Florent BERNIER – BTS SIO SLAM

Cahier des charge ayant pour objectif l’étude et l’accompagnement dans la réalisation de l’Atelier Professionnel n°4 –M2L.

AP4 – M2L

Documentation technique

Table of Contents

[**Expression des besoins** 3](#_Toc167142110)

[**Les rôles obligatoires** 3](#_Toc167142111)

[**Détails de la base de donnée** 3](#_Toc167142112)

[**Table utilisateurs** 3](#_Toc167142113)

[**Table category\_item** 4](#_Toc167142114)

[**Table items** 4](#_Toc167142115)

[**Table panier** 4](#_Toc167142116)

[**Ressources et code** 5](#_Toc167142117)

[**Connexion à la BDD** 5](#_Toc167142118)

[**Page API.dart** 5](#_Toc167142119)

[**API – Connexion de l’utilisateur** 5](#_Toc167142120)

[**API – Récupérer la liste des utilisateurs** 6](#_Toc167142121)

[**API – Récupérer la liste des produits** 7](#_Toc167142122)

[**API – Ajouter un utilisateur employé** 8](#_Toc167142123)

[**API – Ajouter un produit** 9](#_Toc167142124)

[A screen shot of a computer code

Description automatically generated 9](#_Toc167142125)

[**API – Supprimer un utilisateur** 10](#_Toc167142126)

[**API – Supprimer un produit** 11](#_Toc167142127)

[**API – Mettre à jour un utilisateur** 12](#_Toc167142128)

[**API – Mettre à jour un produit** 13](#_Toc167142129)

# **Expression des besoins**

Pour ce 4e Atelier Professionnel, nous allons devoir réaliser une application mobile à l’aide de Flutter et Dart..

Ce projet a pour objectif de développer nos compétences à travailler sur un langage différent et de s’organiser.

L’application mobile permettra aux employés (staff et administrateurs) d’accéder à la liste des produits et des utilisateurs, dans laquelle il va pouvoir : supprimer, modifier et ajouter un produit, et ajouter, modifier et supprimer un utilisateur.

## **Les rôles obligatoires**

* **Administrateur (2)** : Il aura la possibilité d’ajouter, modifier ou supprimer un utilisateur, un staff, un produit, et a l’accès à la base de donnée.
* **Staff (1)** : Il aura la possibilité d’ajouter, modifier ou supprimer un produit, et a la possibilité de contrôler les utilisateurs ou staff.
* **Utilisateur (0)** : Il a aucun accès à l’application.

# **Détails de la base de donnée**

## **Table utilisateurs**

La table « users » est utilisé pour stocké les informations de l’utilisateur :

* L’ID de l’utilisateur, mis en « AUTO\_INCREMENT » qui augmentera à chaque nouvel utilisateur ajouté.
* Le prénom et nom de famille, qui sont le prénom et le nom de l’utilisateur, mis par défaut sur « NULL », n’étant pas obligatoire pour l’utilisateur.
* Le pseudonyme et l’email, étant tout deux utilisés pour la connexion à son compte utilisateur, et son unique évitant la création de plusieurs comptes sur la même adresse mail, et l’utilisation de plusieurs fois le même nom d’utilisateur.
* Le mot de passe est le texte caché, permettant à l’utilisateur de se connecter à son compte, devant être haché pour éviter l’usurpation d’identité.
* Le pays, permettant à l’utilisateur d’indiquer son pays résident, qui sera majoritairement « France » dans le cas du projet.
* Le niveau de permission, qui par défaut est défini à 0. Ce paramètre indique le pouvoir que le compte utilisateur possède :
  + Niveau 0 : Utilisateur
  + Niveau 1 : Staff
  + Niveau 2 : Administrateur

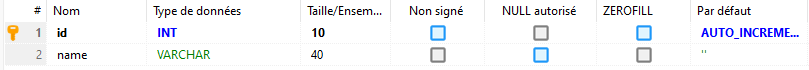
A screenshot of a computer

Description automatically generated

### **Table category\_item**

La table « category\_item » permet de stocker les différentes catégories d’item :

* L’ID, mis en « AUTO\_INCREMENT », qui définit l’id de la catégorie.
* Le nom de la catégorie (par exemple : ballon, vêtement, équipements, … etc.).



### **Table items**

La table « items » permet de stocker les informations d’un article :

* L’ID, mis en « AUTO\_INCREMENT », qui définit l’id de l’article.
* Le nom de l’article, qui sera affiché sur le site.
* La description de l’article, affichée sur le site donnant des informations sur l’article.
* Les stocks, permettant d’afficher le nombre d’article restant.
* L’image de l’article, affichant à quoi ressemble l’article.
* Le prix de l’article, indiquant à l’utilisateur le prix de l’article.

A screenshot of a computer

Description automatically generated

### **Table panier**

La table « cart » permet d’afficher le panier d’un utilisateur :

* L’ID.
* Le nom du produit.
* La quantité d’article ajoutés.
* Le prix total des articles dans le panier.
* L’id\_item, faisant référence à l’article ajouté dans le panier.
* L’id\_user, étant l’identifiant de l’utilisateur ayant ajouté le produit.

A screenshot of a computer

Description automatically generated

# **Ressources et code**

## **Connexion à la BDD**

Pour recevoir les informations de la base de donnée sur l’application, nous passons par le lien du serveur où nous pourrons effectuer les requêtes déjà préparées.

## **Page API.dart**

La page API.dart est la page où les requêtes de l’application sont effectuées. On récupère sur cette page via l’url du serveur, les routes pour l’application mobile.

### **API – Connexion de l’utilisateur**

A screen shot of a computer code

Description automatically generated

### **API – Récupérer la liste des utilisateurs**

A computer screen shot of a program code

Description automatically generated

### **API – Récupérer la liste des produits**

A screen shot of a computer code

Description automatically generated

### **API – Ajouter un utilisateur employé**

A computer screen shot of a program code

Description automatically generated

### **API – Ajouter un produit**

### A screen shot of a computer code Description automatically generated

### **API – Supprimer un utilisateur**

A computer screen shot of a program code

Description automatically generated

### **API – Supprimer un produit**

A computer screen shot of a program code

Description automatically generated

### **API – Mettre à jour un utilisateur**

A screen shot of a computer program

Description automatically generated

### **API – Mettre à jour un produit**

A screen shot of a computer program

Description automatically generated