

BTS SIO 2024  
Option SLAM  
Épreuve E5

**EPSI Montpellier**

**CONCAS Travis**

**N°Candidat : 02344511429**

Situation n° 1/2 :  
Application Mobile

Flutter

## Table des matières

Table des matières .....	2
Présentation de la MAO .....	2
Le client.....	4
Le réseau de l'entreprise .....	5
La réalisation.....	6
Les fonctionnalités .....	7
La démarche .....	7
La base de données .....	8
Quelques fonctionnalités expliquées.....	9
Explication de la fonctionnalité .....	9
Captures d'écran .....	10
Extrait de code .....	10
Affichage des produits .....	12
Explication de la fonctionnalité .....	12
Captures d'écran .....	13
Extrait de code .....	13
Ajout d'un produit.....	15
Explication de la fonctionnalité .....	15
Captures d'écran .....	15
Extrait de code .....	16
Triggers et procédure .....	18
Explication de la fonctionnalité .....	18
Extrait de code .....	18
<a href="https://github.com/zedtroom/CumparKlimbFlutterBTS">https://github.com/zedtroom/CumparKlimbFlutterBTS</a> .....	19
Conclusion .....	19
ANNEXE IV -1 : fiche de présentation d'une situation professionnelle (modèle) .....	19

# Présentation de la MAO

La Maison des Associations d'Occitanie (MAO) est un établissement soutenu par le Conseil Régional dont la gestion est confiée au Comité Régional Olympique et Sportif du Languedoc Roussillon (CROSLR). Son objectif est de fournir des locaux et des services aux associations sportives de la région ainsi qu'aux entités hébergées.

Lorsque la région a repris la Maison des Ligues en 2000, les athlètes ont plaidé en faveur de l'intégration des comités départementaux dans le développement des associations sportives. Depuis lors, la MAO abrite la plupart des associations sportives régionales ainsi que certains comités départementaux. Ce logement est établi par le biais d'une convention d'occupation du domaine public et d'un règlement intérieur qui incluent diverses clauses importantes telles que l'obligation d'assurance responsabilité civile pour les structures hébergées.

Le financement des services de la MAO couvre les frais de gestion courante tels que le nettoyage des locaux, le chauffage, l'électricité et l'accès à internet. La facturation des prestations de la MAO est régulée par une convention de cogestion entre le CROSLR et la Région Occitanie. De plus, la MAO offre aux ligues et aux comités départementaux la possibilité de réserver des salles de réunion via un intranet. En 2009, la MAO a permis l'utilisation de 2732 salles de réunion, accueillant ainsi plus de 47316 participants. Cependant, ce service est facturé uniquement après avoir effectué plus de six réservations par an (à l'exception des amphithéâtres).

Pour assurer une équité entre les entités, la MAO a établi quatre niveaux de tarification. Les clubs sportifs et les comités départementaux bénéficient d'un premier niveau de tarification, tandis que les associations, les lycées et les collèges bénéficient d'un deuxième niveau. Un taux de tarification plus élevé est appliqué aux autres organismes et aux entreprises privées. La MAO emploie un personnel diversifié pour répondre à tous les besoins de la région, comprenant deux employés du Conseil Régional et sept équivalents temps plein (ETP) employés par le CROSLR.

En plus du personnel permanent, la MAO engage également des prestataires (pour la sécurité et le nettoyage) ainsi que des salariés d'associations. Les bénévoles élus par les associations font également partie intégrante du personnel, tout comme les élus du Conseil Régional, les visiteurs et les stagiaires. La MAO, avec moins de 10 ETP, bénéficie du Plan Impact Emploi Association, lui permettant d'accéder au logiciel Impact Emploi pour la gestion des salaires et des déclarations fiscales et sociales. Ce logiciel est géré par des tiers de confiance, facturant à l'association une cotisation annuelle de 60€ par association ainsi que 5€ par fiche de paie.

