



# PPE3 DOCUMENTATION ET MANUEL D'UTILISATION

Application Java

Gestion de vente(s),  
Gestion de produit(s),  
Gestion de client(s),  
Gestion d'utilisateur(s).

# **Sommaire Documentation et Manuel d'utilisation**

Introduction .....	2
<i>Cahier des charges</i> .....	2
Technique Base de Données .....	3
<i>Modèle Conceptuel Des Données</i> .....	3
<i>Modèle Logique Des Données</i> .....	3
<i>Script SQL</i> .....	4
Technique Programmation .....	6
<i>Création du projet et de la première fenêtre d'authentification</i> .....	6
<i>Création classe technique de connexion et développement du formulaire</i> .....	7
<i>Fenêtre de l'Agent</i> .....	10
<i>Fenêtre Administrateur</i> .....	12
Manuel d'Utilisation.....	15
<i>Comment utiliser le logiciel en tant que agent</i> .....	15
<i>Comment utiliser le logiciel en tant que Administrateur</i> .....	22
Amélioration envisageable .....	26

# Introduction

## Cahier des charges

L'objectif est de créer une application développée en JAVA permettant l'authentification d'un « agent » et / ou d'un « administrateur ».

L'agent pourra effectuer une fois authentifier :

- Afficher les différentes catégories correspondant au produit (Nous allons vendre des jeux vidéo)
- Pouvoir gérer une vente d'un ou plusieurs produits à un client
- Pouvoir gérer tous les clients enregistrés si déjà existant, ou alors procéder à la création d'un nouveau client
- Générer un bon de commande à l'écran puis générer une facture au client

L'administrateur pourra gérer une fois authentifier :

- Les profils (des agents ou des administrateurs)
- Les différents agents
- Les produits ainsi que les catégories
- Les ventes effectuées par un agent à un client
- Afficher les statistiques de ventes (Chiffre d'affaire)
- Gestion de stock de produit

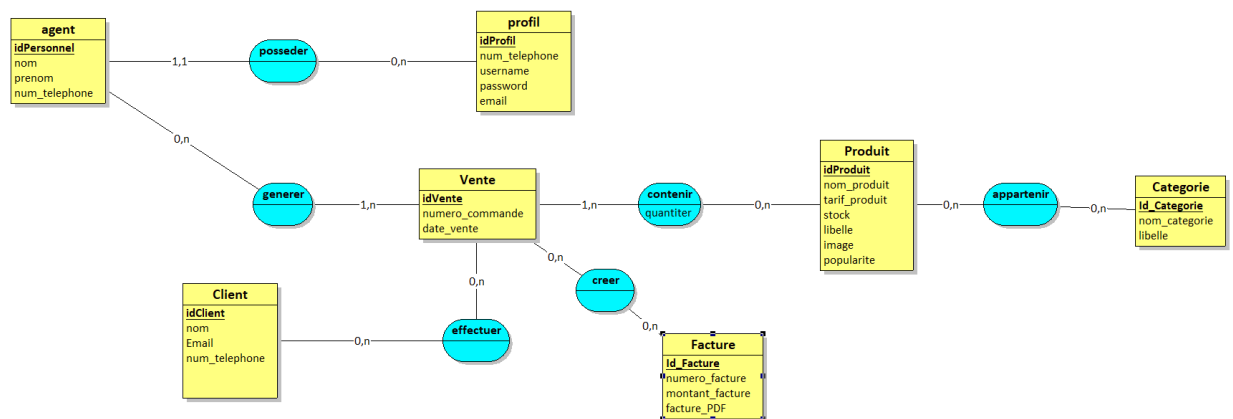
**Annotation :** Une base de données doit donc être conçu héberger dans un serveur Docker sous linux.

# Technique Base de Données

## Modèle Conceptuel Des Données

Pour réaliser une telle application nous allons avoir besoin de réaliser une base de données pour stocker diverses informations, tels que les clients, les agents, les factures, les dates etc...

