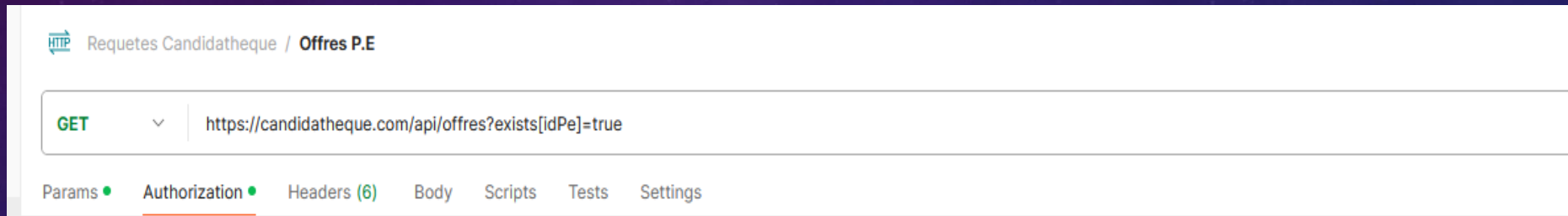


PRÉSENTATION DU PROJET



TESTS DE REQUÊTES AVEC POSTMAN



LES STRUCTURES

The screenshot displays a software development environment with a code editor and a variable declaration table.

Code Editor: The main window shows a Pascal-like structure definition for `infosD2Api`. The code is as follows:

```
40 // *** Structure d'infos des deux Api (Candidathèque et Vérie): *** //
41
42 infosD2Api est une Structure
43 // ***** Candidathèque : *****
44 departement est une chaîne
45 metierLibelle est une chaîne
46 nomEntreprise est une chaîne
47 // ***** Vérie(Etablissement) : *****
48 VerifyUrl est une chaîne
49 VerifyEtabl est une chaîne
50 VerifyCpEtabl est une chaîne
51 CodeApeVery est une chaîne
52 LibApeVery est une chaîne
53 sAdrEtabl est une chaîne
54 // ***** //
55 sAdresseComplete est une chaîne
56 activitePrincipaleEtablissement est une chaîne
57 sEnseigne1Etablissement est une chaîne
58 sEnseigne2Etablissement est une chaîne
59 sEnseigne3Etablissement est une chaîne
60 sNumeroVoieEtablissement est une chaîne
61 sTypeVoieEtablissement est une chaîne
62 sLibelleVoieEtablissement est une chaîne
63 sComplementAdresseEtablissement est une chaîne
64 sCodePostalEtablissement est une chaîne
65 sLibelleCommuneEtablissement est une chaîne
```

Variable Declaration Table: Below the code editor, a table lists variables and their types. The table has columns for Groups, Members, Properties, and Procedures.

Groups	Members	Properties	Procedures
<Élément courant> : FEN_Prospexion	AdiConcat	BoolNomEntr	CandNumDep
COL_Candidatheque	AuthToken	BoolNomEntr2	CandUrl
COL_Verie	BoolAdr	CandCodeDepUrl	CodeApeVery
	BoolAdi2	candNomDep	CodeDep
			ErriInfoApi
			InfosApe
			InfosGlobal
			LibApeVery
			NomDep
			OffresPe
			Query1
			Query2
			sAdrEtabl
			sInfosCommune
			tabApe
			TabInfosGlobal
			TraitementOK
			UriBase
			UriQuery
			VerifyCpEtabl
			VerifyEtabl
			VerifyUrl
			VerifyNumDep
			infosD2Api
			stauth
			StrApe

The status bar at the bottom indicates the current line is 17 (Ln 17), column 1 (Col 1), and the window title is FEN_Prospexion.

CHARGER UNE LISTE AU FORMAT CSV

```
// *** Charger le fichier.csv : *** //
CsvResultat      est une chaîne
//<Résultat> = fChargeTexte(<Nom et chemin du fichier Texte> [, <Mode de chargement>])
// *** Entreprise: ***
CsvResultat = UTF8VersAnsi(fChargeTexte("D:\Mes Projets\Projet_Prospection\CodeAPELibelle.csv"))
// *** Domicile: ***
//CsvResultat est une chaîne = fChargeTexte("C:\Users\Chris\Documents\GitHub\Projet_stage_CDA\Sauv

// *** Remplir mon tableau (structure): *** //
//CSVVersTableau(<Chaîne> , <Tableau> [, <Séparateur de colonne>])
CSVVersTableau(CsvResultat, tabApe,";")
tabApe.Supprime(1)
ListeAffiche[COMBO_SA]

// ***** Token Refresh : ***** //
stauth          est une Structure
    Token        est une chaîne
    RefreshToken est une chaîne
    DatePeremption est un Date
FIN
Authtoken est un stauth
```


