

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ LACAN
Nom d'usage ▶ LACAN
Prénom ▶ NADEGE
Adresse ▶ 6 RUE ANDRE DE CHENIER APT 5439 81000 ALBI

Titre professionnel visé

Développeur web et web mobile

MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité p. 5

Compétences liées à l'activité :

- Maquetter une application
- Réaliser une interface utilisateur web statique et adaptable
- Développer une interface utilisateur web dynamique
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

- ▶ Bikes Stations : diagramme cas utilisation à la maquette finale p. 5
- ▶ Ido Shop : interface utilisateur site web p. 11
- ▶ Ido Shop : intranet interface de connexion utilisateur p. 17

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité p. 22

Compétences liées à l'activité :

- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile
- Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

- ▶ Adrar : MCD, MLD création base de données p. 22
- ▶ Bikes Stations : accès aux données, serveur de l'application p. 27

Titres, diplômes, CQP, attestations de formation (facultatif) p. 33

Déclaration sur l'honneur p. 34

Documents illustrant la pratique professionnelle (facultatif) p. 35

Annexes (Si le RC le prévoit) p. 36

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

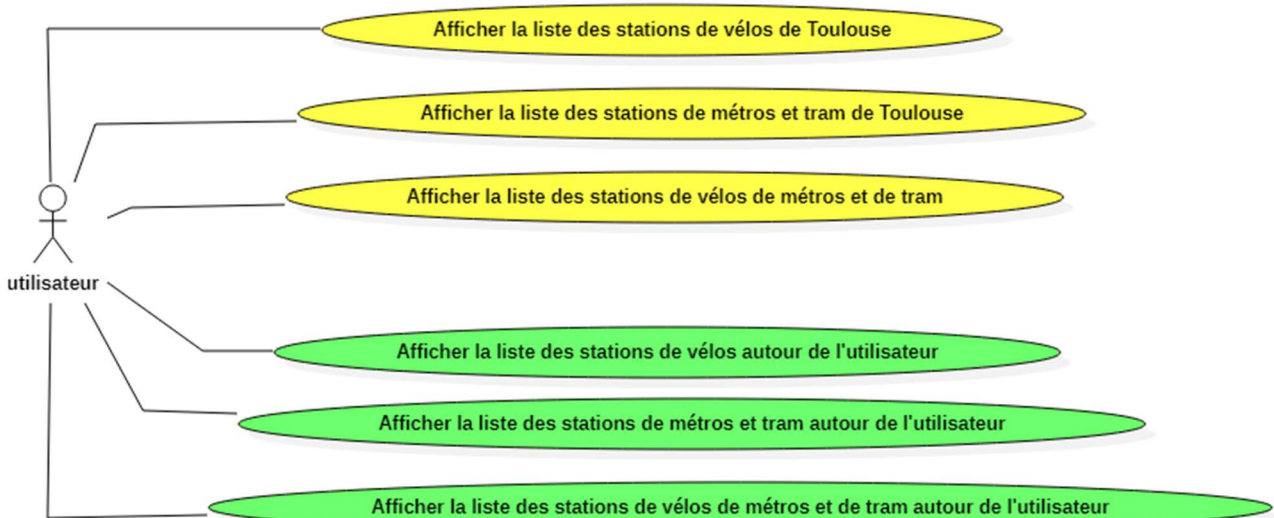
Exemple n°1 ► Bikes Stations : diagramme cas utilisation à la maquette finale

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

En cours, dans le cadre d'un TP de groupe, il nous a été demandé de réaliser une application mobile complète (back et front). Cette application Bikes Stations devait proposer l'affichage des stations de vélos et de métros sur l'agglomération Toulousaine.

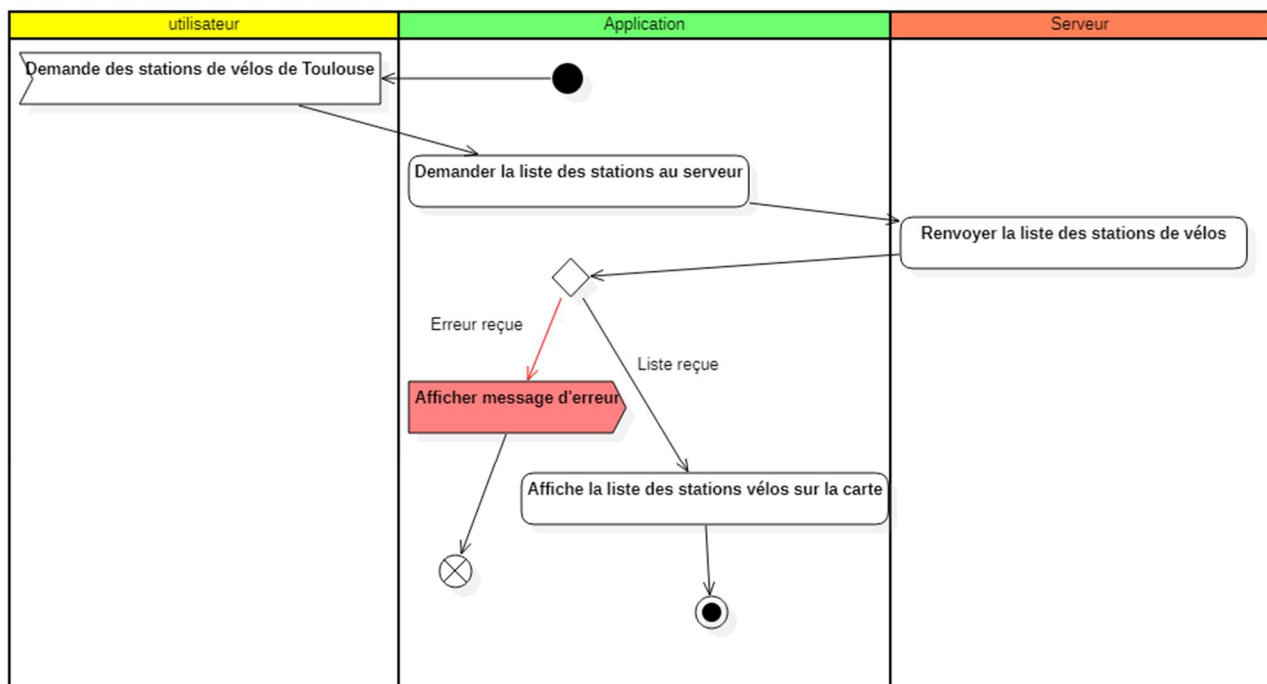
Dans le cadre de ce TP, pour le côté application cliente, nous avons rédigé un cahier des charges succinct, réalisé un diagramme des cas d'utilisation, d'activité, de séquence et de classe.