Nous allons modéliser notre Modèle Conceptuel Des Données (MCD) à l'aide d'un outil de conception. (Vous pouvez télécharger le MDC dans l'archive .ZIP de même pour le script SQL)



## Modèle Logique Des Données

De plus nous pouvons réaliser le Modèle Logique Conceptuel qui correspond à ce schéma :

Client = (idClient, nom, email, num\_telephone)

Produit = (idProduit, nom\_produit, tarif\_produit, stock, libelle, image, popularite)

Vente = (idVente, numero\_commande, date\_vente)

Profil = (idProfil, num\_telephone, username, password, email)

Agent = (idAgent, nom, prenom, num\_telephone, #idProfil)

Categorie = (id\_Categorie, nom\_categorie, libelle)

Facture = (id\_Facture, numero\_facture, montant\_facture, facture\_PDF)

Generer = (#idVente, #idAgent)

Contenir = (#idProduit, #idVente, quantiter)

Effectuer = (#idClient, #idVente)

Appartenir = (#idProduit, #id\_Categorie)

Créer = (#idVente, #id\_Facture)

## Script SQL

Après avoir une vision claire et précise de notre base de données nous pouvons concevoir le script SQL qui va permettre de créer la base de données ainsi que les tables correspondantes.

```
CREATE TABLE Client(  
    idClient VARCHAR(50),  
    nom VARCHAR(50),  
    Email VARCHAR(50),  
    num_telephone BYTE,  
    PRIMARY KEY(idClient)  
);
```

```
CREATE TABLE Produit(  
    idProduit INT,  
    nom_produit VARCHAR(50),  
    tarif_produit CURRENCY,  
    stock INT,  
    libelle VARCHAR(50),  
    image VARCHAR(50),  
    popularite INT,  
    PRIMARY KEY(idProduit)  
);
```

```
CREATE TABLE Vente(  
    idVente INT,  
    numero_commande BIGINT,  
    date_vente DATETIME,  
    PRIMARY KEY(idVente)  
);
```

```
CREATE TABLE profil(  
    idProfil INT,  
    num_telephone VARCHAR(50),  
    username VARCHAR(50),  
    password VARCHAR(50),  
    email VARCHAR(50),  
    PRIMARY KEY(idProfil)  
);
```

```
CREATE TABLE agent(  
    idPersonnel INT,  
    nom VARCHAR(50),  
    prenom VARCHAR(50),  
    num_telephone VARCHAR(50),  
    idProfil INT NOT NULL,  
    PRIMARY KEY(idPersonnel),  
    FOREIGN KEY(idProfil) REFERENCES profil(idProfil)  
);
```

```
CREATE TABLE Categorie(  
    Id_Categorie COUNTER,  
    nom_categorie VARCHAR(50),  
    libelle VARCHAR(50),  
    PRIMARY KEY(Id_Categorie)  
);
```

```
CREATE TABLE Facture(  
    Id_Facture COUNTER,  
    numero_facture VARCHAR(50),  
    montant_facture INT,  
    facture_PDF BINARY(50),  
    PRIMARY KEY(Id_Facture)  
);
```

```

CREATE TABLE generer(
    idVente INT,
    idPersonnel INT,
    PRIMARY KEY(idVente, idPersonnel),
    FOREIGN KEY(idVente) REFERENCES Vente(idVente),
    FOREIGN KEY(idPersonnel) REFERENCES agent(idPersonnel)
);

CREATE TABLE contenir(
    idProduit INT,
    idVente INT,
    quantiter INT,
    PRIMARY KEY(idProduit, idVente),
    FOREIGN KEY(idProduit) REFERENCES Produit(idProduit),
    FOREIGN KEY(idVente) REFERENCES Vente(idVente)
);

CREATE TABLE effectuer(
    idClient VARCHAR(50),
    idVente INT,
    PRIMARY KEY(idClient, idVente),
    FOREIGN KEY(idClient) REFERENCES Client(idClient),
    FOREIGN KEY(idVente) REFERENCES Vente(idVente)
);

CREATE TABLE appartenir(
    idProduit INT,
    Id_Categorie INT,
    PRIMARY KEY(idProduit, Id_Categorie),
    FOREIGN KEY(idProduit) REFERENCES Produit(idProduit),
    FOREIGN KEY(Id_Categorie) REFERENCES Categorie(Id_Categorie)
);

CREATE TABLE creer(
    idVente INT,
    Id_Facture INT,
    PRIMARY KEY(idVente, Id_Facture),
    FOREIGN KEY(idVente) REFERENCES Vente(idVente),
    FOREIGN KEY(Id_Facture) REFERENCES Facture(Id_Facture)
);