La MAO détient également un label "CRIB" lui permettant de conseiller les associations employant du personnel. Son objectif principal est de superviser et de contrôler, via une plateforme de gestion des configurations, les postes de travail de la MAO ainsi que ceux des entités hébergées.

En tant que salarié de la Maison des Associations d'Occitanie (MAO) travaillant dans le domaine informatique, je suis chargé de faire appel aux sociétés informatiques de la région. Un contrat est ainsi établi entre la MAO et ces entreprises.

---

## Le client

Cumparklimb (Entreprise de revente de matériel escalade) ont contacté la MAO pour leur demander la mise en place d'une application Mobile de gestion de leurs produits. Cumparklimb est une entreprise basée dans le secteur sportif créée en 2019 située à Montpellier

Cumparklimb a pour objectifs de gérer ses produits pour une durée indéfinie afin de mieux gérer leurs achats et ventes en ligne

Par suite de la situation sanitaire du Covid, l'entreprise a eu une grosse baisse de son chiffre d'affaires et n'a pas pu se faire connaître au public, ils veulent ainsi revendre en ligne leurs stocks en les gérant.

Ils cherchent actuellement une solution permettant aux acheteurs d'accéder plus facilement à tous les produits ils ont donc opté pour une application mobile comme ça ils auront accès aux produits et pourront les acheter.

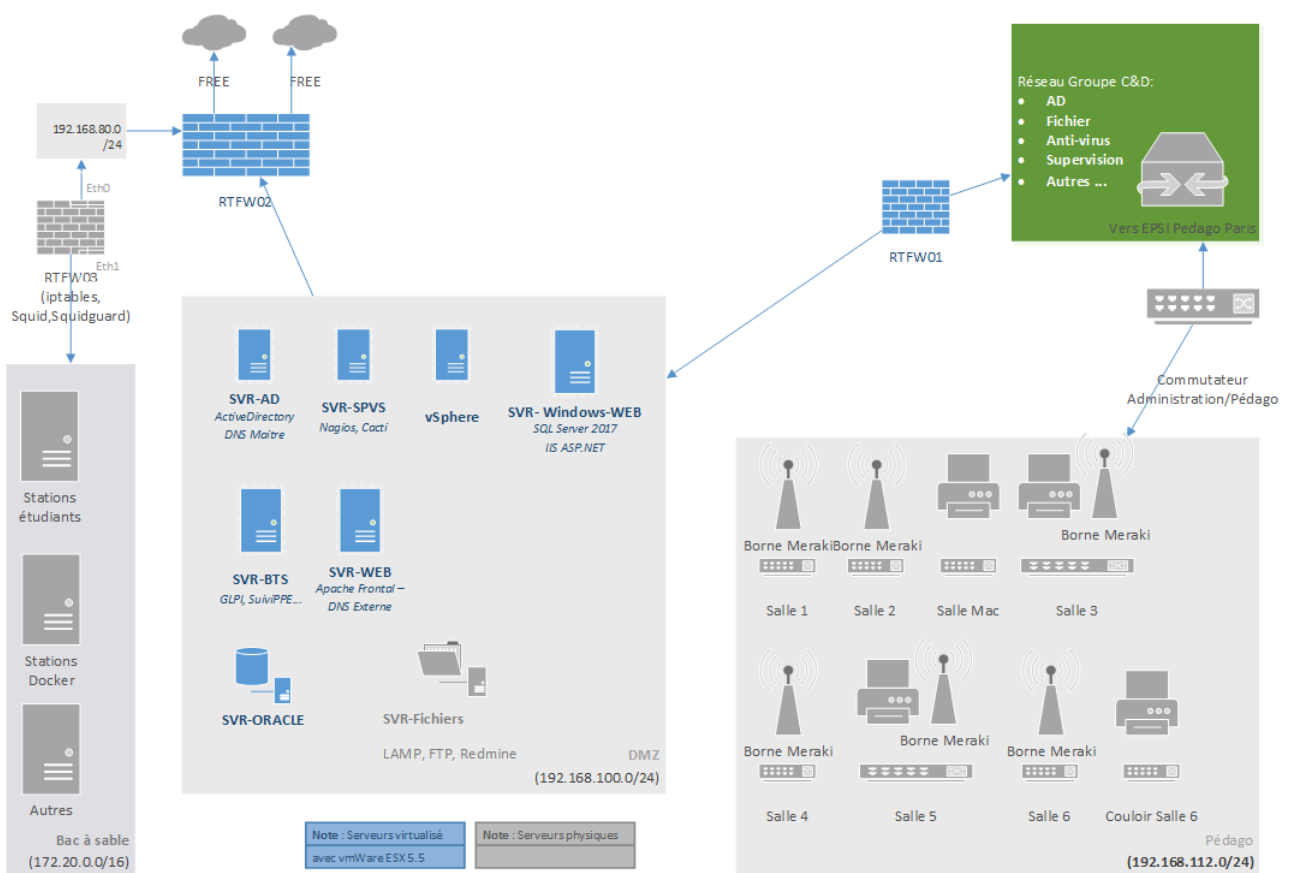
Il faut donc mettre en place une application mobile permettant aux vendeurs d'ajouter des produits et aux clients de récupérer les produits.