Voici le diagramme des cas d'utilisation aussi appelé UseCase, ici on peut voir toutes les utilisations qui peuvent être réalisées par l'utilisateur :

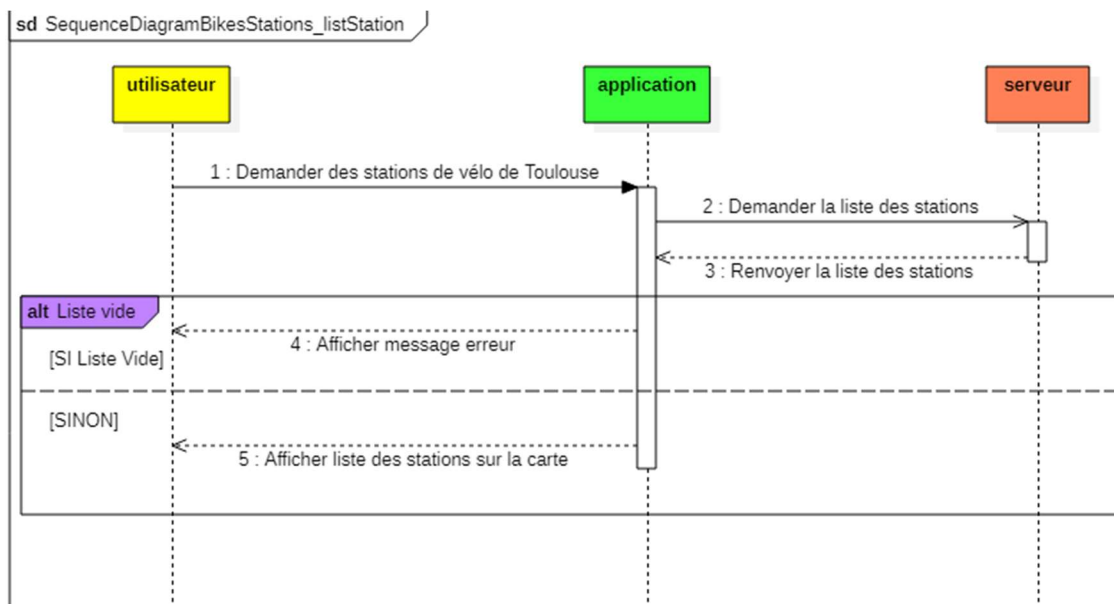


DOSSIER PROFESSIONNEL (DP)

Voici le diagramme d'activité correspondant au 1^{er} cas d'utilisation « Afficher la liste des stations de vélos de Toulouse ». Sur ce diagramme on peut visualiser le détail des étapes nécessaires pour ce cas d'utilisation. On peut également voir les accès aux différentes couches Application Cliente et Serveur.



Voici le diagramme de séquence correspondant :



DOSSIER PROFESSIONNEL (DP)

Le diagramme de classe ci-dessous détaille l'ensemble des classes qui seront à créer, les attributs de chaque classe, ainsi que les méthodes à implémenter.

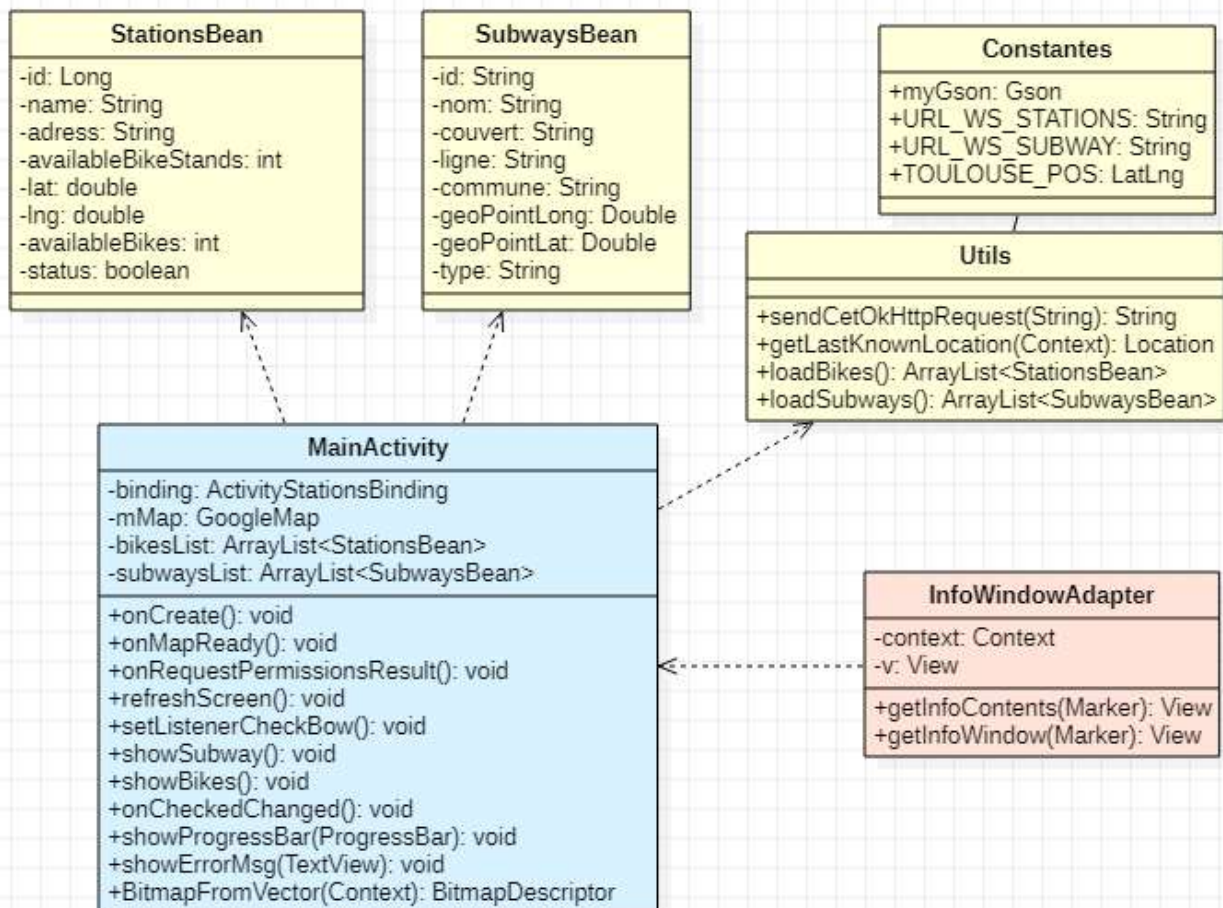
On peut voir la création de 4 classes en plus du MainActivity.