```

(Voir l'archive ZIP du projet, le script SQL est disponible, la base de données a été exporté est prête à l'emploi)

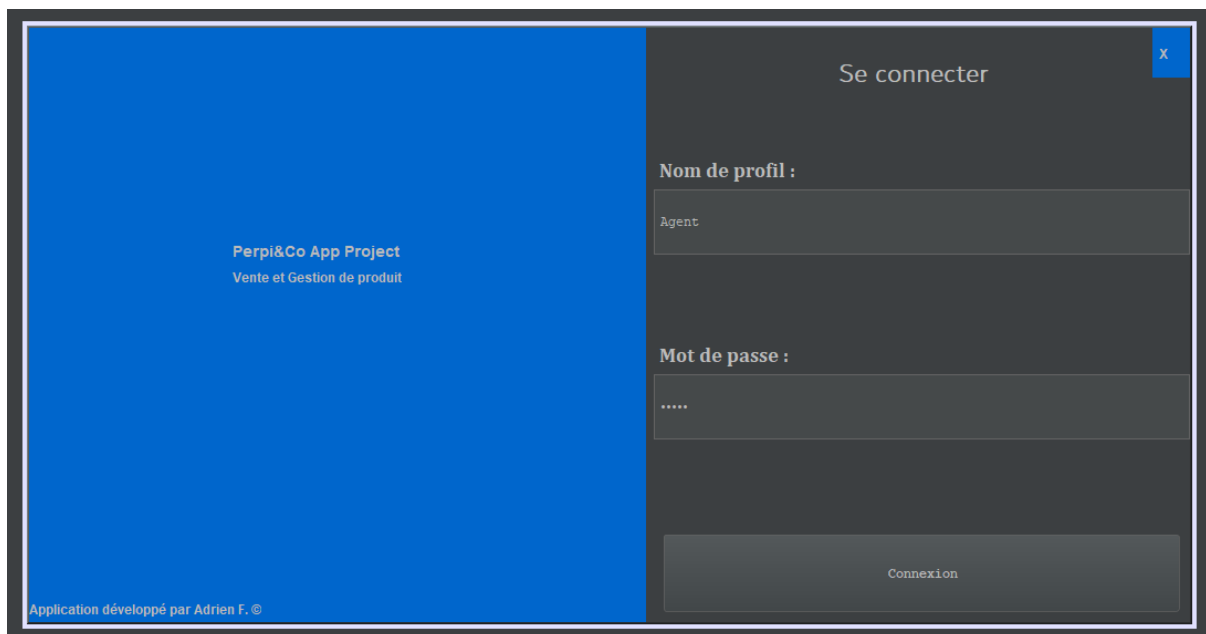
# Technique Programmation

## Création du projet et de la première fenêtre d'authentification

(\***Annotation Importante** à savoir lors de la documentation technique qui va concerner la partie programmation de l'application JAVA à l'aide de l'IDE Netbeans. Sachant que nous avons ici un très gros projet, nous allons détaillés uniquement les choses importantes, certaines méthodes, certaines fonctions, certaines classes etc... Qui pourrons être utile à connaître et appréhender pour potentiellement une future réutilisation du code par des inconnus.