# 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.



# La réalisation

## Les fonctionnalités

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

### 1. La gestion des connexions

On pourra dans cette fonctionnalité :

- Avoir une connexion sécurisée au site : pouvoir se connecter grâce à un mot de passe et un courriel
- Modification d'un mot de passe : on peut modifier le mot de passe
- Suppression d'un utilisateur :
- Lister les utilisateurs

### 2. La gestion des produits

Ce qui permettra de :

- Pour voir le nombre de stocks
- Modification d'un produit : on peut modifier l'image, nom ou autre
- Suppression ou ajout d'un produit :
- Lister les produits par utilité

### 3. La gestion des administrateurs

Comme pour les étudiants on pourra :

- Ajouter un administrateur : vérification des données entrées tels que l'identifiant et mot de passe
- Modification d'un administrateur : on peut modifier l'identifiant ou le mot de passe
- Suppression d'un administrateur :
- Lister les administrateurs

# La démarche

Le développement de l'application c'est passé en deux étapes

## 1) Mise en place de IHM

Cette partie correspond à la création des différentes pages Dart.

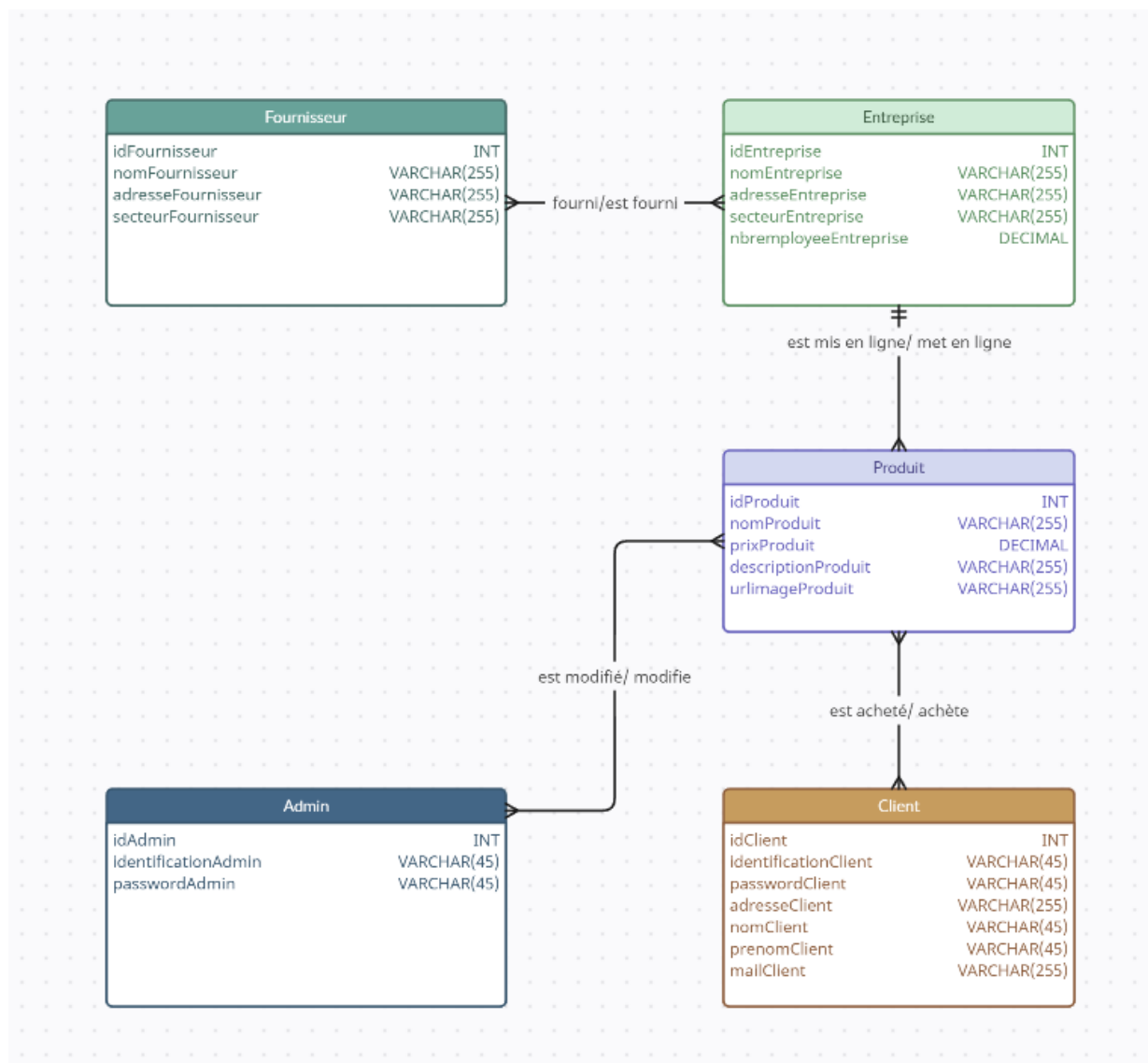
## 2) Mise en place du code Flutter et des requêtes

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



# La base de données

La base de données est constituée de 5 tables correspondant aux fonctionnalités énumérées au début voici une représentation assez claire du MCD :



## Quelques fonctionnalités expliquées

### 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 ses

données ci c'est le cas je vérifie qu'il existe dans la base de données. Puis il est redirigé dans la page qui correspond.

## Captures d'écran



## Extrait de code

```
import 'package:flutter/material.dart';
```

```
void main() {
```

```
runApp(MyApp());  
}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: Scaffold(  
        body: Container(  
          decoration: BoxDecoration(  
            image: DecorationImage(  
              image: AssetImage('assets/images/background.jpg'),  