Les 2 classes de types Bean disposent d'attribut mais d'aucune méthode. Elles correspondent au JSON retourné par les 2 API utilisées.

La plupart des méthodes sont implémentées dans le MainActivity.

Certaines méthodes génériques sont déportées dans la classe Utils.

Certains attributs de type constante sont déclarés dans une classe à part Constantes, on y retrouve notamment les URL des API utilisées.

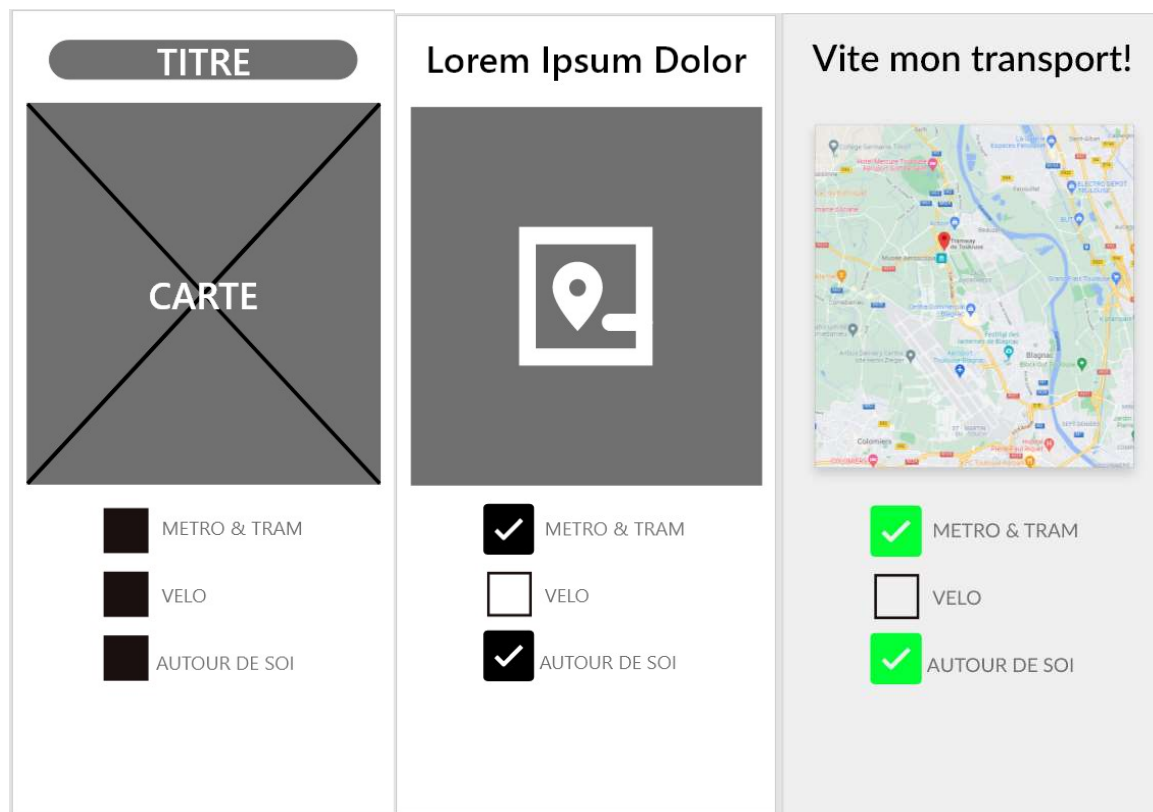


Ensuite nous avons effectué le maquetage de l'application mobile : le zoning, le wireframe et le mockup.

Tout d'abord c'est le zoning qui est établi il permet de mettre en place les différentes zones qui vont composer notre page, les zones images sont représentées par un carré avec une croix et les autres zones sont nommées.

Ensuite sur le wireframe, on donne on positionne peu à peu les éléments graphiques, ici les cases à cocher (checkbox) sont visibles ainsi que l'image qui se précise en carte (map).

Le mockup lui correspond à la maquette finale on peut y voir tous les éléments graphiques visuels du rendu final de notre page : la map, les couleurs, les images finales...



DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour effectuer ces tâches, j'ai utilisé les outils suivants :

- Logiciel de traitement de texte : cahier des charges
- StarUML : diagramme cas d'utilisation, activité et séquence
- Adobe XD : maquettage

3. Avec qui avez-vous travaillé ?

J'ai travaillé en équipe.

Nous étions 4 étudiants pour réaliser l'ensemble du TP, nous avons effectué les tâches ci-dessus en concertation à 4. Puis nous nous sommes réparti le développement du back et front en deux groupes de 2.

4. Contexte

Nom de l'entreprise, organisme ou association ► **ADRAR Pôle Numérique**

Chantier, atelier, service ► Cours sur la gestion de projet

Période d'exercice ► Du 13/12/2021 au 17/12/2021

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Cliquez ici pour taper du texte.

Activité-type 1

Exemple n°2 ▶

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Ido Shop : interface utilisateur site web

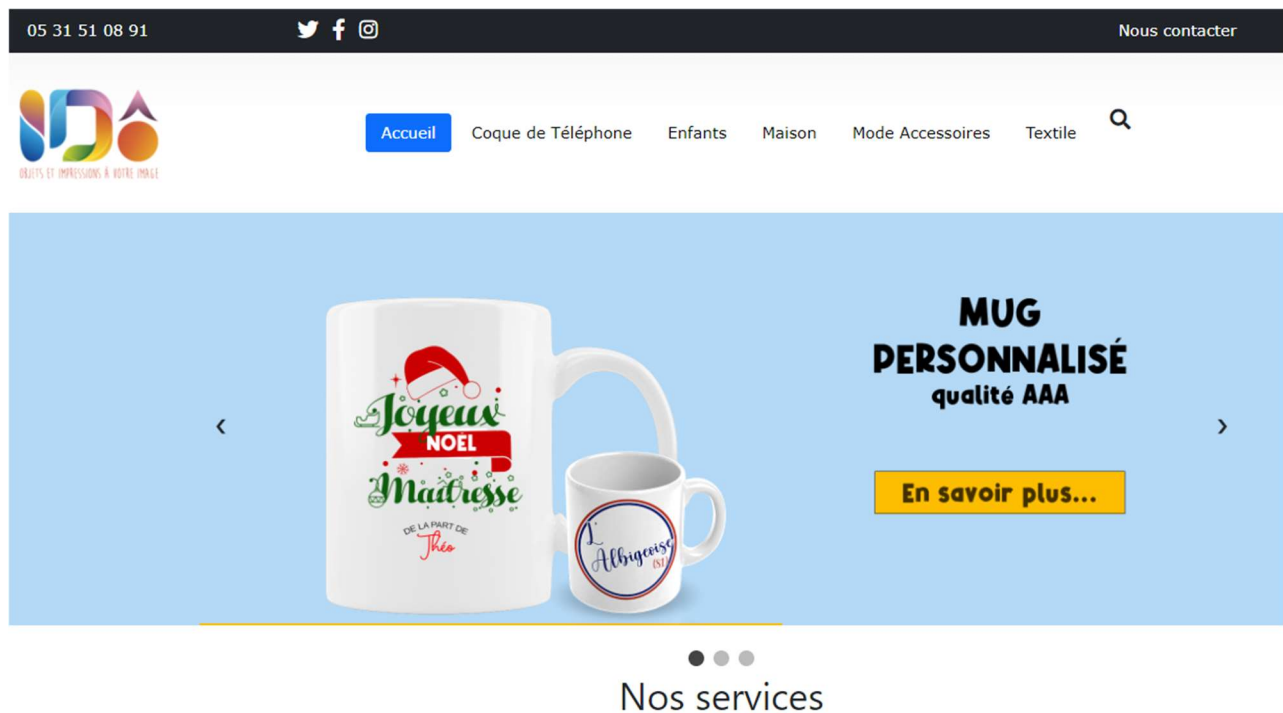
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre de mon stage en entreprise, j'ai participé à création du site internet Ido-Shop.fr.