Cependant une JavaDoc sera présente pour expliquer en supplément les différentes fonctions, méthodes)

Pour commencer nous allons créer un composant qui va s'appeler « Formulaire ». Voici le résultat.



Après avoir créé notre composant formulaire nous allons créer une JFrame (« JFrameMain ») dans lequel nous allons insérer notre composant « Formulaire »

L'objectif de notre formulaire et de s'authentifier via un Pseudo et un Mot de passe qui va être stocké dans notre Base de données (BDD), nous allons donc créer une classe technique qui va nous permettre d'effectuer la connexion entre notre application et notre BDD.

## Création classe technique de connexion et développement du formulaire

```
/* Classe d'accès aux données contenant des membres statiques afin de manipuler
 * la BDD ON implémente ici le Design Pattern Singleton
 *
 */
public class DaoSIO {

    /**
     * Membres static (de classe)
     * Informations concernant la connexion "Nom serveur" "port" "nomBDD" "usernamePHPmyAdmin" "passwordPHPmyAdmin"
     */
    private static String nomServeur = "localhost";
    private static String port = "3306";
    private static String nomBdd = "ppe3_logicielbdd_off";
    private static String nomUtilisateur = "root";
    private static String motDePasse = "toor";

    private static String chaineConnexion;

    //propriété non statique
    private Connection connexion;

    private static DaoSIO monDao = null;

    /**
     * Constructeur privé ! pour construire un objet, il faut utiliser la
     * méthode publique statique getDaoSIO
     *
     */
    private DaoSIO() {
        try {
            //Définition de l'emplacement de la BDD
            DaoSIO.chaineConnexion = "jdbc:mysql://" + DaoSIO.nomServeur + "/" + DaoSIO.nomBdd + "?useUnicode=true"
                + "&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&"
                + "serverTimezone=UTC";

            //Création de la connexion à la BDD
            this.connexion = (Connection) DriverManager.getConnection(DaoSIO.chaineConnexion, DaoSIO.nomUtilisateur, DaoSIO.motDePasse);

        } catch (SQLException ex) {
            Logger.getLogger(DaoSIO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

La première partie nous permet de créer la connexion puis la création du constructeur

```
/**
 * Permet d'obtenir l'objet instancié
 *
 * @return un Objet DaoSIO avec connexion active ... pour une certaine durée
 */
public static DaoSIO getInstance() {

    if (DaoSIO.monDao == null) {
        DaoSIO.monDao = new DaoSIO();
    } else {
        if (!DaoSIO.monDao.connexionActive()) {
            DaoSIO.monDao = new DaoSIO();
        }
    }

    return DaoSIO.monDao;
}
```

La seconde partie nous permet d'obtenir l'objet qui à été instancié

(Exemple d'utilisation : *DaoSIO.getInstance()*) (On instancie notre objet DaoSIO et on récupère la connexion active pour potentiellement faire des requêtes par la suite)



```

public Boolean connexionActive() {
    Boolean connexionActive = true;
    try {
        if (this.connexion.isClosed()) {
            connexionActive = false;
        }
    } catch (SQLException ex) {
        Logger.getLogger(DaoSIO.class.getName()).log(Level.SEVERE, null, ex);
    }
    return connexionActive;
}

/**
 *
 * @param sql, comportera un ordre selec
 * @return
 */
public ResultSet requeteSelection(String sql) {
    try {
        Statement requete = new DaoSIO().connexion.createStatement();
        return requete.executeQuery(sql);
    } catch (SQLException ex) {
        Logger.getLogger(DaoSIO.class.getName()).log(Level.SEVERE, null, ex);
    }
    return null;
}