              fit: BoxFit.cover,  
            ),  
          ),  
        ),  
        child: Center(  
          child: LoginPage(),  
        ),  
      ),  
    );  
  }  
}
```

```
class LoginPage extends StatefulWidget {  
  @override  
  _LoginPageState createState() => _LoginPageState();  
}
```

```
class _LoginPageState extends State<LoginPage> {  
  final _emailController = TextEditingController();  
  final _passwordController = TextEditingController();
```

```
  void _login() {  
    final email = _emailController.text;  
    final password = _passwordController.text;
```

```
    // Remplacez ceci par votre propre logique d'authentification  
    if (email == 'test@test.com' && password == 'password') {  
      print('Connexion réussie');  
    } else {  
      print('Échec de la connexion');  
    }  
  }  
}
```

```
  @override  
  Widget build(BuildContext context) {  
    return Column(  
      mainAxisAlignment: MainAxisAlignment.center,  
      children: <Widget>[  
        Text(  
          'Email: ',
```

```

        'CUMPARKLIMB',
        style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
    ),
    SizedBox(height: 20),
    Text(
        'Connectez vous',
        style: TextStyle(fontSize: 20),
    ),
    SizedBox(height: 20),
    Padding(
        padding: const EdgeInsets.all(8.0),
        child: TextField(
            controller: _emailController,
            decoration: InputDecoration(
                filled: true,
                fillColor: Colors.white,
                hintText: 'Email (obligatoire)',
            ),
        ),
    ),
    Padding(
        padding: const EdgeInsets.all(8.0),
        child: TextField(
            controller: _passwordController,
            obscureText: true,
            decoration: InputDecoration(
                filled: true,
                fillColor: Colors.white,
                hintText: 'Mot de passe (obligatoire)',
            ),
        ),
    ),
    RaisedButton(
        onPressed: _login,
        child: Text('Envoyer'),
    ),
],
);
}
}

