

BTS SIO 2024
Option SLAM
Épreuve E5

EPSI Montpellier

SOLANA-DUPLAA Léo
02344511778

Situation n° 1/2 :
Application Web

Angular
(HTML, CSS et TypeScript)

Table des matières

Présentation de la MAO.....	3
Le client.....	5
Le réseau de l'entreprise.....	6
La réalisation	7
Les fonctionnalités.....	7
La démarche	8
Quelques fonctionnalités expliquées	9
Affichage des produits	9
Explication de la fonctionnalité	9
Capture d'écran	9
Extrait de code.....	9
Ajout d'un produit	11
Explication de la fonctionnalité	11
Capture d'écran	11
Extrait de code.....	11
Connexion sécurisée	13
Explication de la fonctionnalité	13
Capture d'écran	13
Extrait de code.....	13
Triggers et procédure	15
Explication de la fonctionnalité	15
Extrait de code.....	15
Serveur de versions.....	16
Conclusion	16
ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)	17

Présentation de la MAO

La MAO, ou Maison des Associations d'Occitanie, bénéficie du soutien financier du Conseil Régional et est gérée par le Comité Régional Olympique et Sportif du Languedoc Roussillon (CROSLR). Sa principale fonction est de proposer des locaux et des services tant aux associations sportives de la région qu'aux entités qui y sont domiciliées.

La gestion de la Maison des Ligues a été reprise en 2000 par la région, moment auquel les comités départementaux ont été promus par les athlètes, plaidant pour leur inclusion dans le soutien au développement des associations sportives. Depuis, la MAO accueille la plupart des associations sportives régionales et certains comités départementaux, grâce à une convention d'occupation de domaine public et un règlement intérieur qui définissent des critères clés, comme l'exigence pour les entités hébergées de souscrire une assurance responsabilité civile.

Le financement des opérations de la MAO provient en partie des contributions pour les frais de gestion courants, incluant le nettoyage, le chauffage, l'électricité, et l'accès à Internet. Les services fournis par la MAO sont facturés via un accord de cogestion avec le CROSLR et la Région Occitanie. La MAO offre aussi la possibilité aux ligues et comités départementaux de réserver des salles de réunion par un système de réservation en ligne. En 2009, elle a organisé l'utilisation de 2732 salles de réunion, accueillant plus de 47316 participants, avec une facturation s'appliquant seulement après plus de six réservations annuelles.

Pour assurer une tarification équitable, la MAO a établi quatre niveaux de prix. Les clubs sportifs et comités départementaux bénéficient d'un tarif préférentiel, tandis que les associations, écoles secondaires et collèges se voient appliquer un second niveau de tarification. Les tarifs les plus élevés concernent d'autres entités et les entreprises privées. L'équipe de la MAO se compose de divers professionnels, incluant deux membres du personnel du Conseil Régional et sept employés à temps plein du CROSLR.

En plus, la MAO emploie du personnel pour des services spécifiques (sécurité, nettoyage) et compte parmi ses rangs les employés des associations et des bénévoles, ces derniers étant élus parmi les membres de l'association. Les élus du Conseil Régional, les visiteurs et les stagiaires complètent l'équipe de la MAO. Avec moins de 10 employés à temps plein, la MAO bénéficie du Plan Impact Emploi Association, lui donnant accès à un logiciel spécialisé pour la gestion des salaires et des déclarations sociales et fiscales, service offert gratuitement par des tiers de confiance.

Ce système facilite la gestion des récapitulatifs périodiques, des déclarations à l'URSSAF et à l'ASSEDIC, ainsi qu'aux caisses de retraite. La MAO détient également la

certification «CRIB», lui permettant de fournir des conseils aux associations employant du personnel. Son but ultime est de superviser et de contrôler, via une plateforme de gestion, les systèmes informatiques de la MAO et ceux des entités hébergées.

Je travaille actuellement pour la MAO, au sein du département informatique, qui fait régulièrement appel à des entreprises locales spécialisées en informatique, établissant ainsi une relation contractuelle entre la MAO et ces entreprises.

Le client

Frip&Collect ont contacté la MAO pour leur demander la mise en place d'une application WEB de E-commerce et de gestion des stocks. Frip&Collect est une friperie créée en 2017 située à Montpellier.