/**
 *
 * @param sql, comportera un ordre insert, update, select, alter, etc.
 * @return le nombre de lignes impactées par la requête action
 */
public Integer requeteAction(String sql) {
    try {
        Statement requete = new DaoSIO().connexion.createStatement();
        return requete.executeUpdate(sql);
    } catch (SQLException ex) {
        Logger.getLogger(DaoSIO.class.getName()).log(Level.SEVERE, null, ex);
    }
    return 0;
}

```

Cette partie du code ci-dessus représente les différentes méthodes public que l'on pourra appeler si l'on souhaite effectuer des requêtes SQL en fonction de l'action (SELECT, UPDATE, DELETE) c'est-à-dire si l'on souhaite faire une sélection dans la base de données nous allons appeler la méthode `requeteSelection`(« SELECT XXXXX ») autrement si l'on veut faire une action pour supprimer un tuple on appellera `requeteAction`(« DELETE XXXXXXXX »)

(Exemple d'utilisation concret :

`DaoSIO.getInstance().requeteSelection()` // \ DaoSIO.getInstance().requeteAction()

Désormais nous pouvons coder notre formulaire (voir ci-dessous) :

```

/* Cette méthode permet l'authentification d'un utilisateur elle appelle une
 * méthode de la classe BDQql qui permet la connexion a la base de donnée
 */
private void connexion() {

    String loginUsername = jTextFieldUsernameLogging.getText();
    String loginPassword = jPasswordField.getPassword();

    //Si une connexion est active
    if (DaoSIO.getInstance() != null) {

        try {
            //On cherche le profil saisi dans le formulaire pour voir si le profil existe (Username et Password)
            ResultSet result = DaoSIO.getInstance().requeteSelection("SELECT * FROM profil WHERE username = '" + loginUsername + "' AND password = '" + loginPassword + "'");
            if (result.next()) {
                //On définit nos trois objet JFrame
                JFrameAgent JFrameAgent = new JFrameAgent();
                JFrameAdmin JFrameAdmin = new JFrameAdmin();
                JFrameMain JFrameMain = new JFrameMain();
                try {
                    //Si le profil a pour permission = 1 alors il se connecte en admin
                    ResultSet resultPermAdmin = DaoSIO.getInstance().requeteSelection("SELECT * FROM profil WHERE username = '" + loginUsername + "' AND permission = 1");
                    //Si le profil a pour permission = 0 alors il se connecte en tant que agent
                    ResultSet resultPermAgent = DaoSIO.getInstance().requeteSelection("SELECT * FROM profil WHERE username = '" + loginUsername + "' AND permission = 0");

                    if (resultPermAdmin.next()) {
                        JFrameAgent.setVisible(false);
                        JFrameAdmin.setTitle("Vous êtes authentifié en tant que Administrateur : " + loginUsername);
                        JFrameAdmin.setVisible(true); //On ouvre la fenêtre administrateur
                        JFrameMain.setVisible(false);
                    }

                    if (resultPermAgent.next()) {
                        JFrameAdmin.setVisible(false);
                        JFrameAgent.setTitle("Vous êtes authentifié en tant que Agent : " + loginUsername);
                        JFrameAgent.setVisible(true); //On ouvre la fenêtre Agent
                    }

                } catch (SQLException e) {
                    Logger.getLogger(Formulaire.class.getName()).log(Level.SEVERE, null, e);
                }

                jTextFieldUsernameLogging.setEnabled(false);
                jPasswordField.setEnabled(false);
                jButtonLogging.setEnabled(false);
            }
        }

        System.out.println("Authentification réussie, utilisateur trouvée");

    } else {

        Object[] options = {"OK"};
        System.out.println("Authentification Failed not user found");
        JOptionPane.showOptionDialog(null, "Connexion échouée, utilisateur non trouvé !", "Erreur de connexion",
            JOptionPane.DEFAULT_OPTION, JOptionPane.WARNING_MESSAGE,
            null, options, options[0]);

