1;
    } else {
        allDay=0;
    }
    if (dow==null) {
        dow=0
    }
    var Id = info.event.id;
    console.log(start);
    console.log(info.oldEvent.end);
    console.log(end);
    console.log(dow);
    console.log(allDay);
    console.log(Id);
    $.ajax({
        url: './includes/update_time.php',
        method: 'post',
        data: {dow : dow , start : start , end : end, allDay : allDay , Id : Id},
        success: function(response) {
            var alertYes = "<div id='alertYes' class='alert alert-success alert-dis";
            $("#calendar").append(alertYes);
            $("#alertNo").remove();
            console.log(response);
        },
        error: function(response) {
            var alertNo = "<div id='alertYes' class='alert alert-success alert-dis";
            $("#calendar").append(alertNo);
            $("#alertYes").remove();
            console.log(response);
        }
    })
}, /*

```

La modification des horaires entraine vers une page qui détecte les nouvelles données et les mets à jour dans la base de données

Aperçu d'update\_time.php :

```
<?php
require './functions.php';
$bdd = getDatabase();
if (isset($_POST)) {
    $start = $_POST['start'] ?? null;
    $end = $_POST['end'] ?? null;
    $dow = $_POST['dow'] ?? null;
    $allDay = $_POST['allDay'] ?? null;
    $id = $_POST['Id'] ?? null;
    if ($dow==0) {
        $repeatWeek=0;
    } else {
        $repeatWeek=1;
    }
    $query = "UPDATE reservation SET `start`=:debut , `allDay`=:allday , `repeatWeek`=:rep WHERE Id=:id;";
    $requete = $bdd->prepare($query);
    $requete->bindParam(":debut",$start);
    $requete->bindParam(":allday",$allDay);
    $requete->bindParam(":rep",$repeatWeek);
    $requete->bindParam(":id",$id);
    $requete->execute();
    $requete->closeCursor();
}
```

Le module de visualisation possède une barre de tri par salle. Pour permettre le tri, il faut que qu'une réservation se distingue d'une autre. J'ai décidé d'attribuer une couleur différente par salle réservée dans les propriétés des évènements.

```
<?php } else { ?> {
    start: "<?php echo $infos['start'] ?>",
    end: "<?php echo $infos['end'] ?>",
    <?php if (isset($infos['repeatWeek'])&&$infos['repeatWeek']==1) { ?>
    daysOfWeek: [ momentDay('<?php echo $infos['start'];?>') ],
    startTime: formatHour("<?php echo $infos['start'] ?>"),
    startRecur: "<?php echo $infos['start'] ?>",
    stopTime: formatHour('<?php echo $infos['end'] ?>'),
    <?php } ?>
    id: "<?php echo $infos['Id'] ?>",
    <?php if ($infos['libelle'] == "Salle des mariages (les 3 salles)") { ?>
        color: "pink",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salon d'Honneur") { ?>
        color: "gold",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle des Commissions") { ?>
        color: "forestgreen",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle du Conseil Municipal (les 2 salles)") { ?>
        color: "red",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle du Conseil") { ?>
        color: "purple",
    <?php } ?>
    <?php if ($infos['libelle'] == "Salle des commissions") { ?>
```

Voici à quoi ressemble le calendrier :

The screenshot displays a web interface for room reservations. On the left, a sidebar titled "Les salles de la mairie" lists various rooms with checkboxes. The main area shows a calendar for June 2019. The calendar has a header with navigation buttons and a table of dates. The date June 23rd is highlighted in blue.

**Les salles de la mairie**

- ☐ Toutes les salles
- ☐ Salles en cours de validation
- ☐ Salle des mariages (les 3 salles)
- ☐ Salon d'Honneur
- ☐ Salle des Commissions
- ☐ Salle du Conseil Municipal (les 2 salles)
- ☐ Salle du Conseil
- ☐ Salle Louise A. Boyd
- ☐ Salle des Fêtes
- ☐ Salle Rosalind Franklin
- ☐ Salle Bleue
- ☐ Bureau d'Accueil

**Calendar: juin 2019**

Sem.	lun.	mar.	mer.	jeu.	ven.	sam.	din.
22	27	28	29	30	31	1	2
23	3	4	5	6	7	8	9
24	10	11	12	13	14	15	16
25	17	18	19	20	21	22	23
26	24	25	26	27	28	29	30
27	1	2	3	4	5	6	7

## D. Le module de connexion à l'application

La mairie de Sotteville-lès-Rouen possède un Active Directory comme de nombreuses organisations. Un Active Directory est le cœur du réseau d'une entreprise. L'A.D de la mairie comporte tous les identifiants et mots de passe des employés de la mairie afin qu'ils puissent se connecter à leur propre session sécurisée sur le domaine sotteville.local. J'ai donc eu l'idée d'exploiter les données de l'A.D afin de faire mon module de connexion.