Face aux changements de comportement des consommateurs accentués depuis 2020, Frip&Collect souhaite s'adapter aux nouvelles habitudes d'achat. La pandémie ayant renforcé la tendance au shopping en ligne, l'entreprise vise à atteindre une clientèle désireuse de faire ses achats de manière plus pratique, sans avoir à se déplacer, ou pour ceux qui, par manque de temps ou de moyens, préfèrent explorer et acheter des articles depuis leur canapé.

L'objectif est de développer une application web qui, en plus de faciliter l'accès à un large éventail de produits vintage et de seconde main, intégrera une fonctionnalité de gestion des stocks. Cette innovation permettra à Frip&Collect non seulement de proposer une expérience utilisateur fluide et engageante, mais aussi d'optimiser la gestion de son inventaire en temps réel. Les utilisateurs pourront ainsi bénéficier d'une visibilité accrue sur la disponibilité des articles, tandis que Frip&Collect pourra ajuster son offre en fonction de la demande et des tendances, assurant une gestion efficace de ses ressources.

Le réseau de l'entreprise

Présentation schématique du réseau de l'entreprise :

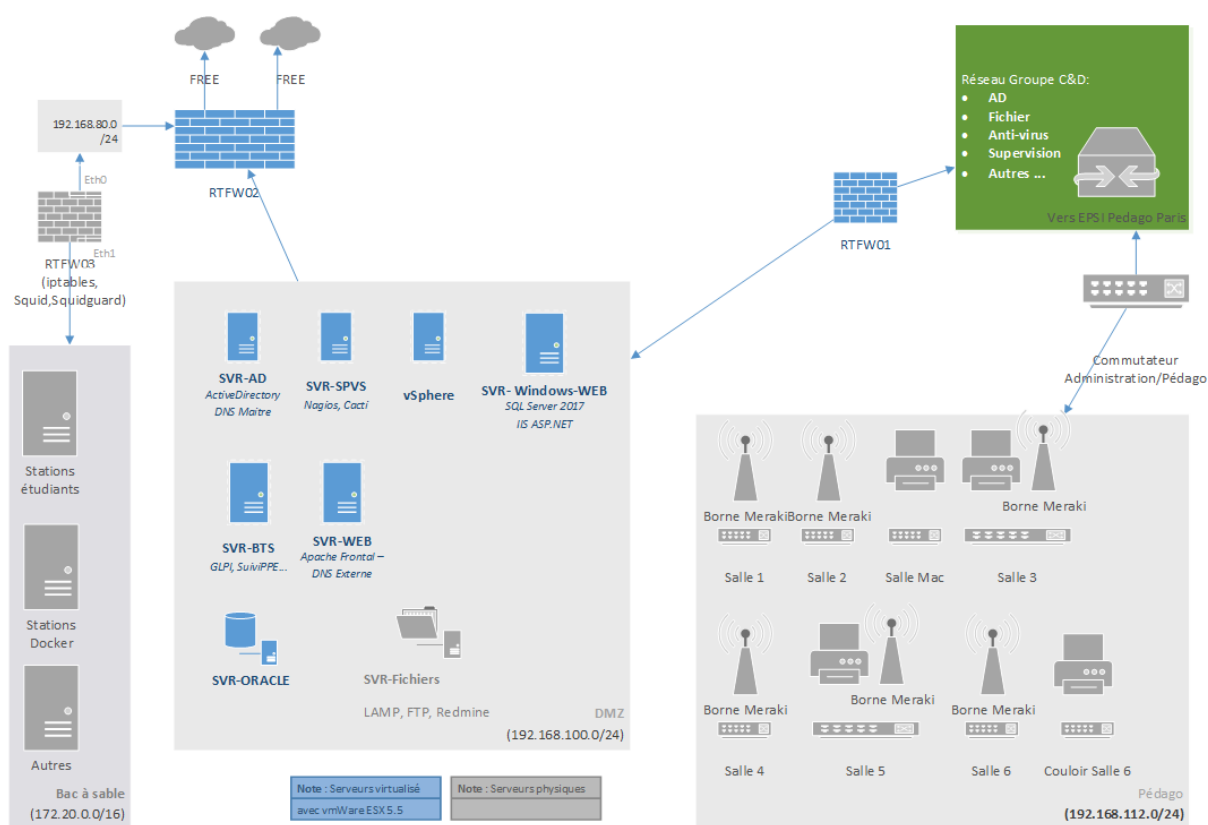
La MAO dispose actuellement :

- D'un réseau commun aux associations et à l'administration,
- D'un réseau de type DMZ dans lequel on trouve un ensemble de serveurs accessibles depuis l'extérieur,
- D'une connexion à internet.