```

## Affichage des produits

### Explication de la fonctionnalité

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

- Récupérer tous les cours présents dans la table « Cours » en utilisant la requête SQL « SELECT \* FROM... »

- BDquery est une fonction créée pour récupérer les données de la table et l'ajouter dans une liste.
- Puis j'affiche le contenu de la liste dans l'application

## Captures d'écran

Recherche



Scarpa Drago LV  
**169,90 €**



## Extrait de code

```
import 'package:flutter/material.dart';  
import 'package:mysql1/mysql1.dart';  
  
void main() {  
  runApp(MaterialApp(  
    home: Scaffold(  
      appBar: AppBar(  
        title: Text('Scarpa'),  
      ),  
    ),  
  ),  
);
```

```

        title: Text('Scarpa'),
        actions: <Widget>[
            IconButton(
                icon: Icon(Icons.search),
                onPressed: () {
                    print('Recherche');
                },
            ),
        ],
    ),
    body: FutureBuilder<List<Product>>(
        future: fetchProducts(),
        builder: (context, snapshot) {
            if (snapshot.hasData) {
                return ListView.builder(
                    itemCount: snapshot.data.length,
                    itemBuilder: (context, index) {
                        return ListTile(
                            leading: Image.network(snapshot.data[index].imageUrl),
                            title: Text(snapshot.data[index].name),
                            subtitle: Text(snapshot.data[index].price.toString()),
                        );
                    },
                );
            } else if (snapshot.hasError) {
                return Text("${snapshot.error}");
            }
            return CircularProgressIndicator();
        },
    ),
));
}

```

```

class Product {
    final String name;
    final double price;
    final String imageUrl;

```

```

    Product({this.name, this.price, this.imageUrl});
}

```

```

Future<List<Product>> fetchProducts() async {
    final conn = await MySqlConnection.connect(ConnectionSettings(
        host: 'localhost',
        port: 3306,
        user: 'votre_utilisateur',
        password: 'votre_mot_de_passe',
        db: 'votre_base_de_donnees',
    ));
}

```

```

var results = await conn.query('select name, price, imageUrl from products');
var products = <Product>[];
for (var row in results) {
    products.add(Product(
        name: row[0].toString(),
        price: double.parse(row[1].toString()),
        imageUrl: row[2].toString(),
    ));
}
return products;
}

```

## Ajout d'un produit

### Explication de la fonctionnalité

Pour ajouter un nouveau cours il faut :

- Vérifier que le produit, n'existe pas déjà pour ne pas avoir de doublon sur l'application, ce qui est fait manuellement par le créateur
- Puis quand je suis sûr j'ajoute les informations entrées par l'administrateur, avec l'image, le nom, le prix et la description.
- Enfin avec une requête INSERT INTO on insère les valeurs du mon tableau dans un dans la base de données