LES VARIABLES

```
■ Déclarations globales de FEN_Prospexion  Si Erreur : par programme Quand Exception : par programme
1
2 ■ PROCÉDURE MaFenêtre()
3
4   UrlBase           est une chaîne
5   UrlQuery          est une chaîne
6   Query1            est une chaîne
7   Query2            est une chaîne
8   TraitementOk      est un booléen
9   // *** Chaîne liste combo département: *** //
10  // "Ardennes - 08"
11  comboNomDep        est une chaîne = ExtraitChaine(COMBO_Département..Valeur, rangDernier, " - ")
12  comboNomDep = SansEspace(comboNomDep)
13  //infos(comboNomDep) // Exemple visuel résultat = "Ardennes"
14  comboNumDep        est une chaîne = ExtraitChaine(COMBO_Département..Valeur, rangPremier, " - ")
15  comboNumDep = SansEspace(comboNumDep)
16  //infos(comboNumDep) // Exemple visuel résultat = "08"
17  AdrConcat          est une chaîne = comboNomDep + " " + comboNumDep
18  BoolAdr            est un booléen
19  BoolAdr2           est un booléen
20  ErrInfoApi         est un booléen
21  // *** Candidathèque *** //
22  CandUrl            est une chaîne = "https://candidatheque.com/api/"
23  OffresPe           est une chaîne = "offres?exists[idPe]=true"
24  CodeDep            est une chaîne = "&departement.code="
25  NomDep             est une chaîne = "&departement.nom="
26  CandCodeDepUrl     est une chaîne = CandUrl + OffresPe + CodeDep + comboNumDep
27  CandNumDep         est une chaîne = CandUrl + OffresPe + CodeDep
28  candNomDep         est une chaîne = CandUrl + OffresPe + NomDep
29  BoolNomEntr        est un booléen
30  BoolNomEntr2       est un booléen
31  // *** Verify *** //
32  sAdrEtabl          est une chaîne
33  VeryNumDep         est une chaîne
34  sInfosCommune      est une chaîne
35  VerifyUrl          est une chaîne = "https://verify.fr/api/v1/"
36  VeryEtabl          est une chaîne = VerifyUrl + "etablisements"
37  VeryCpEtabl        est une chaîne = VeryEtabl + "?codePostalEtablissement=" + comboNumDep
38  CodeApeVerify      est une chaîne
39  LibApeVerify       est une chaîne
```

LES PROCÉDURES

```
WL Procédure globale Token_Verifie * Si Erreur : par programme Quand Exceptio

PROCÉDURE Token_Verifie(sLogin est une chaîne="", sMdp est une chaîne="")

    vToken    est un Variant
    vCompte    est un Variant
    vCompte.username    = "NomUtilisateur"
    vCompte.password    = "MotDePasse"
    sLogin        = vCompte
    sMdp          = vCompte

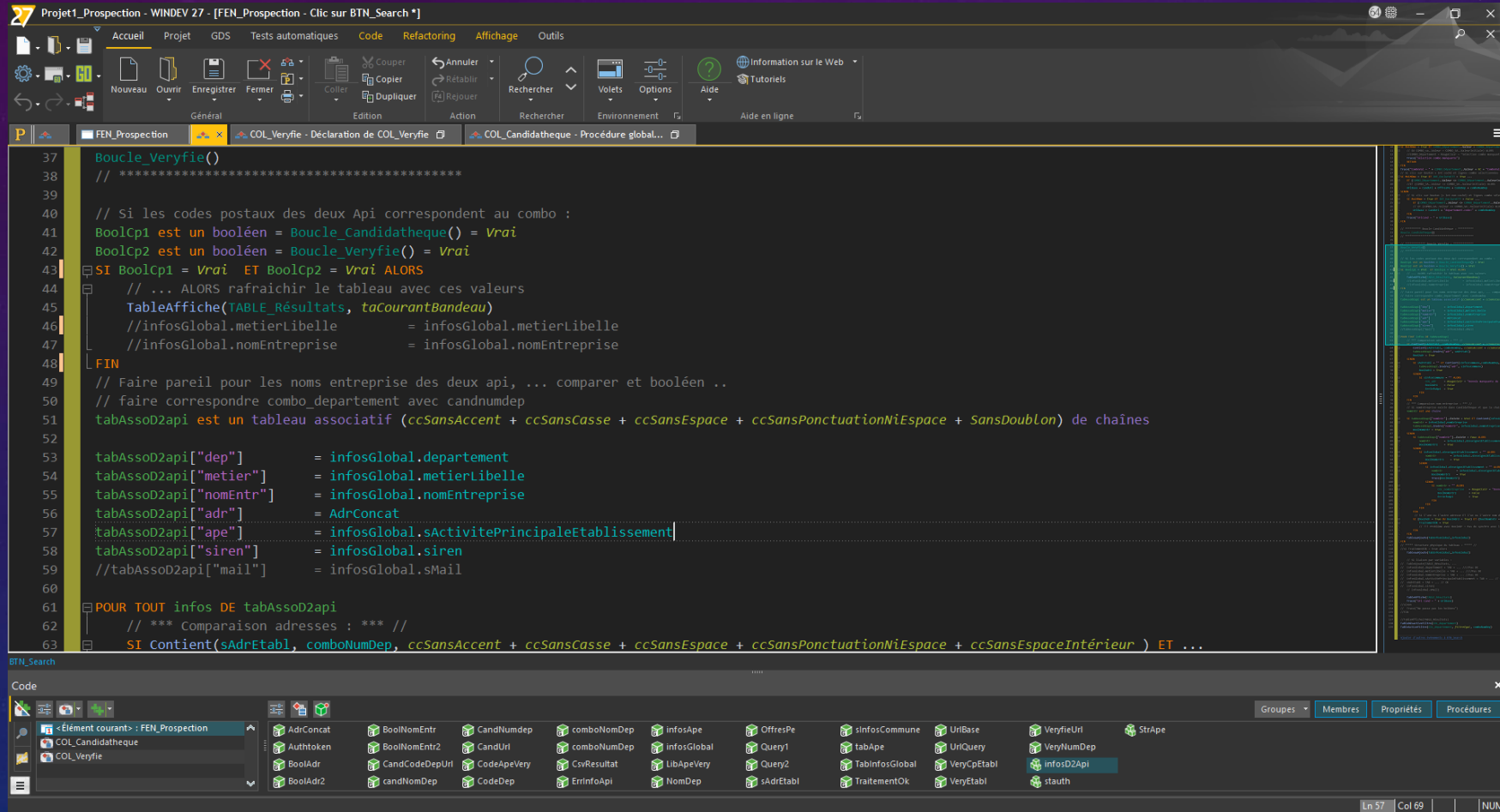
    // Lire les informations d'identification à partir d'un fichier INI sécurisé
    //INILit("C:\Donnees\Verifie.ini")

    SI vCompte.username = "" OU vCompte.password = "" ALORS
        Erreur("Les informations d'identification de l'API Verifie sont manquantes.")
        RENVOYER ""
    FTN
```

