

<b>BTS SERVICES INFORMATIQUES AUX ORGANISATIONS</b>	<b>SESSION 2022</b>
<b>ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)</b>	
<b>Épreuve E5 - Conception et développement d'applications (option SLAM) - Coefficient 4</b>	

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>		<b>N° réalisation : 1</b>		
<b>Nom, prénom : SIEGEL Matthieu</b>		<b>N° candidat : 02144599146</b>		
<b>Épreuve ponctuelle</b> <input checked="" type="checkbox"/> <b>Contrôle en cours de formation</b> <input type="checkbox"/>		<b>Date : 07 / 04 / 2022</b>		
<b>Organisation support de la réalisation professionnelle</b> La MAISON DES LIGUES DE LA LORRAINE (M2L), établissement du Conseil Régional de Lorraine, est responsable de la gestion du service des sports et en particulier des ligues sportives ainsi que d'autres structures hébergées. La M2L doit fournir les infrastructures matérielles, logistiques et des services à l'ensemble des ligues sportives installées.				
M2L souhaiterait mettre en place un service en ligne de gestion de réservation de salles de réunion.				
<b>Intitulé de la réalisation professionnelle</b> Développement d'une plate-forme en ligne de réservation de salles.				
<b>Période de réalisation : 2021/2022</b> <b>Lieu : Groupe GEFOR Paris 1<sup>e</sup></b>				
<b>Modalité :</b> <input type="checkbox"/> <b>Seul(e)</b> <input checked="" type="checkbox"/> <b>En équipe</b>				
<b>Compétences travaillées</b> <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données				
<b>Conditions de réalisation<sup>1</sup> (ressources fournies, résultats attendus)</b>				
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> <b>Ressources fournies :</b> <ul style="list-style-type: none"> <li>Cahier des charges M2L</li> </ul> </td> <td style="width: 50%; vertical-align: top;"> <b>Résultats attendus :</b> <ul style="list-style-type: none"> <li>Application Web de réservation de salle               <ul style="list-style-type: none"> <li>Inscription et gestion de compte utilisateur</li> <li>Réserver une salle</li> <li>Ajout de participants</li> <li>Création de tickets de réclamation</li> </ul> </li> <li>Documentation de mise en service</li> </ul> </td> </tr> </table>			<b>Ressources fournies :</b> <ul style="list-style-type: none"> <li>Cahier des charges M2L</li> </ul>	<b>Résultats attendus :</b> <ul style="list-style-type: none"> <li>Application Web de réservation de salle               <ul style="list-style-type: none"> <li>Inscription et gestion de compte utilisateur</li> <li>Réserver une salle</li> <li>Ajout de participants</li> <li>Création de tickets de réclamation</li> </ul> </li> <li>Documentation de mise en service</li> </ul>
<b>Ressources fournies :</b> <ul style="list-style-type: none"> <li>Cahier des charges M2L</li> </ul>	<b>Résultats attendus :</b> <ul style="list-style-type: none"> <li>Application Web de réservation de salle               <ul style="list-style-type: none"> <li>Inscription et gestion de compte utilisateur</li> <li>Réserver une salle</li> <li>Ajout de participants</li> <li>Création de tickets de réclamation</li> </ul> </li> <li>Documentation de mise en service</li> </ul>			
<b>Description des ressources documentaires, matérielles et logicielles utilisées <sup>2</sup></b>				
<u>Base de données :</u> SGBD MariaDB v15, logiciel DBeaver <u>Développement côté serveur :</u> environnement Node.js v16.13.1, librairie Express.js v4.17.2, gestionnaire de paquet NPM v8.3.0 et logiciel Postman <u>Développement côté client :</u> cadre applicatif React.js v17.0.2 <u>Outil de développement :</u> IDE VSCode <u>Outils collaboratifs :</u> outils de versionnage Git et GitHub, Google Sheets				

<sup>1</sup> En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

<sup>2</sup> Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

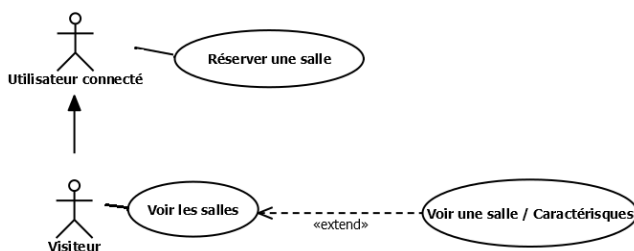
**Épreuve E5 - Conception et développement d'applications (option SLAM) - Coefficient 4**

### Contexte :

Je me suis occupé plus spécifiquement :

- ### Procédure de développement :

## 1. Elaboration du Use Case diagramme



```

graph LR
    users((users)) -- "1" --> "123" reservations((reservations))
    reservations -- "1" --> "123" salles((salles))
    reservations -- "1" --> "123" reservations
  
```

The diagram illustrates the database schema for a reservation system. It consists of three tables: **users**, **reservations**, and **salles**.