Mon module vient comparer la valeur du champ rentré avec le hash du mot de passe entré dans l'A.D. J'ai utilisé le protocole LDAP qui est une extension de PHP, afin de communiquer avec l'A.D :

```
function LoginLDAP($user, $pass)
{
    if (!defined("LDAP_OPT_DIAGNOSTIC_MESSAGE")) {
        define("LDAP_OPT_DIAGNOSTIC_MESSAGE", 0x0032); // pour avoir une connexion plus détaillée
    }
    $login = 'secret';
    $pwd = 'secret';
    $succes = true;
    $ad = getLDAP();
    $connexion = @ldap_bind($ad, $login, $pwd);
    if ($connexion) {
        $filtre = "(sAMAccountName=$user, userPassword=$pass)";
        $resultatLDAP = ldap_search($ad, "chemin jusqu'au dossier des utilisateurs", $filtre);
        $infos = ldap_get_entries($ad, $resultatLDAP);
        if ($infos['count'] == 1) {
            $succes = true;
        }
    }
    if ($succes) {
        @ldap_close($ad);
        unset($ad);
        startSession();
        session_id($user);
        $_SESSION['user'] = $user;
        header("Location: ../index.php");
    } else {
        @ldap_close($ad);
        unset($ad);
        $_SESSION['error_login'] = "erreur de connexion";
        header("Location: ../pages/login.php");
    }
    die();
}
```

Voici à quoi ressemble ma page de connexion :

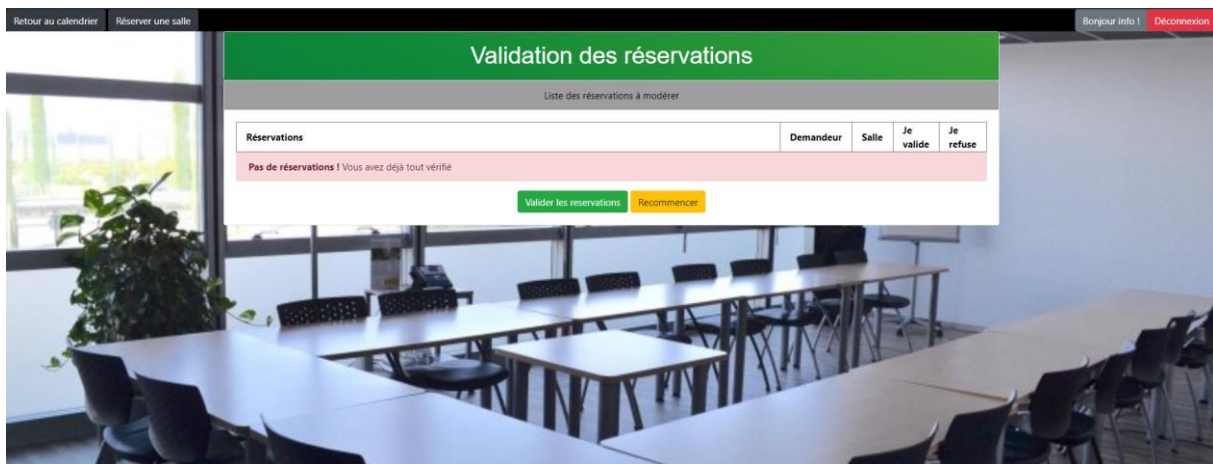


## E. Le module de validation des réservations

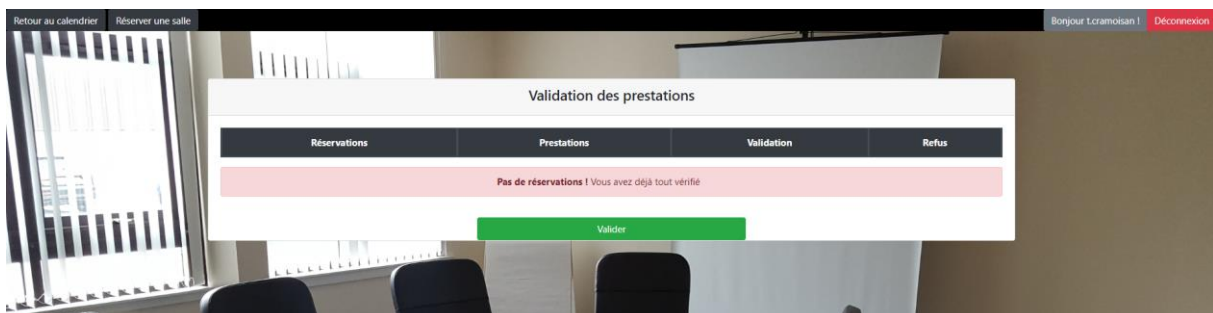
Une réservation doit être validée par les relations publiques pour attester de sa véracité. Dans la démarche de digitaliser le service, j'ai réalisé une page permettant aux supers-utilisateurs de valider les réservations entrés par les utilisateurs.