LES REQUÊTES

```
PROCEDURE Boucle_Verifie()  
  
  // ***** Boucle Verifie : ***** //  
  vReponseVerifie est un Variant = Verifie_Query()  
  BoolCp2         est un booléen  
  
  SI vReponseVerifie.Erreur..Existe _ET_ PAS vReponseVerifie.Erreur ALORS  
  POUR TOUT infos DE vReponseVerifie.Resultats  
    //SI infos <> "" ET infos <> Null ALORS  
  
    Cp      est une chaîne = infos.codePostalEtablissement  
    // S'il ne trouve pas le code postal établissement, alors il prend le Cp commune  
    SI infos.codePostalEtablissement = "" ALORS  
      Cp = infos.codeCommuneEtablissement  
    FIN  
    VeryNumDep est une chaîne = FEN_Prospection.VeryNumDep  
    VeryNumDep = ExtraitChaîne(Cp,1,"/",DepuisFin)  
    VeryNumDep = SansEspace(VeryNumDep)  
    //VerifieNomDep  
    comboNumDep est une chaîne = FEN_Prospection.comboNumDep  
    // Si ma variable département(Very) contient le numéro département sélectionné dans mon combo alors ...  
    SI VeryNumDep = Contient(Cp, comboNumDep, SansCasse) ALORS  
      Query2 est une chaîne = FEN_Prospection.VeryCpEtabl  
      BoolCp2 = Vrai  
    FIN  
  
    sAdrEtabl      est une chaîne  
    sInfosCommune  est une chaîne  
    sCodePostalEtablissement est une chaîne  
    sActivitePrincipaleEtablissement est une chaîne // = FEN_Prospection.infosGlobal.sActivitePrincipaleEtablissement  
    siren          est une chaîne // = FEN_Prospection.infosGlobal.siren  
    // *** Etablissement : *** //  
    sInfosCommune = ...  
    infos.codeCommuneEtablissement + " " + infos.libelleCommuneEtablissement  
    sCodePostalEtablissement = infos.codePostalEtablissement  
    sAdrEtabl = ...  
    infos.numeroVoieEtablissement + " " + infos.typeVoieEtablissement + " " + infos.libelleVoieEtablissement + ", " + ...  
    infos.codeCommuneEtablissement + " " + infos.libelleCommuneEtablissement  
    sActivitePrincipaleEtablissement = infos.activitePrincipaleEtablissement //code APE //Trace ok  
    // *** Unite légale : *** //  
    siren = infos.uniteLegale.siren  
    //sMail = infos.uniteLegale.mail  
    //FIN  
  
    FIN  
    Trace("Infos Verifie : " + RC + ...  
    "APE : " + FEN_Prospection.infosGlobal.sActivitePrincipaleEtablissement + RC + ...  
    "ADR : " + FEN_Prospection.sAdrEtabl + RC + ...  
    "SIREN : " + FEN_Prospection.infosGlobal.siren)  
    //"MAIL : " + infosGlobal.sMail)  
  
    sAdrEtabl      est une chaîne = FEN_Prospection.sAdrEtabl  
    sInfosCommune  est une chaîne = FEN_Prospection.sInfosCommune  
    SI sAdrEtabl = "" _OU_ sInfosCommune = Null ALORS  
      FEN_Prospection.COL_adr = FEN_Prospection.sInfosCommune  
      Trace("Adresse complète manquante. Adresse commune = " + sInfosCommune)  
    FIN  
  
  FIN  
  Trace("BoolCp2(Verifie) = " + BoolCp2)  
  RENVOYER BoolCp2
```

LE TABLEAU ASSOCIATIF



The screenshot displays the WinDev 27 IDE interface. The main window is titled "Projet1_Prospexion - WINDEV 27 - [FEN_Prospexion - Clic sur BTN_Search *]". The menu bar includes "Accueil", "Projet", "GDS", "Tests automatiques", "Code", "Refactoring", "Affichage", and "Outils". The toolbar contains icons for file operations (Nouveau, Ouvrir, Enregistrer, Fermer), editing (Couper, Copier, Coller, Dupliquer), and development (Rechercher, Volets, Options, Aide). The code editor shows a script for "Boucle_Verifie()" with the following content:

```
37 Boucle_Verifie()  
38 // *****  
39  
40 // Si les codes postaux des deux Api correspondent au combo :  
41 BoolCp1 est un booléen = Boucle_Candidattheque() = Vrai  
42 BoolCp2 est un booléen = Boucle_Verifie() = Vrai  
43 SI BoolCp1 = Vrai ET BoolCp2 = Vrai ALORS  
44   // ... ALORS rafraichir le tableau avec ces valeurs  
45   TableAffiche(TABLE_Résultats, taCourantBandeau)  
46   //infosGlobal.metierLibelle  
47   //infosGlobal.nomEntreprise  
48   FIN  
49   // Faire pareil pour les noms entreprise des deux api, ... comparer et booléen ..  
50   // faire correspondre combo_departement avec candnumdep  
51   tabAssoD2api est un tableau associatif (ccSansAccent + ccSansCasse + ccSansEspace + ccSansPonctuationNiEspace + SansDoublon) de chaînes  
52  
53   tabAssoD2api["dep"] = infosGlobal.departement  
54   tabAssoD2api["metier"] = infosGlobal.metierLibelle  
55   tabAssoD2api["nomEntr"] = infosGlobal.nomEntreprise  
56   tabAssoD2api["adr"] = ADRConcat  
57   tabAssoD2api["ape"] = infosGlobal.sActivitePrincipaleEtablissement  
58   tabAssoD2api["siren"] = infosGlobal.siren  
59   //tabAssoD2api["mail"] = infosGlobal.sMail  
60  
61   POUR TOUT infos DE tabAssoD2api  
62     // *** comparaison adresses : *** //  
63   SI Contient(sAdrEtabl, comboNumDep, ccSansAccent + ccSansCasse + ccSansEspace + ccSansPonctuationNiEspace + ccSansEspaceIntérieur ) ET ...
```