- users** table: Attributes include **id** (primary key), **nom**, **prenom**, **email**, **tel**, **PASSWORD**, **ddn**, **adresse**, **is\_active**, and **is\_admin**.
- reservations** table: Attributes include **id** (primary key), **date\_resa**, **id\_user** (foreign key to users), **id\_salle** (foreign key to salles), and **is\_paid**.
- salles** table: Attributes include **id** (primary key), **nom**, **description**, **capacite**, **prix**, and **is\_active**.

The relationships between the tables are as follows:

- users** (1) to **reservations** (123): A one-to-many relationship.
- reservations** (1) to **salles** (123): A one-to-many relationship.
- reservations** (1) to **reservations** (123): A self-referencing relationship.

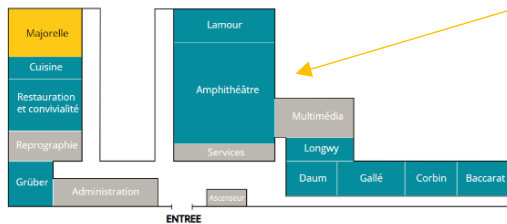
Pour finir, j'ai conçu la partie front-end permettant à l'utilisateur de réserver une salle à une date choisie. L'utilisateur doit être connecté pour pouvoir réserver une salle, sinon il a seulement accès aux caractéristiques de celle-ci. Le développement de cette partie, a été fait via React.js qui est un environnement de travail basé sur Javascript. Il permet de développer des composants réutilisables, de créer des rendus dynamiques et permet une meilleure maintenabilité du code. J'ai utilisé différentes bibliothèques, tel que Axios qui autorise l'accès à notre API via le protocole HTTP. Pour la création de cette page, j'ai modélisé un plan des salles depuis le contexte national qui nous a été fourni. Je l'ai recopié sur Adobe Illustrator pour en faire un fichier SVG et pouvoir interagir avec celui-ci depuis ma page web. J'ai ensuite intégré un calendrier, pour cela j'ai pris la bibliothèque React-Calendar et j'ai dû me documenter dessus pour comprendre son utilisation. Le principal souci rencontré, était de pouvoir rendre inaccessible une date déjà réservée. L'utilisation de cette fonctionnalité du calendrier n'était pas très claire, après plusieurs essais et modifications j'ai réussi à l'implémenter.

### 3. Captures d'interface web liée à la réservation d'une salle

#### Utilisateur non connecté



#### Salles mise à disposition des ligues



#### Plan des salles interactif

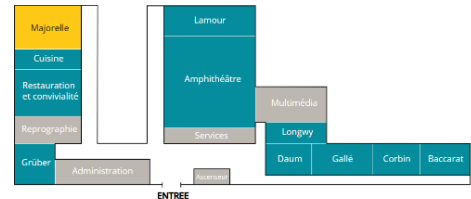
Descriptif de chaque salle, sa capacité d'accueil et le prix de celle-ci.

Nom	Description	Capacité	Prix (€)
Majorelle	Service de sonorisation et vidéo projecteur disponible	30	2000

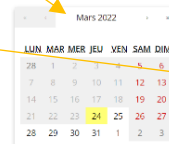
Une fois connecté, l'utilisateur a accès au calendrier.

Possibilité de réserver la salle choisie.

#### Utilisateur connecté



Nom	Description	Capacité	Prix (€)
Majorelle	Service de sonorisation et vidéo projecteur disponible	30	2000



Réserver !

#### Tests de bon fonctionnement :

Les fonctionnalités de visualisation de salle et de réservation ont été testées comme suit :

- Test des procédures stockées SQL via le logiciel DBeaver.
- Test des points d'accès (URL) et contrôleurs (Node.js), via Postman, ainsi que la méthode appelée (GET, POST, etc...).
- Test unitaire, test d'intégration et d'utilisation de l'interface.

### 4. Exemple de test de point d'accès avec Postman

POST localhost:3001/users/1/reservation

URL de connexion à l'API. Le chemin « /users/1/reservation » permet la création d'une réservation, associée à l'utilisateur ayant l'ID N°1, dans notre base de données. L'envoi des données se fait par la méthode POST.

Les données envoyées au format JSON nécessaires à la réservation d'une salle.

```

1 {
2   "date": 20220420,
3   "salle_id": 2,
4   "is_paid": 0
5 }

```

Réserve réussie, notre API retourne un « success » et un code « HTTP » 200 qui correspond succès de la requête.

```

1 {
2   "success": {
3     "affectedRows": 3,
4     "insertId": 0,
5     "warningStatus": 0
6   }
7 }

```