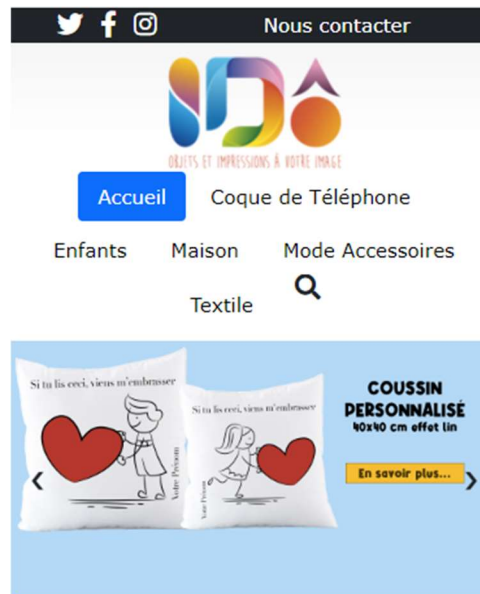
J'ai créé les pages internet en HTML.

Conformément aux maquettes et à la charte graphique, j'ai effectué la mise en page, ainsi que l'adaptabilité aux différentes tailles d'écran et navigateur avec des feuilles de style CSS.

Voici la page d'accueil du site sur pc :



Voici la page d'accueil sur mobile :



Nos services



Objets personnalisés



Sur l'extrait de code ci-dessous, on peut voir le code HTML correspondant au bandeau noir du header :

```
<div class="ligne_info_entete">
  <div class="forme_ligne_info">
    <p class="lien_telephone"></p>
    <span class="color-icon">
      <a href="tel:0531510891">05 31 51 08 91</a>
    </span>
    <div class="logo_reseau">
      <a href="https://www.twitter.com/idoshopalbi/" target="_blank">
        <i class="fab fa-twitter fa-lg color-icon"></i>
      </a>
      <a href="https://www.facebook.com/IdoAlbi/" target="_blank">
        <i class="fab fa-facebook-f fa-lg color-icon"></i>
      </a>
      <a href="https://www.instagram.com/IdoshopAlbi" target="_blank">
        <i class="fab fa-instagram fa-lg color-icon"></i>
      </a>
    </div>
  </div>
</div>
```

Ci-dessous, on peut voir l'utilisation des MediaQuery CSS pour masquer le numéro de téléphone à partir d'une largeur d'écran inférieure à 911 pixels.

Ici on utilise un sélecteur pour les balises span incluse juste après dans les balises de classe .forme_ligne_info

La valeur none à la propriété display pour masquer l'affichage.

```
/* RESPONSIVE LARGEUR ECRAN <911px */
@media only screen and (max-width: 911px) {

    .forme_ligne_info > span {
        display: none;
    }
}
```

L'affichage des données sur les pages a été réalisé en PHP en accédant à une Base de Données PHPMyAdmin.

Ci-dessous, un extrait de code qui effectue une requête SELECT pour afficher les familles de produit et la liste des produits sur le menu d'entête du site.

Dans cette 1^{ère} partie on récupère les données depuis la BDD grâce à la requête et on enregistre le résultat dans des tableaux.

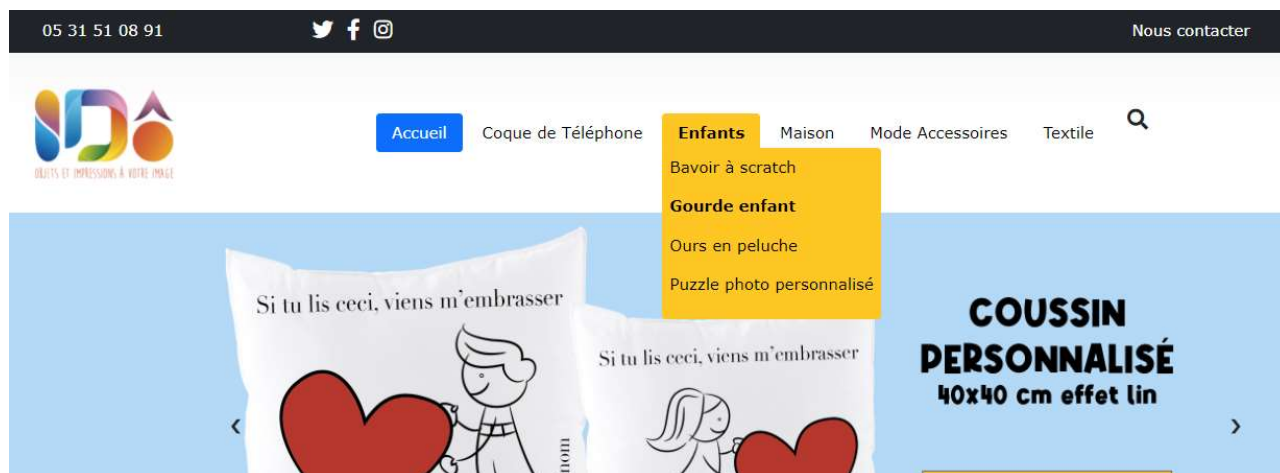
```
<?php
include "connectBDD.php";
// requete pour recuperer la liste des familles et des produits liés à cette famille
$sql = "SELECT * FROM APPARTENIR A INNER JOIN PRODUIT P ON A.Id_PRODUIT = P.Id_PRODUIT ";
$sql = $sql . "INNER JOIN FAMILLE_PRODUIT F ON A.Id_FAMILLE = F.Id_FAMILLE ";
$sql = $sql . "ORDER BY F.LIBELLE_FAMILLE, P.LIBELLE_COURT ";
$resultat = $pdo -> prepare($sql);
$resultat -> execute();
//on charge la liste des familles et produits dans un tableau
$id_famille = array();
$liste_famille = array();
$liste_produit = array();
while ($donnees = $resultat -> fetch()) {
    $id_famille[] = $donnees["Id_FAMILLE"];
    $liste_famille[] = $donnees["LIBELLE_FAMILLE"];
    $liste_produit[] = $donnees["LIBELLE_COURT"];
}

?>
```