Below the code editor, the "Code" window is open, showing a list of variables and their types. The variables are organized into groups: "Général", "Membres", "Propriétés", and "Procédures". The "Membres" group is currently selected, showing a list of variables including "AdrConcat", "AuthToken", "BoolAdr", "BoolNomEntr", "CandCodeDepUrl", "CandNumDep", "CandUrl", "CodeApeVery", "CodeDep", "EriInfoApi", "InfoApe", "InfoGlobal", "InfoNomDep", "OffresPe", "Query1", "Query2", "sAdrEtabl", "sInfoCommune", "tabApe", "tabInfoGlobal", "TraitementOk", "UriBase", "UriQuery", "VerifyCpEtabl", "VerifyEtabl", "VerifyUri", "VerifyNumDep", "InfosD2Api", and "stauth".

CONDITIONS PAR BOOLEÉNS

```
POUR TOUT infos DE tabAssoD2api
// *** Comparaison adresses : *** //
SI Contient(sAdrEtabl, comboNumDep, ccSansAccent + ccSansCasse + ccSansEspace + ccSansPonctuationNiEspace + ccSansEspaceIntérieur ) ET ...
Contient(sAdrEtabl, comboNomDep, ccSansAccent + ccSansCasse + ccSansEspace + ccSansPonctuationNiEspace + ccSansEspaceIntérieur ) ALORS
tabAssoD2api.Insère("adr", sAdrEtabl)
BoolAdr = True

SINON
SI sAdrEtabl = "" ET Contient(sInfosCommune, comboNumDep, SansCasse) ALORS
tabAssoD2api.Insère("adr", sInfosCommune)
BoolAdr2 = True

SINON
SI sInfosCommune = "" ALORS
COL_adr = RougeClair = "Donnée manquante de l'Api"
BoolAdr2 = False
ErrInfoApi = True

FIN

FIN

// *** Comparaison nom entreprise : *** //
// Si nomEntreprise existe dans Candidattheque et que la chaine est contenue dans Enseigne de VÉryfie alors ...
nomEntr est une chaîne

SI tabAssoD2api["nomEntr"]..Existe = Vrai ET Contient(InfosGlobal.sEnseigne1Etablissement, InfosGlobal.nomEntreprise, SansCasse) ALORS
nomEntr = InfosGlobal.nomEntreprise
tabAssoD2api.Insère("nomEntr", InfosGlobal.nomEntreprise)
BoolNomEntr = True

SINON
SI tabAssoD2api["nomEntr"]..Existe = Faux ALORS
nomEntr = InfosGlobal.sEnseigne1Etablissement