    }

} catch (SQLException ex) {
    Logger.getLogger(Formulaire.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

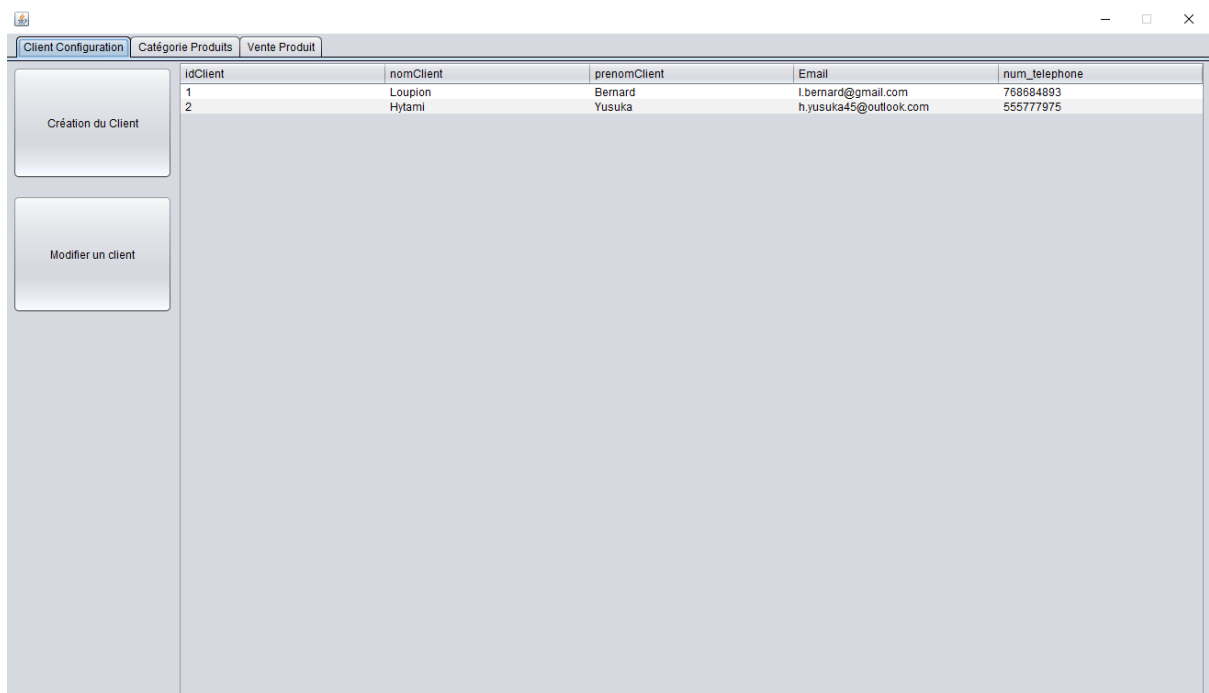
Le code permet l'authentification grâce à l'utilisation de notre classe technique, et de nos requêtes SQL. Cette partie du code permet de se connecter en tant qu'Agent et d'ouvrir la fenêtre spécifique à l'Agent (JFrameAgent), et de se connecter en tant qu'Administration et d'ouvrir la fenêtre spécifique à l'Administrateur (JFrameAdmin)