Dans cette 2^{ème} partie de code on utilise les tableaux pour afficher le menu et le sous-menu avec du HTML :

```
<?php
$i=0;
$famille =null;
while ($i<sizeof($liste_famille)){
    if ($famille!=$liste_famille[$i]){
        echo '<div class="dropdown_menu">
        <span><a href="Nos-Produits-Famille.php?famille=' . $id_famille[$i] . '"> . $liste_famille[$i] . '</a></span>
        <div class="dropdown_content_menu">';
    }
    echo ' <p><a href="' . $liste_produit[$i] . '.php"> . $liste_produit[$i] . '</a></p>';
    $famille=$liste_famille[$i];
    $i++;
    if ($famille!=$liste_famille[$i]){
        if ($famille=="Textile"){
            echo ' <p><a href="https://ido.sowebshop.com/" target=_blank>Catalogue complet</a></p>';
        }
        echo '</div> </div>';
    }
}
?>
```

Voici le résultat produit sur le site :



Les animations du carrousel de la page d'accueil du site ont été réalisé en JS et CSS.

Le carrousel est réalisé en JS, et l'animation de chaque slide du carrousel est réalisée en CSS.

Ci-dessous on peut voir que l'animation `fadeInS1` est appliqué aux balises dont l'id est `imglien_carrousel1` ou `imglien_carrousel2` ou `imglien_carrousel3`, et aura une durée de 15 secondes.

```
#imglien_carrousel1, #imglien_carrousel2, #imglien_carrousel3 {
    -webkit-animation-name: fadeInS1;
    -webkit-animation-duration: 15000ms;
    animation-duration: 15000ms;
    animation-name: fadeInS1;
}
```

Ici on peut voir que l'animation effectue un affichage avec un effet de zoom à 2% de la durée totale de l'animation.

```
@keyframes fadeInS1 {
  0% {
    opacity: 0;
    webkit-transform: scale(0); /* Safari et Chrome */
    transform: scale(0);
  }
  2% {
    opacity: 1;
    webkit-transform: scale(1); /* Safari et Chrome */
    transform: scale(1);
  }
  100% {
    opacity: 1;
    webkit-transform: scale(1); /* Safari et Chrome */
    transform: scale(1);
  }
}
```

Sur ce projet, les framework Bootstrap (pour les cards des articles notamment) et Fontawesome (pour les icones du header) ont également été utilisés.

Le site internet est sur un serveur d'hébergement web Ionos et publié.

2. Précisez les moyens utilisés :

J'ai utilisé les outils suivants :

- IntelliJ : pour le code en langage HTML, CSS, PHP et JS
- Bootstrap et Fontawesome : comme framework
- PhpMyAdmin : pour la base de données
- Filezilla : pour le transfert des fichiers sur le serveur
- Ionos : serveur d'hébergement

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Principalement seule. Dans une entreprise avec uniquement le gérant, demandeur du projet, mais qui n'est pas compétent en informatique, et un alternant en informatique qui n'était présent qu'à 50%.

4. Contexte

Nom de l'entreprise, organisme ou association ► **IDO Shop ALBI**

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du **04/10/2021** au **26/11/2021**

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°3 ▶

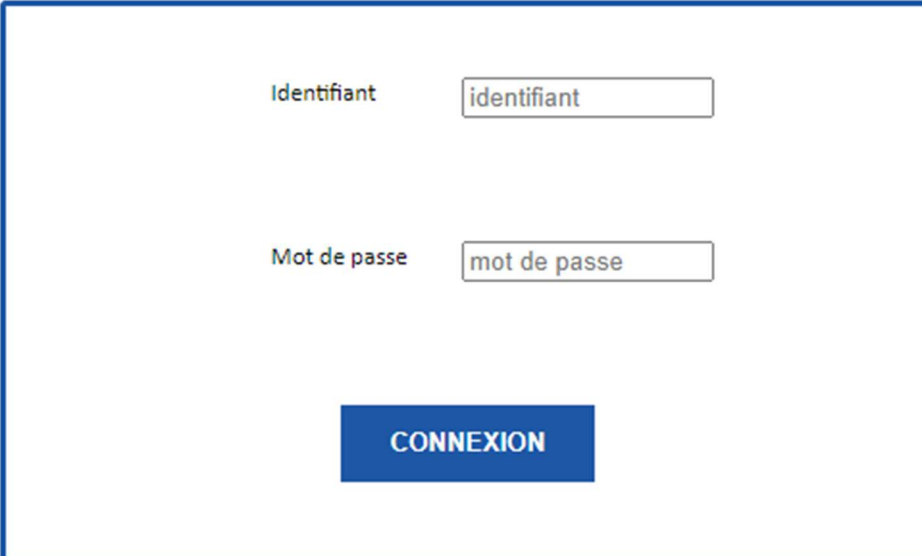
Ido Shop : intranet interface de connexion utilisateur

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre de mon stage, j'ai été amené à réaliser une partie intranet permettant la gestion du back-office du site internet.

Cet intranet a été développé en HTML, CSS et PHP. Je vais vous présenter ici l'interface de connexion de l'utilisateur à l'intranet.

Voici le formulaire de connexion de la page d'accueil de l'intranet :



Identifiant

Mot de passe

CONNEXION

Ci-dessous, vous pouvez voir le code HTML de ce formulaire de connexion. On peut voir que ce formulaire est type POST et lors de sa validation le fichier connectUSER.php est exécuté.

On peut voir ici que les champs du formulaire (input) sont obligatoires en saisies car nous avons précisé requiered dans les propriétés et le champ mot de passe ne peut pas être inférieur à 8 caractères (propriété minlength).

```
<!-- ZONE DE CONNEXION -->
<div class="connexion">
  <form method="POST" action="connectUSER.php">
    <fieldset class="formConnexion">
      <div class="champConnexion">
        <label for="emailConnexion">Identifiant</label>
        <input type="email" name="emailConnexion" id="emailConnexion" requiered placeholder="identifiant"/>
      </div>
      <div class="champConnexion">
        <label for="pwdConnexion">Mot de passe</label>
        <input type="password" name="pwdConnexion" id="pwdConnexion" minlength=8 requiered placeholder="mot de passe"/>
      </div>
      <div class="champConnexion">
        <input type="submit" name="validationConnexion" id="validationConnexion" value="CONNEXION"/>
      </div>
    </fieldset>
  </form>
</div>
```

Voici le code du fichier connectUSER.php.