BoolNomEntr2 = True

SINON
SI InfosGlobal.sEnseigne1Etablissement = "" ALORS
nomEntr = InfosGlobal.sEnseigne2Etablissement
BoolNomEntr2 = True

SINON
SI InfosGlobal.sEnseigne2Etablissement = "" ALORS
nomEntr = InfosGlobal.sEnseigne3Etablissement
BoolNomEntr2 = True
Trace(BoolNomEntr)

SINON
SI nomEntr = "" ALORS
COL_nomEntreprise = RougeClair = "Donnée manquante de l'Api"
BoolNomEntr2 = False
ErrInfoApi = True

FIN

FIN

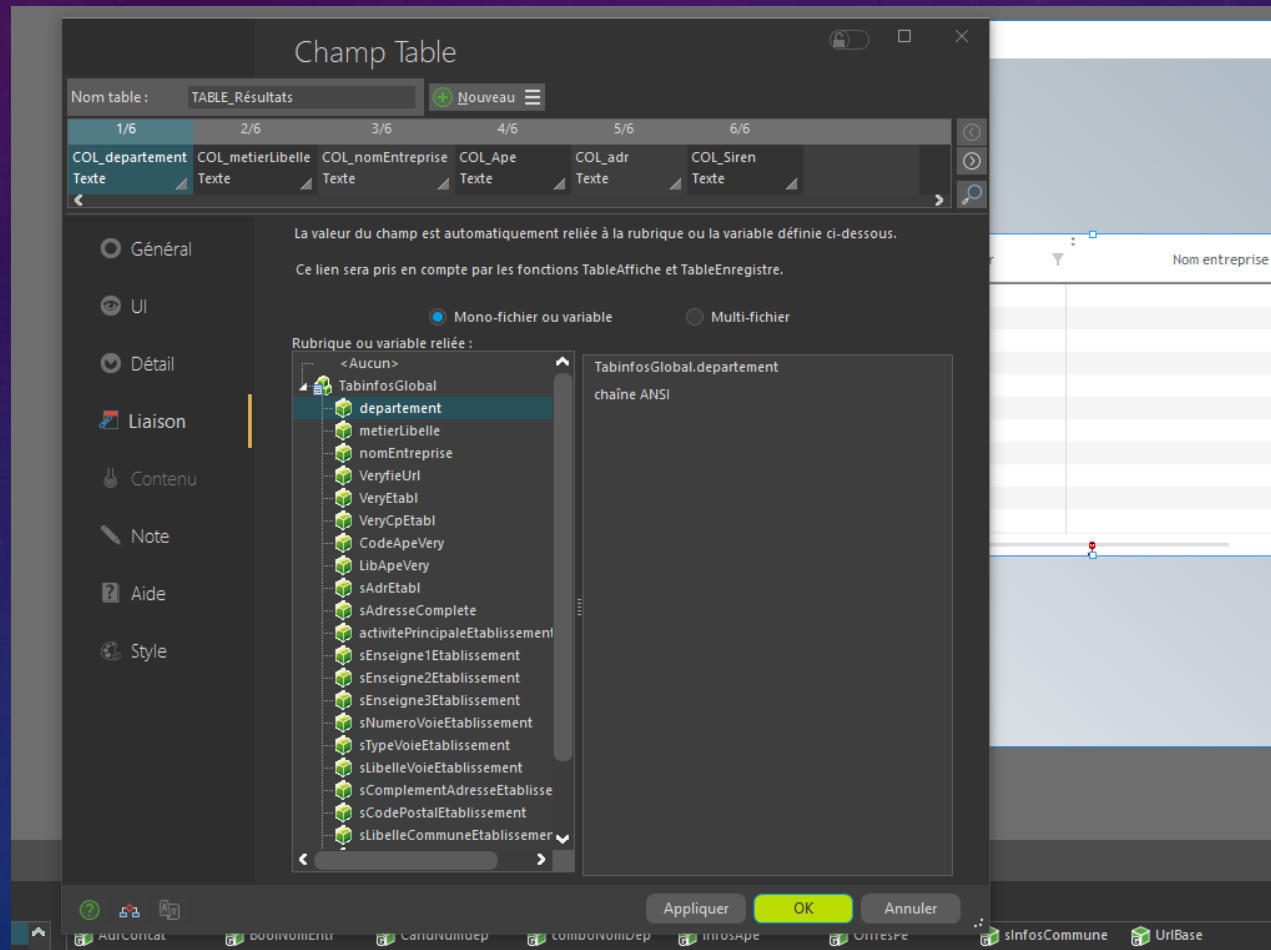
// Si l'une ou l'autre adresse ET l'un ou l'autre nom d'entreprise à été trouvé ALORS ...
SI (BoolAdr = True OU BoolAdr2 = True) ET (BoolNomEntr = True OU BoolNomEntr2 = True) ALORS
TraitementOk = True
// !!!! Problème avec BoolAdr = Pas de synchro avec le combo département !!!

FIN

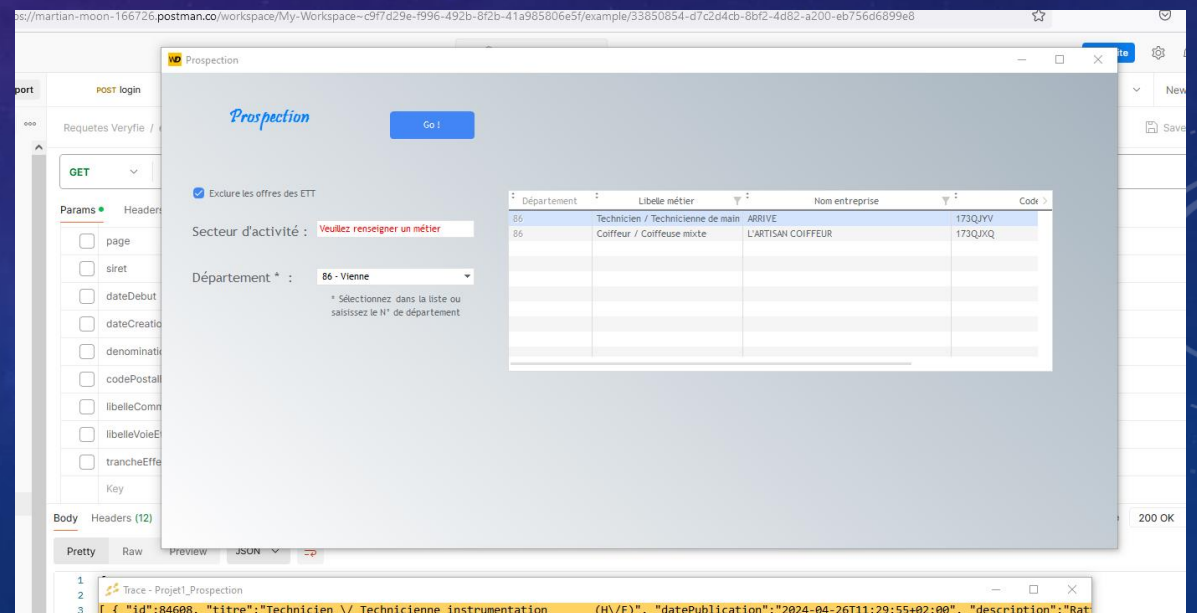
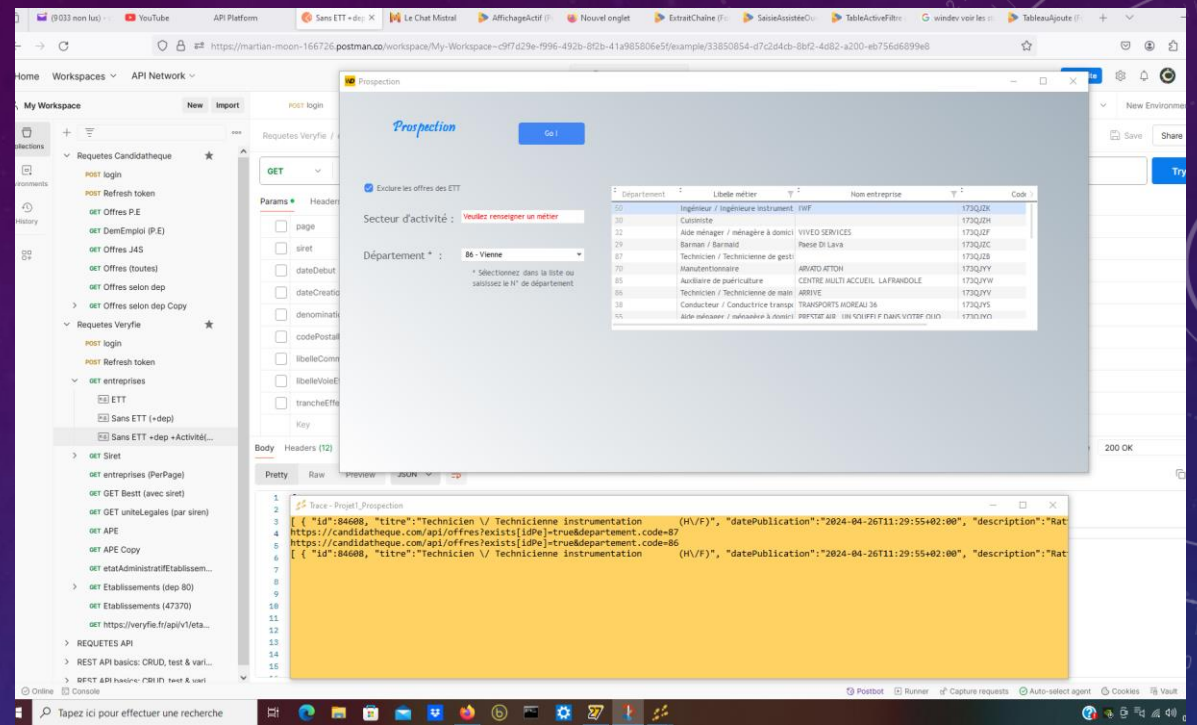
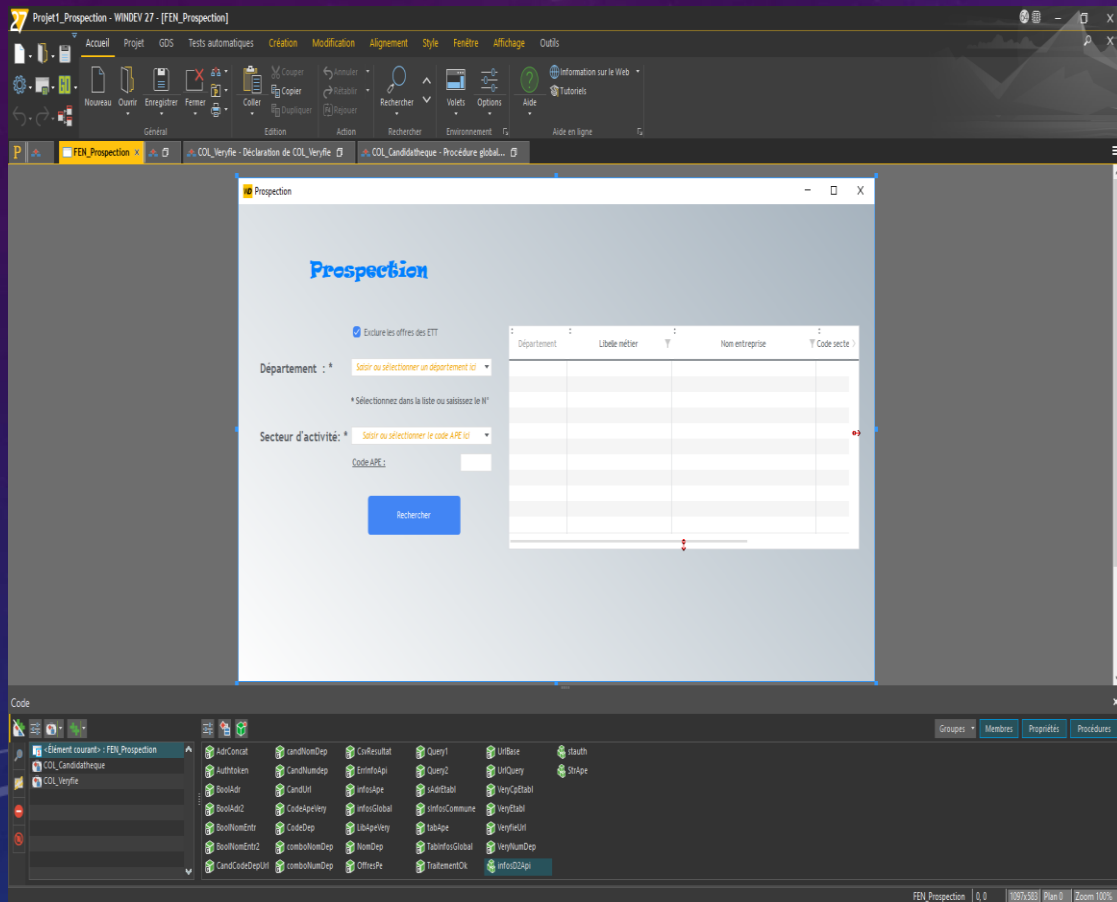
FIN
TableauAjoute(TabInfosGlobal, InfosGlobal)

FIN
```