La manière dont j'ai procédé pour rendre cela fonctionnel est la suivante : J'ai ajouté un champ booléen dans la table réservation permettant de distinguer si une réservation est validée ou non. La valeur de ce champ change si l'administrateur valide la réservation, sinon la réservation est supprimée de la base de données. La page qui modifie cela est sous forme de formulaire comme la page de réservation et de modification.

Voici à quoi ressemble la page de validation uniquement accessible par les administrateurs :



Il faut savoir aussi qu'une réservation peut-être validé mais avec des prestations refusée, dans ce cas présent on ne supprime pas la réservation mais on affiche que celle-ci n'en possède pas. J'ai donc réalisé une page qui traite des prestations comme celle-ci :



### III. Documentation du projet

#### A. Guide d'utilisation de l'application pour les employés de la mairie

Vous pouvez trouver sur la page de mon site concernant mon projet de stage de première année le guide d'utilisation pour employés de mairie : [www.richardinfo.tk](http://www.richardinfo.tk)

#### B. Guide d'utilisation de l'application pour les relations publiques

Vous pouvez trouver sur la page de mon site concernant mon projet de stage de première année le guide d'utilisation pour les relations publiques : [www.richardinfo.tk](http://www.richardinfo.tk)



## IV. Annexes

### A. Documentations utilisées lors du développement

Lors du développement, je me suis principalement aidé d'internet, je vais donc vous donner quelques sites qui m'ont permis de régler des problèmes où qui m'ont tout simplement servi à apprendre de nouvelles notions :

Concernant PHP – LDAP :

- <http://www.manuelphp.com/php/ldap.using.php>
- <https://openclassrooms.com/forum/sujet/active-directory-php>
- <http://jean-luc.massat.perso.luminy.univ-amu.fr/ens/jee/ldap.html>
- <https://www.developpez.net/forums/d1235137/php/php-base-donnees/php-mysql/active-directory-php/>

Concernant fullcalendar.js :

- <https://fullcalendar.io/docs>
- [http://www.w3schools.com/php/php\\_ajax\\_database.asp](http://www.w3schools.com/php/php_ajax_database.asp)
- <https://api.jquery.com/category/ajax/>
- <https://api.jquery.com/>

Et globalement le site qui m'a apporté le plus de solutions : <https://stackoverflow.com/>

## B. Documentations utilisées lors du déploiement

Le déploiement ayant été assisté par mon tuteur, je n'ai pas beaucoup utilisé internet. Néanmoins il y a un site qui m'a beaucoup aidé dans la configuration de mon serveur Web :

- <https://doc.ubuntu-fr.org/lamp>

## C. Organisation personnelle

Le développement d'une application en autonomie nécessite de la discipline personnelle, pour m'organiser j'ai décidé d'utiliser des outils comme Excel pour planifier mes tâches afin de ne pas dépasser la contrainte de temps imposée (1 mois).

Objectifs à atteindre	Statut	Deadline	Motif de retard
Compréhension et Étude du contexte	✓	20/05/2019	
Conception de la BdD	✓	21/05/2019	
Réalisation de la BdD pour l'environnement de test	✓	21/05/2019	
Développement du formulaire de réservation	✓	29/05/2019	Contraintes rediscutées avec le service publique + Complexité de la tâche mal évaluée (2 jours de retard)
Développement du module de visualisation	✓	06/06/2019	Modification sur demande du tuteur (3 jours de retard)
Mise en relation de l'AD et de l'application (LDAP)	✓	11/06/2019	Nouvelle notion (PHP-LDAP) + difficulté à trouver un moyen de comparer les mots de passe de l'AD et ceux rentrés dans l'application (4 jours de retard)
Développement du module de connexion	✓	06/06/2019	
Installation du serveur Apache + MySQL	✓	07/06/2019	
Développement de la page de validation	✓	14/06/2019	
Écriture du guide d'utilisation de l'appli pour les admins	✓	15/06/2019	
Écriture du guide d'utilisation de l'appli	✓	14/06/2019	
Développement de la page de suppression	✓	18/06/2019	
Développement de la page de modification	✓	18/06/2019	
Application fonctionnelle	✓	21/06/2019	

Le motif de retard est important dans la mesure où je désire m'améliorer dans mon organisation personnelle. Noter pourquoi la deadline n'a pas été respectée me permettra à l'avenir d'éviter de recommencer l'erreur. Que cela soit une mauvaise préparation ou anticipation, ou une tâche mal évaluée comme ici présent.