On peut voir que ce fichier récupère les données saisies par l'utilisateur grâce aux variables \$_POST.

On recherche ensuite ces données dans notre BDD par le biais d'une requête SELECT et on contrôle le mot de passe saisi via la méthode password_verify car le mot de passe est crypté en BDD.

Ici si l'utilisateur est trouvé on lance la page d'accueil de l'intranet connecté accueil.php

S'il n'est pas trouvé, on effectue un message d'alerte et on redirige vers la page de connexion.

```
<?php
session_start();
$_SESSION['login']=$_POST['emailConnexion'];
$_SESSION['LAST_ACTIVITY'] = time();

include 'connectBDD.php';

// On récupère dans des variables les données saisies dans le formulaire de connexion
$email=$_POST['emailConnexion'];
$password=$_POST['pwdConnexion'];

try {
    // Requête pour vérifier si l'utilisateur existe et récupérer son mot de passe
    $resultat = $pdo->prepare("SELECT * FROM UTILISATEURS WHERE MAIL=?");
    $resultat->execute(array($email));

    //Afficher l'accueil si OK ou msg erreur si KO et renvoi sur la page de connexion
    while ($donnees = $resultat->fetch()){
        if (password_verify($password, $donnees['PWD'])){
            echo "<script type='text/javascript'>document.location.replace('accueil.php');</script>";
            exit();
        }
    }
    echo "<script type='text/javascript'>window.alert('Identifiant incorrect'); document.location.replace('ido.php');</script>";
    $_SESSION['login']="";
    exit();
}
catch (PDOException $exception) {
    exit('Erreur de connexion à la base de données');
}
?>
```

DOSSIER PROFESSIONNEL (DP)

Voici l'écran affiché si l'utilisateur est trouvé, on peut voir que le nom de l'utilisateur connecté apparaît en haut à gauche.

Utilisateur :contact@ido-shop.fr
jeudi 06 janvier 2022, 16:34



Liste des contacts

PRODUITS

Nouveau produit
Mise à jour des produits
Liste des produits à publier

CATEGORIE PRODUITS

Nouvelle catégorie
Mise à jour catégorie

FAMILLE PRODUITS

Nouvelle famille
Mise à jour famille

SOUS PRODUITS

Nouveau sous-produit
Mise à jour sous-produit

CARROUSEL

Mise à jour du Carrousel

2. Précisez les moyens utilisés :

J'ai utilisé les outils suivants :

- **IntelliJ** : pour le code en langage HTML, CSS et PHP
- **PhpMyAdmin** : pour la base de données
- **Filezilla** : pour le transfert des fichiers sur le serveur
- **Ionos** : serveur d'hébergement

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Saule, dans une entreprise avec uniquement le gérant, demandeur du projet, mais qui n'est pas compétent en informatique, et un alternant en informatique qui n'est pas intervenu sur la partie intranet.

4. Contexte

Nom de l'entreprise, organisme ou association ► **Idô Shop Albi**

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du **04/10/2021** au **26/11/2021**

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°1 ► Adrar : MCD, MLD création base de données

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pendant ma formation, dans le cadre d'un TP, il nous été demandé de réaliser l'analyse et la base de données permettant la gestion des employés de l'ADRAR.

Le prof nous a expliqué le fonctionnement de l'ADRAR et les attentes.

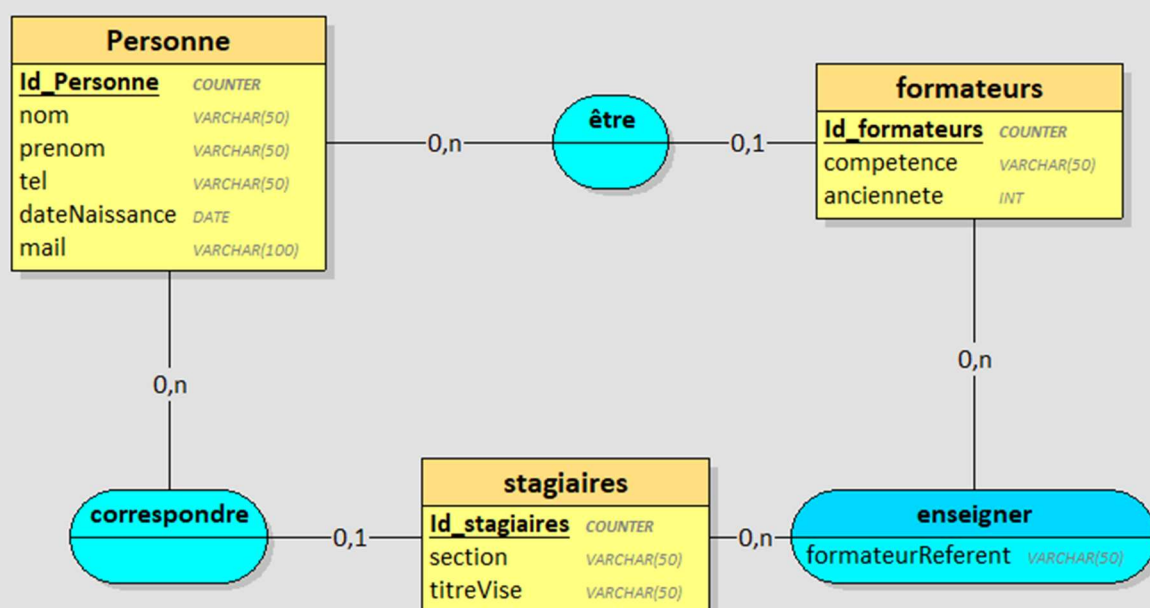
A partir de cet échange j'ai pu :

- Effectuer une analyse des besoins
- Identifier les données à stocker
- Créer le Modèle Conceptuel des Données MCD
- A partir du MCD créer le Modèle Logique des Données MLD
- A partir du MLD écrire le script SQL de création de la base de données

Voici le MCD, on peut y visualiser les 3 entités, les 3 associations ainsi que les cardinalités de chacune de ces associations.

