

<b>BTS SERVICES INFORMATIQUES AUX ORGANISATIONS</b>		<b>SESSION 2024</b>
<b>ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)</b>		
<b>Épreuve E5 - Conception et développement d'applications (option SLAM)</b>		

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>		<b>N° réalisation :2</b>										
<b>Nom, prénom :BOUHAJJA Mariem</b>		<b>N° candidat : 01950008457</b>										
<b>Épreuve ponctuelle</b> <input checked="" type="checkbox"/>	<b>Contrôle en cours de formation</b> <input type="checkbox"/>	<b>Date : 14 / 06 /2024</b>										
<b>Organisation support de la réalisation professionnelle</b> La société StreamIO. est une jeune startup de 5 personnes créée en 2020 qui innove dans le domaine de la production audio-visuelle sur Internet. Elle propose des services d'édition vidéo, de diffusion en direct sur Internet, de création de contenu numérique, ainsi que la gestion d'identité numérique pour les vidéastes du web (Youtuber/Streamer), elle propose aussi des salles pour diffusion en direct sous réservation. <b>Comment gérer la réservation des salles ?</b>												
<b>Intitulé de la réalisation professionnelle</b> Mise en place d'une application bureau de gestion de réservation (GR)												
<b>Période de réalisation :</b> Du 18/12/23 au 29/04/24 <b>Lieu :</b> ISCIO – CFA EVE <b>Modalité :</b> <input checked="" type="checkbox"/> <b>Seul(e)</b> <input 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 (ressources fournies, résultats attendus)</b>  <table border="0"> <tr> <td><u>Ressources fournies :</u></td> <td><u>Résultats attendus :</u></td> </tr> <tr> <td>- Cahier des charges StreamIO</td> <td>- Une application bureautique en C#</td> </tr> <tr> <td>- Environnement de développement</td> <td>- Une API REST pour la gestion des données en Node.js</td> </tr> <tr> <td>- une base de données utilisée pour une application web de réservation</td> <td>- Documentation d'utilisation</td> </tr> <tr> <td></td> <td>- Documentation technique</td> </tr> </table>			<u>Ressources fournies :</u>	<u>Résultats attendus :</u>	- Cahier des charges StreamIO	- Une application bureautique en C#	- Environnement de développement	- Une API REST pour la gestion des données en Node.js	- une base de données utilisée pour une application web de réservation	- Documentation d'utilisation		- Documentation technique
<u>Ressources fournies :</u>	<u>Résultats attendus :</u>											
- Cahier des charges StreamIO	- Une application bureautique en C#											
- Environnement de développement	- Une API REST pour la gestion des données en Node.js											
- une base de données utilisée pour une application web de réservation	- Documentation d'utilisation											
	- Documentation technique											
<b>Description des ressources documentaires, matérielles et logicielles utilisées</b>  - Système Windows pour l'exécution du code source de l'application - Système Linux pour l'exécution du code source de l'API - Documentation d'utilisation de l'application - Documentation de déploiement du code source et intégration des données												
<b>Modalités d'accès aux productions et à leur documentation</b> <b>URL :</b> <a href="https://tinyurl.com/iscio-bts-sio-2024">https://tinyurl.com/iscio-bts-sio-2024</a>  <b>Mot de passe :</b> 1SCIO@91												

**ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle  
(verso, éventuellement pages suivantes)****Épreuve E5 - Conception et développement d'applications (option SLAM)****Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs****Contexte :**

La société StreamIO est une jeune startup de 5 personnes créée en 2020 qui innove dans le domaine de la production audio-visuelle sur internet.

Elle propose des services d'édition vidéo, de diffusion directe sur internet, de création de contenu numérique, mais aussi de la gestion d'identité numérique pour les vidéastes du web (Youtubeurs/Streamers...).

Cette société propose aussi des salles à réserver pour filmer/ produire des vidéos accompagnées ou pas par un équipement de tournage selon le choix du client.

M. Miller, directeur de StreamIO, nous a contacté suite à une hausse de demandes concernant la réservation de ses salles.

Après nous avoir expliqué que les réservations se font avec une application web sous forme d'un calendrier qui affiche les dates disponibles pour chaque salle, ce dernier a exprimé que cette solution n'est pas optimale : pour éviter les annulations des réservations à la dernière minute, le développeur du calendrier de réservation a fait le choix de ne pas autoriser aux utilisateurs la suppression de leurs réservations, pour le moment il est impossible d'annuler ni de modifier une réservation. En plus, les informations de chaque personne qui utilise le calendrier sont stockées dans la base de données même ceux qui n'utilisent plus les services de StreamIO, il demande d'avoir la main sur la gestion des clients. Finalement, il exprime son besoin d'avoir la possibilité de supprimer, ajouter et modifier les informations des salles dans la base de données de l'application de réservation.