TABLE ET LIAISON

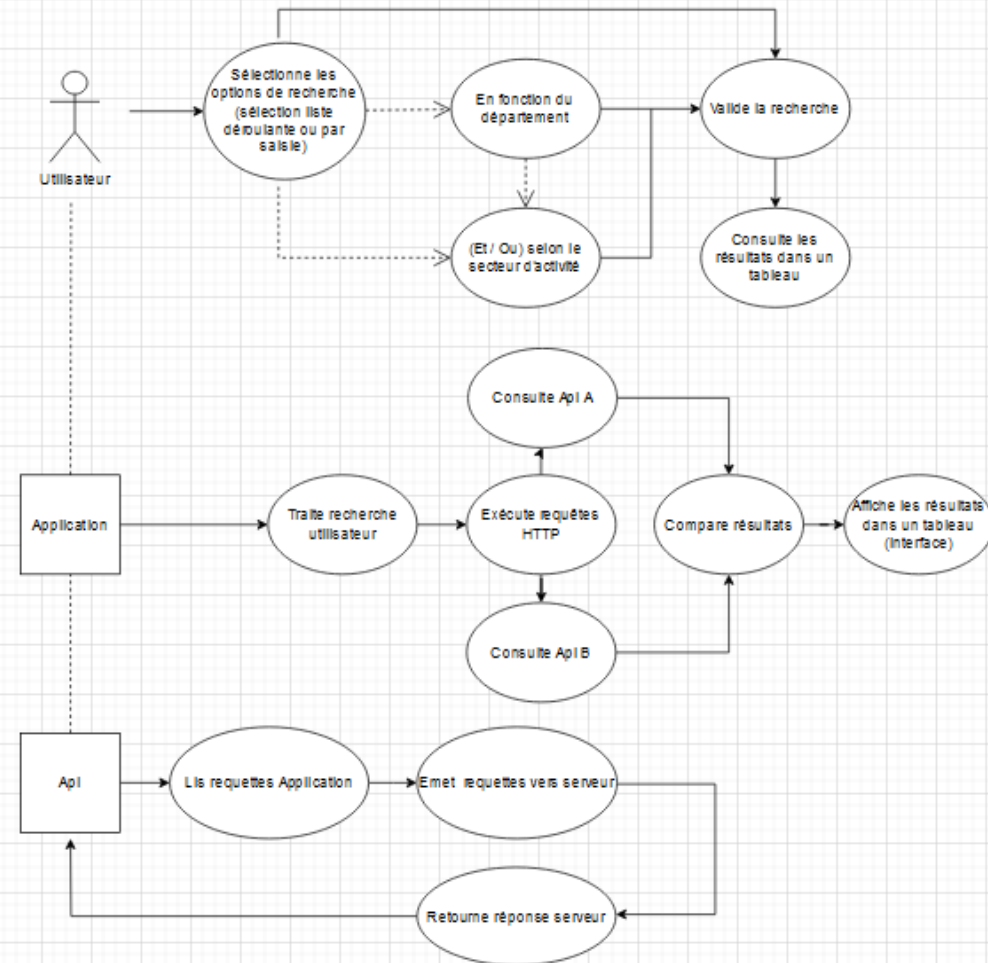


IMAGES PROJET



DIAGRAMMES PROJETS

Cas d'utilisation de l'application



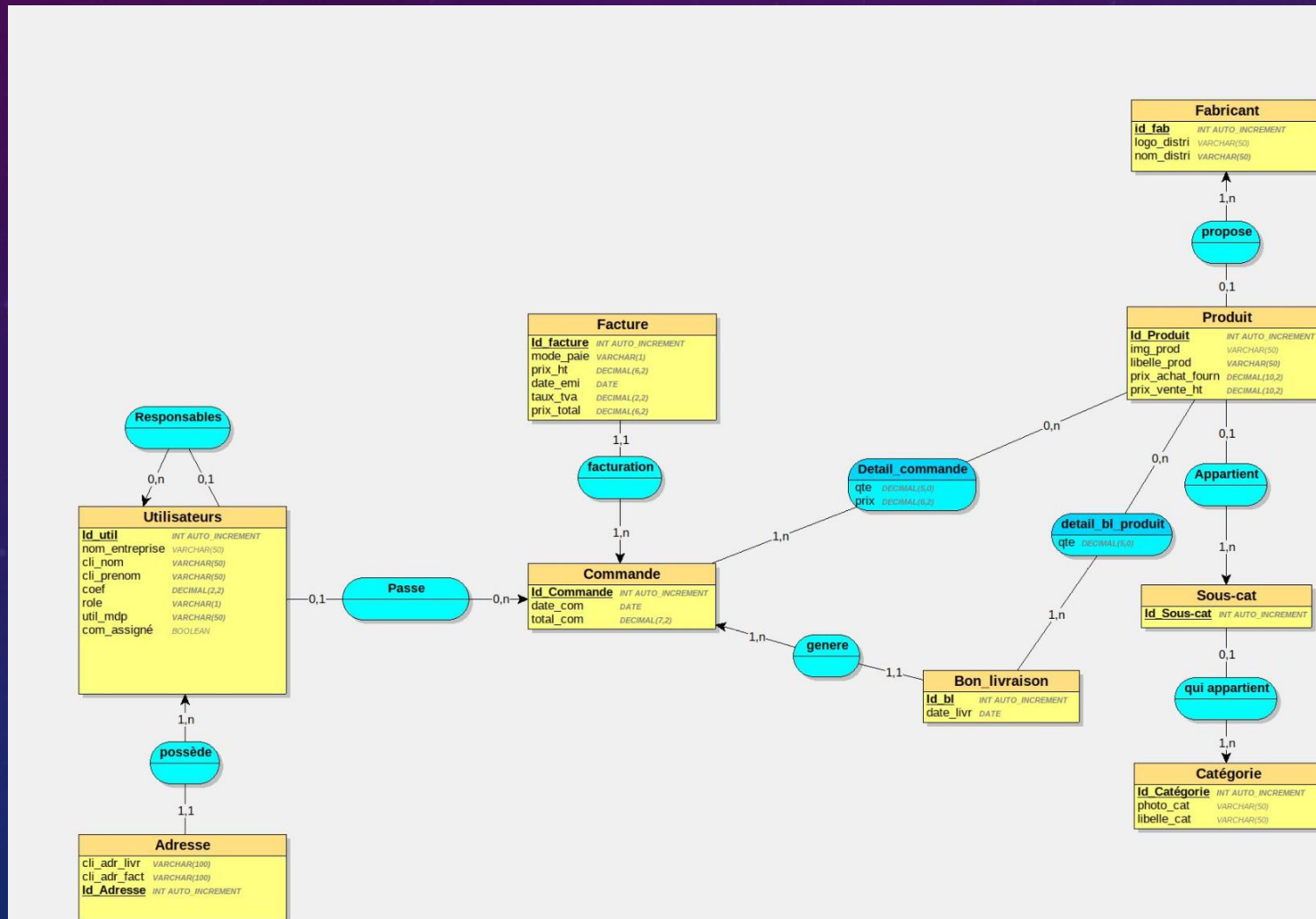
Flot nominal alternatif

Côté utilisateur : Mauvaise saisie : La recherche ne s'exécutera pas tout en affichant un message d'erreur dans le champ et suggérant un élément de recherche pertinent avec la dernière entrée de l'utilisateur.

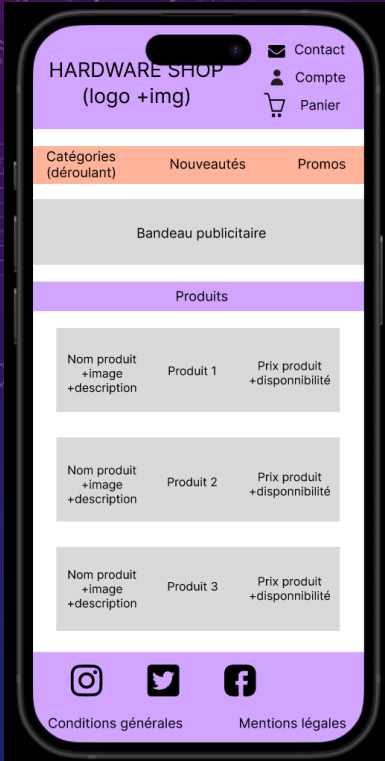
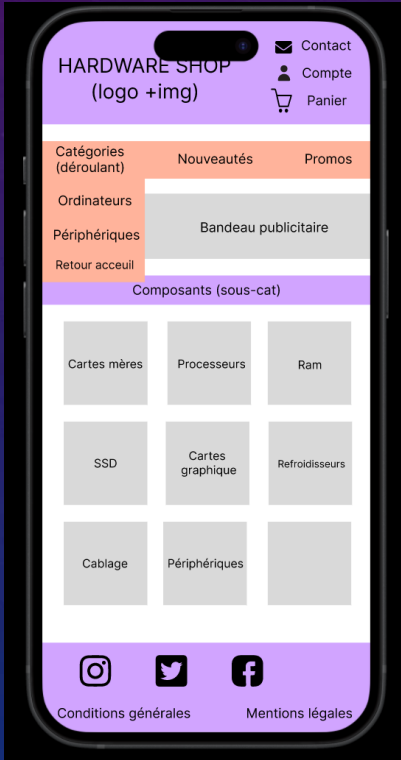