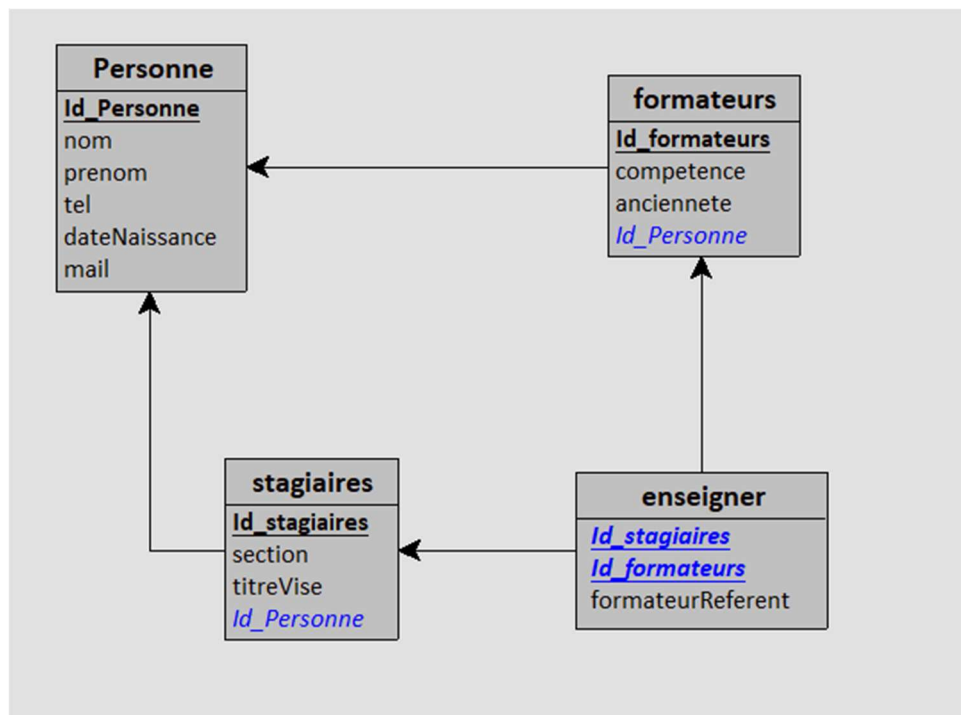
Chaque entité dispose d'une clé primaire de type auto incrémentée.

Ici l'association enseigner à une cardinalité 0.N de chaque côté, car un formateur peut enseigner à plusieurs stagiaires, et un stagiaire peut se faire enseigner par plusieurs formateurs différents.



Voici le MLD correspondant, on peut voir que l'association être se traduit par l'absorption par la table formateurs de la clé primaire de la table personnes. Ici ce sera une clé étrangère.

L'association enseigner, qui elle avait une cardinalité 0.N de chaque côté se traduit par la création d'une table d'association dont la clé primaire est composée de l'association des 2 clés étrangères absorbées depuis les tables stagiaires et formateurs.



Voici un extrait du script de création de la BDD correspondante. Ici ce sont les créations des tables stagiaires et enseigner.

```
create table stagiaires (  
    idStagiaire bigint not null primary key auto_increment,  
    section varchar(50),  
    titreVise varchar(50),  
    idPersonne bigint,  
    foreign key (idPersonne) references personnes(idPersonne)  
);  
create table enseigner (  
    idStagiaire bigint not null,  
    idFormateur bigint not null,  
    formateurReferent varchar(50),  
    foreign key (idStagiaire) references stagiaires(idStagiaire),  
    foreign key (idFormateur) references formateurs(idFormateur),  
    primary key (idStagiaire,idFormateur)  
);
```

Voici un extrait du jeu de test, les insertions des données dans les tables stagiaires et enseigner, avec le résultat obtenu.

Ici l'ordre d'exécution des INSERT à son importance, on ne peut faire référence à un IdStagiaire dans enseigner uniquement s'il est au préalable existant dans la table stagiaires.

TABLE stagiaires :

```
insert into stagiaires (section, titreVise, idPersonne) values
("dev","concepteur",3),
("reseau","admin reseau",4),
("dev","developpeur web",1),
("dev","developpeur js",2);
```

idStagiaire	section	titreVise	idPersonne
1	dev	concepteur	3
2	reseau	admin reseau	4
3	dev	developpeur web	1
4	dev	developpeur js	2
NULL	NULL	NULL	NULL

TABLE enseigner :

```
insert into enseigner (idStagiaire, idFormateur, formateurReferent) values
(1,1,"florence calmettes"),
(2,1,"jonathan"),
(3,2,"ophelie"),
(3,1,"matthieu"),
(4,3,"florence");
```

idStagiaire	idFormateur	formateurReferent
1	1	florence calmettes
2	1	jonathan
3	1	matthieu
3	2	ophelie
4	3	florence
NULL	NULL	NULL

Voici 2 exemples de requêtes SELECT effectuées sur cette base de données.

La 1^{ère} permet de calculer l'ancienneté moyenne des formateurs en utilisant la fonction avg sur le champ anciennete de la table formateurs :

```
select avg(anciennete)
from formateurs;
```

avg(anciennete)
6.2500

DOSSIER PROFESSIONNEL (DP)

La 2^{ème} permet l’affichage des formateurs avec leurs nom, prénom et compétence trié par ordre alphabétique sur le nom de famille. Pour faire cet affichage nous avons réalisé une jointure entre les tables formateurs et personnes par le biais du champ IdPersonne.

```
select personnes.nom, personnes.prenom, formateurs.compétences
from personnes
inner join formateurs on formateurs.idPersonne=personnes.idPersonne
order by personnes.nom;
```

nom	prenom	compétences
calmettes	florence	sql algo
prof1	preprof1	git reseau
prof2	preprof2	maquetage uml
prof3	preprof3	reseau sociaux com

2. Précisez les moyens utilisés :

Pour effectuer ces tâches, j’ai utilisé :

- Looping : pour la création du MCD et du MLD
- MySQL : pour la création de la Base de Données BDD

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Seule en autonomie.

4. Contexte

Nom de l'entreprise, organisme ou association ► **ADRAR Pôle Numérique**

Chantier, atelier, service ► Cours sur conception MCD MLD SQL

Période d'exercice ► Du 08/06/2021 au 11/06/2021

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°2 ► Bikes Stations : accès aux données, serveur de l'application

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre d'un TP de groupe, nous avons dû réaliser le back-end d'une application mobile. Dans ce cadre, nous avons créé un serveur avec SpringBoot afin de mettre un Webservice à disposition de notre application Android cliente.

Ce Webservice renvoi la liste des stations de vélos de l'agglomération Toulousaine.

Toutes les 10 minutes, les données sont chargées depuis l'appel à une API et stockées dans notre BDD. Pour ce faire, nous avons créé 3 Beans différents BikesStationsBean et LocationBean qui correspondent à la structure du JSON retourné par l'API interrogé, et StationsBean qui correspond à l'architecture de notre table en BDD et qui sera utilisé pour stocker les données récupérées dans notre table.