M. Miller nous a demandé alors une solution sous forme d'application qui permet d'afficher, ajouter, modifier et supprimer les informations concernant les réservations, les salles et les utilisateurs de l'application CRS.

**Le besoin :**

Le besoin principal ici semble être d'avoir la main sur la gestion des réservations, des salles et des Clients.

Après discussion avec M. Miller, nous avons décidé que la solution optimale sera la mise en place d'une application bureautique qui permet la gestion de la base de données des réservations.

**Analyse du besoin :**

**Après entretien avec M. Miller, nous nous sommes mis d'accord sur les fonctionnalités suivantes :**

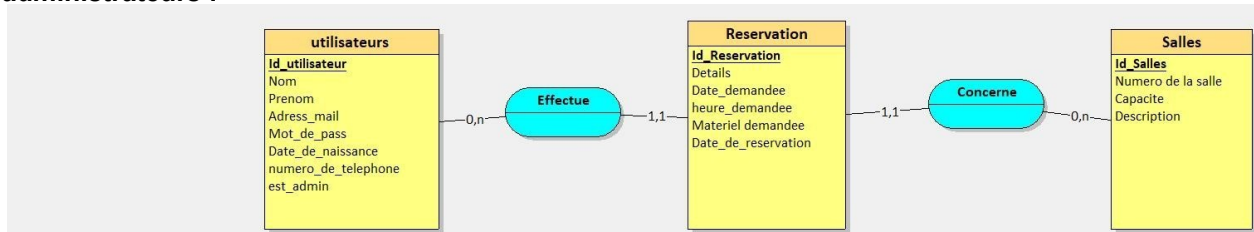
**Fonction principale :**

- Avoir la possibilité d'ajouter, modifier ou supprimer un client.
- Avoir la possibilité d'ajouter, modifier ou supprimer une salle.
- Avoir la possibilité d'ajouter ou supprimer une réservation.

**Fonctions secondaires :**

- Avoir la possibilité d'ajouter, modifier ou supprimer un Admin.
- Afficher le nombre de salles, de clients et d'admins enregistrés dans la base de données

**Après étude de besoin nous avons décidé de garder la même base de données MySQL en y ajoutant plus d'informations dans la table les utilisateurs ainsi que qu'une colonne « est\_admin » pour les administrateurs :**



### Technologies utilisées :

#### Langages de programmation utilisées :

- C# v11
- Node.js v18.17.1
- MySQL

#### Librairies/framework utilisées :

- Express : <https://expressjs.com/>
- Jest : <https://jestjs.io/>
- Nodemon : <https://nodemon.io/>
- WinForms : <https://learn.microsoft.com/fr-fr/dotnet/desktop/winforms/?view=netdesktop-8.0>
- .NET Framework : <https://dotnet.microsoft.com/>
- Newtonsoft.Json (JSON.NET) : <https://www.newtonsoft.com/json>
- System.Net.Http : <https://learn.microsoft.com/fr-fr/dotnet/api/system.net.http?view=net-6.0>

#### Les logiciels utilisés :

- WAMP : wamp server 64 3.3.2 (pour la base de données et phpmyadmin)
- LucidChart : pour la conceptualisation des diagrammes d'utilisation
- Visual Studio 2022

#### Ordinateur :

- Windows 10

### Procédure de la réalisation :

1. Développement du Frontend : mon Frontend a été développé à l'aide de WinForms et est composé de deux parties ; une page de connexion et une deuxième page à laquelle on accède une fois authentifiés et qui comporte un tableau de bord permettant la gestion des réservations.

Ce tableau est composée de «UserControl» qui permettent de créer des contrôles personnalisés et réutilisables dans l'application.

2. Développement de l'API : mon API comporte 17 routes:

- => 5 routes pour effectuer un CRUD sur la table client
- => 5 routes pour effectuer un CRUD sur la table salle
- => 4 routes pour afficher, chercher, ajouter et supprimer une réservation
- => 2 routes pour afficher, chercher et ajouter et un Admin (pour modifier et supprimer un Admin j'utilise les mêmes routes que pour les clients)

3. Mise en relation entre le Frontend et l'API : après la création de l'API avec node.js et le frontend avec C#, j'ai relié les deux applications en utilisant System.Net.Http qui me permet d'envoyer/recevoir des requêtes HTTP et

Newtonsoft.Json qui me permet la sérialisation/désérialisation des données en Json.

**Testes du bon fonctionnement :**

Pour m'assurer du bon fonctionnement de mon application j'ai effectué plusieurs types de tests :

- des tests unitaires : en utilisant Jest, Jest.Mock(pour simuler ma base de données) et supertest (pour pouvoir tester les routes d'une API)
- des tests en utilisant Postman : pour pouvoir voir les réponses des requêtes
- des test manuels : en effectuant une action dans le Front et voir si il y a les changements attendus dans la base de données.