## Captures d'écran

produit

**URL** (lien de l'image affiché)

**Nom**

**Prix**

**Description**

**Soumettre**

**Annuler**

## Extrait de code

```
import 'package:flutter/material.dart';  
import 'package:mysql1/mysql1.dart';
```



```

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text('Ajouter un produit'),
      ),
      body: ProductForm(),
    ),
  ));
}

```

```

class ProductForm extends StatefulWidget {
  @override
  _ProductFormState createState() => _ProductFormState();
}

```

```

class _ProductFormState extends State<ProductForm> {
  final _formKey = GlobalKey<FormState>();
  final _product = Product();

```

```

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        children: <Widget>[
          TextFormField(
            decoration: InputDecoration(labelText: 'URL (lien de l\'image
affiché)'),
            onSave: (value) {
              _product.imageUrl = value;
            },
          ),
          TextFormField(
            decoration: InputDecoration(labelText: 'Nom'),
            onSave: (value) {
              _product.name = value;
            },
          ),
          TextFormField(
            decoration: InputDecoration(labelText: 'Prix'),
            onSave: (value) {
              _product.price = double.parse(value);
            },
          ),
          RaisedButton(
            onPressed: () {
              if (_formKey.currentState.validate()) {
                _formKey.currentState.save();
                addProduct(_product);
              }
            }
          )
        ],
      ),
    );
  }
}

```

```

        },
        child: Text('Soumettre'),
      ),
    ],
  ),
);
}
}

```

```

class Product {
  String name;
  double price;
  String imageUrl;

  Product({this.name, this.price, this.imageUrl});
}

```

```

Future<void> addProduct(Product product) async {
  final conn = await MySqlConnection.connect(ConnectionSettings(
    host: 'localhost',
    port: 3306,
    user: 'votre utilisateur',
    password: 'votre mot de passe',
    db: 'votre base de donnees',
  ));

  await conn.query(
    'insert into products (name, price, imageUrl) values (?, ?, ?)',
    [product.name, product.price, product.imageUrl],
  );
}

```

## Triggers et procédure

### Explication de la fonctionnalité

Il s'agit d'un trigger qui permet de tenir l'administrateur au courant des modifications du nom des produits :