Voici le code de ces 3 Beans :

```
public class BikesStationsBean {  
  
    private String name;  
    private String address;  
    private int available_bike_stands;  
    private int available_bikes;  
    private String status;  
    private LocationBean position;  
  
@Entity  
@Table(name = "stations")  
public class StationsBean {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private long id;  
    private String name;  
    private String address;  
    private int available_bike_stands;  
    private int available_bikes;  
    private boolean status;  
    private double lat;  
    private double lng;  
  
    public class LocationBean {  
        private double lat;  
        private double lng;  
    }  
}
```

Nous avons également créé une méthode `getStation()`, qui s'occupe de faire appel à une API et de retourner le résultat de cette API sous forme de liste de `BikesStationsBean`.

```
public static List<BikesStationsBean> getStation() throws Exception {
    //Toujours effectuer un contrôle d'url en l'affichant en console
    String url = "https://api.jcdecoux.com/vls/v1/stations?contract=Toulouse&apiKey=2a1b07b2a523f81188fe34e348206a57ffa6f2a7";
    System.out.println("Url : " + url);
    OkHttpClient client = new OkHttpClient();
    //Création de la requête
    Request request = new Request.Builder().url(url).build();
    //Exécution de la requête
    Response response = client.newCall(request).execute();
    //Analyse du code retour
    if (response.code() < 200 || response.code() > 299) {
        throw new Exception("Réponse du serveur incorrect : " + response.code());
    } else {
        Gson g = new Gson();
        List<BikesStationsBean> maListeStations = g.fromJson(response.body().string(), new TypeToken<List<BikesStationsBean>>() {
        }.getType());
        //Résultat de la requête
        return maListeStations;
    }
}
```

Afin d'actualiser notre BDD toutes les heures nous avons créé une classe ScheduledTasks afin de créer une tâche programmée. Ici la méthode refreshData() est programmée toutes les 600000ms soit chaque 10 minutes. Cette méthode permet de récupérer les données de l'API par l'intermédiaire de getStation() et de transformer la liste de BikesStationsBean en liste de StationsBean. Une fois les données enregistrées au bon format on peut effectuer la mise à jour de notre table en BDD, grâce à l'interface stationDAO et les méthodes deleteAll() et saveAll().

```
@Component
public class ScheduledTasks {
    @Autowired
    private StationsDAO stationDao;

    // programmation automatique toutes les 10 minutes de la méthode refreshData()
    @Scheduled(fixedRate = 600000)
    public void refreshData() throws Exception {
        System.out.println("refreshData()");
        // chargement de la liste des stations depuis getStation()
        List<BikesStationsBean> listeStation = OkHttpUtils.getStation();
        ArrayList<StationsBean> listeStationBDD = new ArrayList<>();

        // boucle sur la liste résultat de getStation() pour récupérer les attributs utiles pour créer des StationsBean
        for (var i=0;i<listeStation.size();i++){
            String nameBDD = listeStation.get(i).getName();
            String addressBDD = listeStation.get(i).getAddress();
            int availableBikesStandsBDD = listeStation.get(i).getAvailable_bike_stands();
            int availableBikesBDD = listeStation.get(i).getAvailable_bikes();
            boolean statusBDD=false;
            if (listeStation.get(i).getStatus().equals("OPEN")){
                statusBDD = true;
            }
            Double latBDD = listeStation.get(i).getPosition().getLat();
            Double lngBDD = listeStation.get(i).getPosition().getLng();
            // ici on crée un nouveau StationsBean pour chaque enregistrement de la liste et on l'ajoute à la nouvelle liste adaptée pour la BDD
            StationsBean station = new StationsBean(nameBDD,addressBDD,availableBikesStandsBDD, availableBikesBDD, statusBDD,latBDD,lngBDD );
            listeStationBDD.add(station);
        }

        if (!listeStationBDD.isEmpty()){
            // grâce à l'interface stationDAO on vide la table de la BDD et on l'alimente avec la nouvelle liste à jour
            stationDao.deleteAll();
            stationDao.saveAll(listeStationBDD);
        }
        else {
            throw new Exception("Liste vide BDD non mise à jour");
        }
    }
}
```

Enfin, afin de mettre un Webservice à disposition, nous avons créé une classe Webservice de type Contrôleur Rest :

```
@RestController
public class WebServices {

    @Autowired
    private StationsDAO stationDao;
```

Dans cette classe nous avons implémenté une méthode /GetBikesInToulouse() de type GetMapping qui retourne une liste de StationsBean au format JSON. Ici ce sont les données stockées dans notre BDD qui sont renvoyées grâce à l'interface stationDAO et la méthode findAll()

```
@GetMapping("/getBikesInToulouse")
public List<StationsBean> getBikesInToulouse() {
    System.out.println("/getBikesInToulouse");
    return stationDao.findAll();
}
```

Voici un extrait du JSON retourné par ce Web Service :

```
[
  {
    "id": 39756,
    "name": "00055 - SAINT-SERNIN - G. ARNOULT",
    "available_bike_stands": 2,
    "available_bikes": 13,
    "status": true,
    "lat": 43.6089519604964,
    "lng": 1.4410035987262,
    "adress": "2 RUE GATIEN ARNOULT"
  },
  {
    "id": 39757,
    "name": "00195 - LARROUSSE - RIEUX",
    "available_bike_stands": 13,
    "available_bikes": 5,
    "status": true,
    "lat": 43.5972354030358,
    "lng": 1.45907112459247,
    "adress": "FACE 15 RUE PIERRE LAROUSSE"
  },
  {
    "id": 39758,
    "name": "00029 - VALADE",
    "available_bike_stands": 3,
    "available_bikes": 16,
    "status": true,
    "lat": 43.605071904633625,
    "lng": 1.437032051956223,
    "adress": "31 RUE VALADE"
  }
]
```

2. Précisez les moyens utilisés :

Pour réaliser cette partie serveur, nous avons utilisé :

- IntelliJ comme IDE
- JAVA comme langage de développement
- SpringBoot et OkHttp comme framework
- MySQL comme BDD

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

Nous étions 4 étudiants pour réaliser le projet complet, et nous étions 2 pour réaliser cette partie back-end.

4. Contexte

Nom de l'entreprise, organisme ou association ► **ADRAR Pôle Numérique**

Chantier, atelier, service ► Cours sur la gestion de projet

Période d'exercice ► Du 13/12/2021 au 17/12/2021

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] *Nadège LACAN* ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je
suis l'auteur(e) des réalisations jointes.

Fait à *Albi* le *06/01/2021*

pour faire valoir ce que de droit.

Signature :



DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

ANNEXES

(Si le RC le prévoit)