Note : une segmentation des réseaux est en projet pour augmenter la sécurité.

Voici le lien vers la base de données :

`mongodb+srv://admin:zgRCG5e02wk9jRnD@cluster0.vrkeyk.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0`



La réalisation

Les fonctionnalités

Après avoir discuté avec le client il m'a énuméré les différentes fonctionnalités qu'il aimera retrouver dans cette application Web, j'ai donc ainsi pu faire un cahier de charges :

L'application Web stocks doit permettre :

- De suivre les mouvements de stocks en temps réel
- De contrôler les niveaux de stocks
- De gérer les commandes clients
- De gérer les livraisons fournisseurs

Fonctionnalités du site :

- Un tableau de bord pour suivre les stocks en temps réel.
- Une interface pour ajouter des articles, modifier des articles et supprimer des articles du stock.
- Un système d'alerte pour signaler les niveaux de stocks critiques et les retards de livraison.
- Une interface pour commander des articles.
- Une formulaire d'inscription et de connexion avec une page de modification du profil.

Exigences techniques :

- Le site doit être développé avec les dernières technologies web et être compatible avec les navigateurs les plus courants
- Le site doit être hébergé sur un serveur sécurisé et fiable avec une sauvegarde régulière des données
- Le site doit être rapide et réactif pour permettre une utilisation fluide

Design :

- Le site doit avoir un design moderne et attractif, en adéquation avec l'image de l'entreprise
- Le site doit être ergonomique pour permettre une utilisation intuitive

La démarche

Le développement de l'application s'est passé en trois étapes :

- **La création de l'interface de l'application Web**

Cette partie correspond à la création des différentes pages HTML et CSS (Angular).

- **Mise en place du code TypeScript et des requêtes**

Cette partie correspond à la mise en place du code et des différentes fonctions qui permet la récupération, l'ajout ou la suppression des données.

- **Mise en place de l'api et de la base de données**

Cette partie correspond à la mise en place du code NodeJS qui permet la récupération, l'ajout, la modification ou la suppression des données directement dans la base de données.

Quelques fonctionnalités expliquées

Affichage des produits

Explication de la fonctionnalité

Dans cette partie de l'application pour afficher tous les produits présents dans la base de données il faut :

- Récupérer tous les produits présents dans la table « Produit » en utilisant la requête SQL « SELECT * FROM Produit »
- Puis j'affiche le contenu de la liste.

Capture d'écran

Liste des produits

Rechercher un fournisseur

--Genres-- --Tailles-- --États-- --Fournisseurs-- --Types-- Prix max

Costume 2 pièces

Id : 1 Type : costume

Genre : homme Taille : xxs

État : bon-etat Prix : 19.99€

Fournisseur : En stock : 5

Nike

En savoir plus

T-shirt rouge

Id : 2 Type : t-shirt

Genre : homme Taille : xs

État : bon-etat Prix : 19.99€

Fournisseur : En stock : 5

Nike

En savoir plus

Jean blue

Id : 3 Type : jean

Genre : homme Taille : s

État : tres-bon-etat Prix : 19.99€

Fournisseur : En stock : 5

Puma

En savoir plus

Pantalon beige

Id : 4 Type : pantalon

Genre : homme Taille : m

État : tres-bon-etat Prix : 19.99€

Fournisseur : En stock : 5

Puma

En savoir plus

Costume 3 pièces

Id : 5 Type : costume

Genre : femme Taille : xxs

État : bon-etat Prix : 19.99€

Fournisseur : En stock : 5

Puma

En savoir plus

T-shirt vert

Id : 6 Type : t-shirt

Genre : femme Taille : xs

État : bon-etat Prix : 19.99€

Fournisseur : En stock : 5

Puma

En savoir plus

Extrait de code

TypeScript :

```
produits: Produit[] = [];  
ngOnInit(): void {  
    this.produits = this.produitService.getProduits();  
}
```

HTML :

```
<div class="row">
  <div *ngFor="let produit of produits" class="col-3">
    <h2>{{ produit.nom }}</h2>
    <div class="infos row">
      <p class="col-5">Id : {{ produit.id }}</p>
      <p class="col-5">Type : {{ produit.type }}</p>
    </div>
  </div>
</div>
```

Ajout d'un produit

Explication de la fonctionnalité

Pour ajouter un nouveau produit il faut :

- Vérifier que l'utilisateur soit bien connecté en admin et qu'il a bien cliqué sur le bouton « Ajouter un produit ».
- Puis j'ajoute les informations entrées par l'utilisateur avec une requête INSERT INTO dans la table « Produit » de la base de données.