### Extrait de code

```

CREATE TRIGGER ModificationNomProduit
AFTER UPDATE ON Produit
FOR EACH ROW
BEGIN

```

```
IF OLD.nomProduit != NEW.nomProduit THEN  
    CALL EnvoyerNotificationAdmin('Le nom du produit ' || OLD.nomProduit  
    || ' a été modifié en ' || NEW.nomProduit);  
END IF;  
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 CumparkKlimbFlutterBTS

Voici le lien pour accéder au projet :

<https://github.com/zedtroom/CumparkKlimbFlutterBTS>

## Conclusion

Le projet d'une application mobile demander par l'entreprise de revente Cumparklimb a été mener à bien les différentes fonctionnalités demandées ont été ajoutés à l'application. Cumparklimb peut désormais rendre les produits accessibles à tous les clients, même en période de confinement.

**ANNEXE IV -1 : fiche de présentation d'une situation professionnelle (modèle)**

<b>BTS Services informatiques aux organisations</b>	
<b>Session 2023</b>	
<b>E5 – Conception et maintenance de solutions informatiques</b>	
<b>Coefficient 4</b>	
<b>DESCRIPTION D'UNE SITUATION PROFESSIONNELLE</b>	
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>
OPTION SISR <input type="checkbox"/>	OPTION SLAM <input checked="" type="checkbox"/>
NOM et prénom du candidat : <b>CONCAS Travis</b>	
N° candidat : <b>02344511429</b>	
Contexte de la situation professionnelle <sup>1</sup> : <b>Contexte de la Maison des Associations d'Occitanie (MAO)</b>	
Intitulé de la situation professionnelle : <b>Site Web de vente en ligne</b>	
Période de réalisation :	Lieu : <b>Montpellier</b>
Modalité : <input checked="" type="checkbox"/> Seul <input type="checkbox"/> En équipe	
Principale(s) activité(s) concernée(s) <sup>2</sup> <b>Concevoir et développer une solution applicative</b>	
Conditions de réalisation <sup>2</sup> (ressources fournies, résultats attendus)  <a href="https://urlz.fr/qfqt">https://urlz.fr/qfqt</a>	
Productions associées : <b>VScode, MySQL PHPMyAdmin, AndroidStudio, ordinateur personnel</b>	
Modalités d'accès aux productions <sup>3</sup> :  <b>Vous devez indiquer ici les différentes URL (site web, bdd, ...) et les documents pour accéder à vos applications</b>	
Modalités d'accès à la documentation des productions <sup>4</sup> :  <a href="https://ifagparis-my.sharepoint.com/:f/g/personal/travis_concas_ecoles-epsi_net/EvMRH3k_7aRKgVpAF-D0xcQBtq8ara2IV306o_WPLPrc4A?e=J1L331">https://ifagparis-my.sharepoint.com/:f/g/personal/travis_concas_ecoles-epsi_net/EvMRH3k_7aRKgVpAF-D0xcQBtq8ara2IV306o_WPLPrc4A?e=J1L331</a>	
Au verso de cette page, le candidat présente un descriptif détaillé de la situation professionnelle et des productions réalisées sous forme d'un rapport d'activité permettant notamment de mettre en évidence la démarche suivie et les méthodes retenues.	

1 Conformément au référentiel du BTS SIO, le contexte doit être conforme au cahier des charges national en matière d'environnement technologique dans le domaine de spécialité correspondant à l'option du candidat.

2 En référence à la description des activités des processus prévue dans le référentiel de certification.

3 Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. Les candidats qui n'en sont pas munis sont pénalisés dans les limites prévues par la grille d'aide à l'évaluation proposée par la circulaire nationale d'organisation. ». Il s'agit par exemple des identifiant, mot de passe, URL d'un espace de stockage et de la présentation de l'organisation du stockage.

4 Lien vers le document décrivant la situation professionnelle tant au niveau logiciel (par exemple service fourni par la situation, interfaces utilisateurs, description des classes, de la base de données...) que matériel (par exemple schéma complet de réseau mis en place et configurations des services).

## ANNEXE VII-1 : modèle d'attestation de respect du cahier des charges pour l'épreuve E5 (parcours SLAM)

Nom : **CONCAS Travis**, Éléments complémentaires<sup>5</sup>

Éléments	Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)	Code exécuté sur <sup>6</sup>				Remarques de la commission d'interrogation <sup>7</sup>
		3.2.1 le système d'exploitation d'une STA fixe	3.2.2 un navigateur <i>web</i>	3.2.3 le système d'exploitation d'une STA mobile	3.2.4 le système d'exploitation d'un serveur	
Solution applicative N° 1	Visual Studio – Vue.JS MySQL – Procédures stockées, Trigger	X	X			
Solution applicative N°2	Android Studio – Framework Android Flutter MySQL – Procédures stockées, Trigger			X	X	

<sup>5</sup> Extrait du cahier des charges : « 3.2 Les activités de l'organisation cliente s'appuient sur **au 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 :

3.2.1 du code exécuté sur le système d'exploitation d'une solution technique d'accès fixe (type client lourd) ;

3.2.2 du code exécuté dans un navigateur *web* (type client léger ou riche, *applet*, etc.) ;

3.2.3 du code exécuté sur le système d'exploitation d'une solution technique d'accès mobile ;

3.2.4 du code exécuté sur le système d'exploitation d'un serveur (*servlet*, procédure cataloguée, etc.).

3.3 Une solution applicative peut être issue d'un développement spécifique ou de la modification du code d'un logiciel (*open source* par exemple).

3.4 **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.** »

<sup>6</sup> Indiquer une croix si la solution mise en œuvre intègre le type de code indiqué

<sup>7</sup> À renseigner pour les candidats passant l'épreuve sous forme ponctuelle.