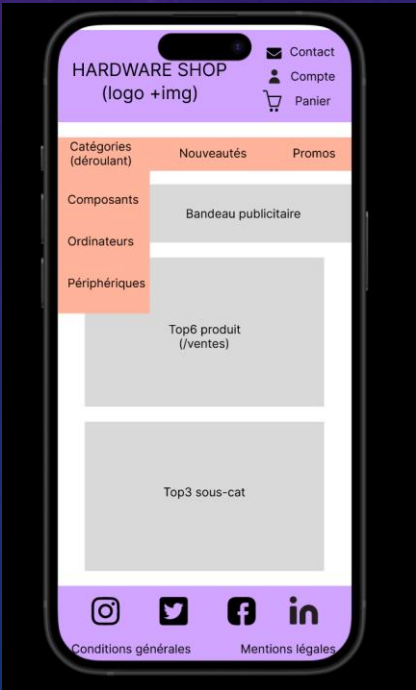
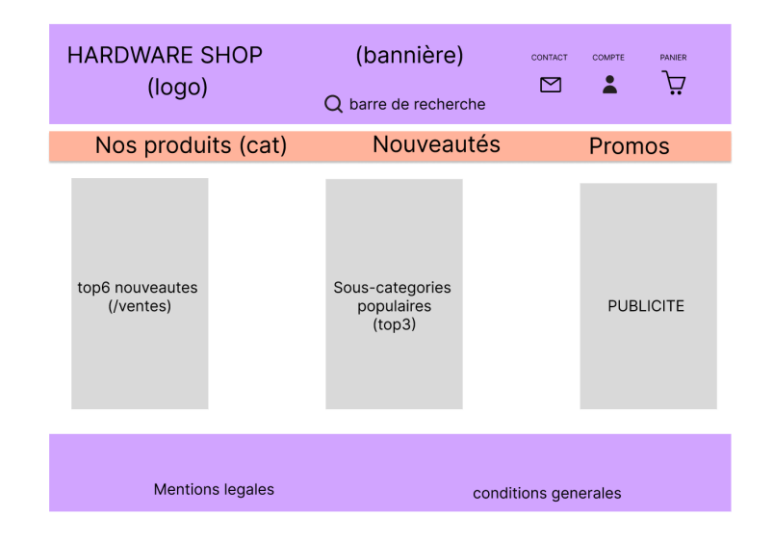
Côté application : Informations de l'Api absentes : S'il s'agit d'une adresse ou d'un nom d'entreprise, l'application tentera de compléter les informations manquantes d'une Api à l'autre et ainsi afficher les résultats.

S'il ne trouve rien, l'application affichera alors un message dans les colonnes concernés du tableau d'affichage.

MODÈLE CONCEPTUEL DES DONNÉES (PROJET FIL-ROUGE)



FIGMA (PROJET FIL-ROUGE)



MERCI POUR VOTRE ATTENTION