Capture d'écran

Créer un produit

id :	<input type="text" value="19"/>
Nom :	<input type="text" value="T-shirt de sport"/>
Type :	<input type="text" value="T-shirt"/>
Genre :	<input type="text" value="Homme"/>
Taille :	<input type="text" value="M"/>
Prix :	<input type="text" value="19.99"/>
État :	<input type="text" value="Neuf avec étiquette"/>
Fournisseur :	<input type="text" value="Celio"/>
En stock :	<input type="text" value="3"/>

Extrait de code

TypeScript :

```
this.produitForm = this.fb.group({  
  id:[],  
  nom: [''],  
  type: [''],
```

```
genre: [],  
taille: [],  
prix: [],  
etat: [],  
fournisseur: [],  
nombre: [],  
});
```

HTML :

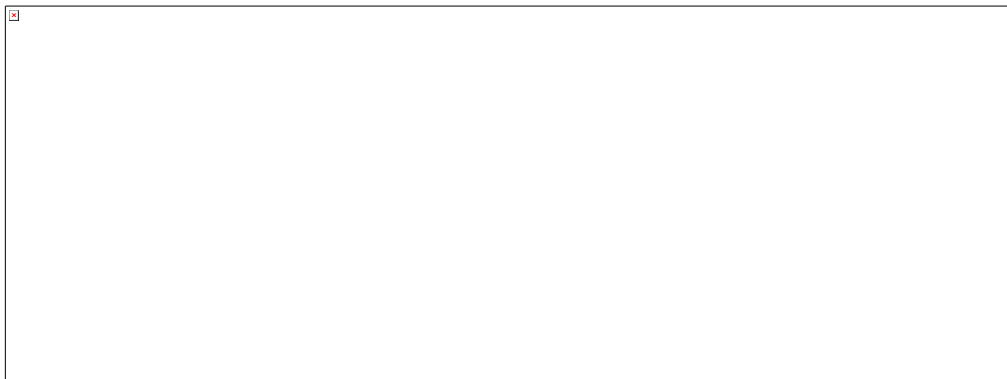
```
<div class="row">  
  <label for="id" class="col-4">id :</label>  
  <input type="text" id="id" formControlName="id" class="col-6">  
</div>  
<div class="row">  
  <label for="nom" class="col-4">Nom :</label>  
  <input type="text" id="nom" formControlName="nom" class="col-6">  
</div>
```

Connexion sécurisée

Explication de la fonctionnalité

Pour sécuriser la connexion l'utilisateur doit entrer son identifiant et son mot de passe si celui-ci existe dans la base de données alors il pourra accéder à l'application. J'ai utilisé une page de connexion pour tous les utilisateurs après avoir vérifié que l'utilisateur à entrer c'est donné ci c'est le cas je vérifie qu'il existe dans la base de données. Puis il est redirigé dans la page qui correspond. S'il est admin, il aura accès à plus de pages et de droits. Par exemple, si un client tente en modifiant l'URL de se diriger vers la page de création de produit, il ne va pas pouvoir y accéder.

Capture d'écran



Inscription



Extrait de code

TypeScript :

```
login(): void {  
  const success = this.authService.login(this.loginUsername, this.loginPassword);  
  if (success) {  
    console.log('Connexion réussie !');  
    this.router.navigate(['/']);  
  } else {  
    console.log('Identifiants incorrects.');
```

```

}

register(): void {
  const success = this.authService.register(this.username, this.password);
  if (success) {
    console.log('Inscription réussie !');
    this.connexionForm();
  } else {
    console.log('L\'utilisateur existe déjà. ');
  }
}

```

HTML :

```

<div class="row">
  <input type="text" placeholder="Nom d'utilisateur" [(ngModel)]="loginUsername"
class="col-3"/>
</div>
<div class="row">
  <input type="password" placeholder="Mot de passe" [(ngModel)]="loginPassword"
class="col-3"/>
</div>
<div class="row">
  <button routerLinkActive="active" class="col-2" (click)="InscriptionForm()">Créer
un compte</button>
  <button (click)="login()" class="col-2 connecter">Se connecter</button>
</div>

```

Triggers et procédure

Explication de la fonctionnalité

Il s'agit d'un trigger qui nous envoie un message lorsqu'un produit passe sous le nombre de 5 en stock.