Voici une représentation des tâches effectuées quotidiennes :

Jour	Tâches réalisées	Problèmes rencontrés	Solutions apportées
20/05/2019	Installation du PC Étude du contexte Création de la BdD Début du développement de l'application interne	Pas de clé d'activation pour JMerise sur le lieu de stage	Conception de la BDD avec un schéma
21/05/2019	Développement HTML/CSS/PHP (module de connexion) Mise en place d'une organisation		
22/05/2019	Développement HTML/CSS/PHP (module de connexion) Shématisation du réseau virtuel (environnement de test) Shématisation du réseau physique (environnement de production) Écriture des futures requêtes SQL en français Recherche de documentation pour PHP-LDAP		
23/05/2019	Festi/Robot (Absence)		
24/05/2019	IRM (Absence)		
25/05/2019	WE		
26/05/2019	WE		
27/05/2019	Développement HTML/CSS/PHP (formulaire de réservation)		
28/05/2019	Développement HTML/CSS/PHP (formulaire de réservation) Évolution de l'application Début du développement PHP-LDAP Modification de la base de donnée		
29/05/2019	Finition du module de réservation (HTML/CSS/PHP/JS) Shématisation du module de visualisation Shématisation du module de validation Shématisation du module d'affichage Xibo		
30/05/2019	WE Finition du module de réservation Développement en amont du module de visualisation (planning)		
31/05/2019	Grande avancée dans le développement du calendrier (planning)		
01/06/2019	WE		
02/06/2019	WE Début de la rédaction du rapport de stage Début de la rédaction d'un guide d'utilisation (pas à pas) Finition du module de visualisation (planning)		
03/06/2019	Finition d'écriture du guide d'utilisation Avancement dans le rapport de stage Début de la connexion entre l'application et l'AD Documentation LDAP-PHP		
04/06/2019	Déploiement de l'application sur un serveur test Amélioration esthétique de l'application		
05/06/2019	Tentative de connexion de l'application avec l'active directory de la mairie Amélioration du calendrier avec des alertes personnalisées par réservations		
06/06/2019	Tentative de connexion de l'application avec l'active directory de la mairie		
07/06/2019	Tentative de connexion de l'application avec l'active directory de la mairie Amélioration du calendrier avec un système de tri par salle		
08/06/2019	WE Écriture du rapport de stage Réorganisation des tâches		
09/06/2019	WE Écriture du rapport de stage		
10/06/2019	Férié		
11/06/2019	Développement HTML/CSS/PHP (page de validation) Amélioration de la page de validation avec JavaScript Tentative de connexion de l'application avec l'active directory de la mairie		
12/06/2019	Modification des requêtes SQL (Formulaire) Modification du JavaScript (Visualisation/Validation)		
13/06/2019	Modification du module de visualisation		
14/06/2019	Écriture du guide d'utilisation pour utilisateurs		
15/06/2019	WE		
16/06/2019	WE		
17/06/2019	Amélioration du calendrier avec des fonctionnalités personnalisées + module de validation		
18/06/2019	Développement JS/HTML5/CSS3/PHP7 (calendrier + module de validation) Redéfinition des nouveaux composants applicatifs avec l'accueil Déploiement de l'application sur un des serveurs de la mairie Accompagnement des superutilisateurs à la prise en main de l'application		
19/06/2019	Développement JS/HTML5/CSS3/PHP7 (Modification/suppression) Développement JS/HTML5/CSS3/PHP7 (Double validation de réservation) Développement JS/HTML5/CSS3/PHP7 (Amélioration du calendrier) Rédaction du guide d'utilisation pour superutilisateurs		
20/06/2019	Présentation de l'application aux employés de la mairie		
21/06/2019	Résolution de bugs et nettoyage de la base de données		

Cela me permet de savoir si je travaille à une vitesse régulière et de me situer dans l'avancement du projet tout en m'informant sur le temps restant.