Pour ce faire :

Extrait de code

```
CREATE TRIGGER check_stock
ON Produit
AFTER UPDATE
AS
BEGIN
    DECLARE @stock_limit INT;
    SET @stock_limit = 5;

    IF EXISTS (SELECT 1 FROM inserted WHERE nombre_en_stock < @stock_limit)
    BEGIN
        DECLARE @nom NVARCHAR(100);
        SELECT @nom = nom FROM inserted WHERE nombre_en_stock < @stock_limit;
        DECLARE @message NVARCHAR(255);
        SET @message = 'Attention! Le stock du produit ' + @nom + ' est inférieur à 5.';
        RAISERROR (@message, 16, 1);
    END
END;
```

Serveur de versions

Pour sauvegarder mon travail j'ai utilisé un serveur de version. Le serveur de version que j'ai utilisé est GitHub le projet se nomme Frip&Collect.

Voici le lien pour accéder au projet :

<https://github.com/leosdlp/Frip-collect>

Conclusion

Le projet d'une application web demandé par la friperie Frip&Collect a été mené à bien, les différentes fonctionnalités demandées ont été ajoutées à l'application. Frip&Collect peut désormais vendre les produits accessibles à tout le monde, même s'ils sont dans leur canapé.

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

ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation :1
Nom, prénom : SOLANA-DUPLAA Léo		N° candidat : 02344511778
Épreuve ponctuelle X	Contrôle en cours de formation	Date : 12 / 04 / 2024
Organisation support de la réalisation professionnelle EPSI Montpellier		
Intitulé de la réalisation professionnelle Application Web Frip&Collect		
Période de réalisation : Octobre 2023 à Mars 2024 Lieu : Montpellier		
Modalité : X Seul(e) En équipe		
Compétences travaillées Concevoir et développer une solution applicative Assurer la maintenance corrective ou évolutive d'une solution applicative Gérer les données		
Conditions de réalisation ⁵ (ressources fournies, résultats attendus) Cahier des charge / liste de fonctionnalité / MCD Site Web fonctionnel		
Description des ressources documentaires, matérielles et logicielles utilisées ⁶ VScode, MongoDB, ordinateur personnel		
Modalités d'accès aux productions ⁷ et à leur documentation ⁸ OneDrive: https://drive.google.com/drive/u/1/folders/1lBIUhkEyMmQEAwv2iJ6QCetJwkZbXnZ4 GIT : https://github.com/leosdlp/Frip-collect Base de données : mongodb+srv://admin:zgRCG5e02wk9jRnD@cluster0.vrkeyk.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0 User : admin, Pwd : zgRCG5e02wk9jRnD		

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

**ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Voir la documentation

2.2 Les activités de l'organisation cliente s'appuient sur aux moins deux solutions applicatives opérationnelles permettant d'offrir un accès sécurisé à des données hébergées sur un site distant. Au sein des architectures de ces solutions applicatives doivent figurer l'exploitation de mécanismes d'appel à des services applicatifs distants et au moins trois des situations ci-dessous :

Éléments	Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)	Remarques de la commission D'interrogation
Du code exécuté sur le système d'exploitation d'une solution technique d'accès fixe (type client lourd)		
Du code exécuté dans un navigateur Web (type client léger ou riche)	Visual studio code - Framework Angular(HTML, CSS, TypeScript)	
Du code exécuté sur le système d'exploitation d'une solution technique d'accès mobile	AndroidStudio – Flutter (Dart)	
Du code exécuté sur le système d'exploitation d'un serveur	PhpMyAdmin (Procédures stockées, Trigger)	

2.3 Une solution applicative peut être issue d'un développement spécifique ou de la modification du code d'un logiciel notamment open source.

2.4 Les solutions applicatives présentes dans le contexte sont opérationnelles et leur code source est accessible dans un environnement de développement opérationnel au moment de l'épreuve.