## Fenêtre de l'Agent

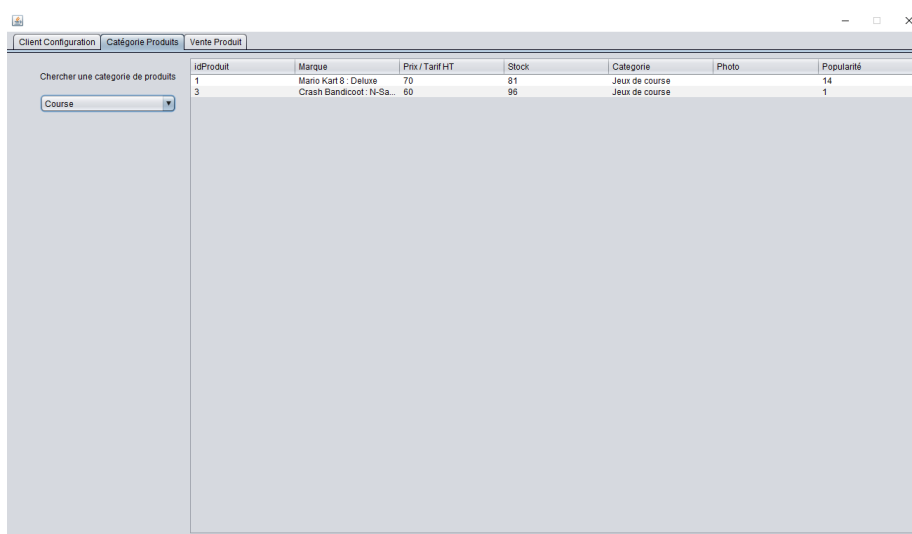
Création d'une JFrame (JFrameAgent)

Implémentation d'un composant de type JTabbedPaneOnglet qui va nous permettre d'avoir plusieurs onglets (Premier Onglet → la gestion des clients, Second Onglet → La gestion des produits par catégories, Troisième Onglet → Les ventes à un clients (Effectuer des ventes à un client))

Premier Onglet (La gestion des clients) :



Le second Onglet (La gestion des produits par catégories) :



Le troisième onglet (La gestion des ventes) :

The screenshot shows a software window titled 'Vente Produit' with three tabs: 'Client Configuration', 'Catégorie Produits', and 'Vente Produit'. The 'Vente Produit' tab is active. It contains a table titled 'Liste des produits disponibles à la vente :'. The table has columns: idProduit, nom produit, tarif produit, stock disponible, description pr..., photo produit, and Popularité / Ev... The table lists 5 products. Below the table is a large empty rectangular area. To the right of the table are three empty rectangular boxes labeled 'Produit(s)', 'Prix', and 'Quantité'. At the bottom of the window, there are two buttons: 'Ajouter les produits sélectionner au panier' and 'Effectuer la vente'. To the right of these buttons are two input fields, one with the value '0' and a label 'Montant HT'.

idProduit	nom produit	tarif produit	stock disponible	description pr...	photo produit	Popularité / Ev...
1	Mario Kart 8 : ...	70	81	Jeux de course		14
2	Mario Tennis : ...	70	74	Jeux de sport		15
3	Crash Bandic...	60	96	Jeux de course		1
4	Mario et Sonic...	60	95	Jeux de sport		0
5	Animal Cressi...	80	194	Jeux Bac à Sa...		2

Les différentes méthodes utilisées (Vous pouvez vous référer à la JavaDoc de l'application pour en savoir plus) :

*filtreCategorieComboBox()* → Permet de filtrer les produits par catégories

*numeroVenteAleatoire()* → Permet de générer un numéro (Chiffre et lettre) aléatoire et unique

*dateVente()* → Permet d'afficher la date à laquelle à été effectuer une vente par exemple

*infoFacturationApresVente()* → Permet d'afficher un bon de commande à l'écran

*generationPDF()* → génère une Facture sous formes PDF dans un dossier créer si il n'existe pas par l'application dans (« C:\\PPE3-facture »)

*traitementVente()* → Validation d'une vente (Génération facture etc...)

*InsertionClient()* → Permet d'ajouter un client

*ajoutPanierTraitement* → Permet l'ajout au panier d'un produit

*afficherProduitVente()* → Affiche au tableau java tous les produits au niveau de l'onglet vente

*afficherAgent()* → Affiche au tableau java tous les agents au niveau de l'onglet agent

*afficherClient()* → Affiche au tableau java tous les clients au niveau de l'onglet client

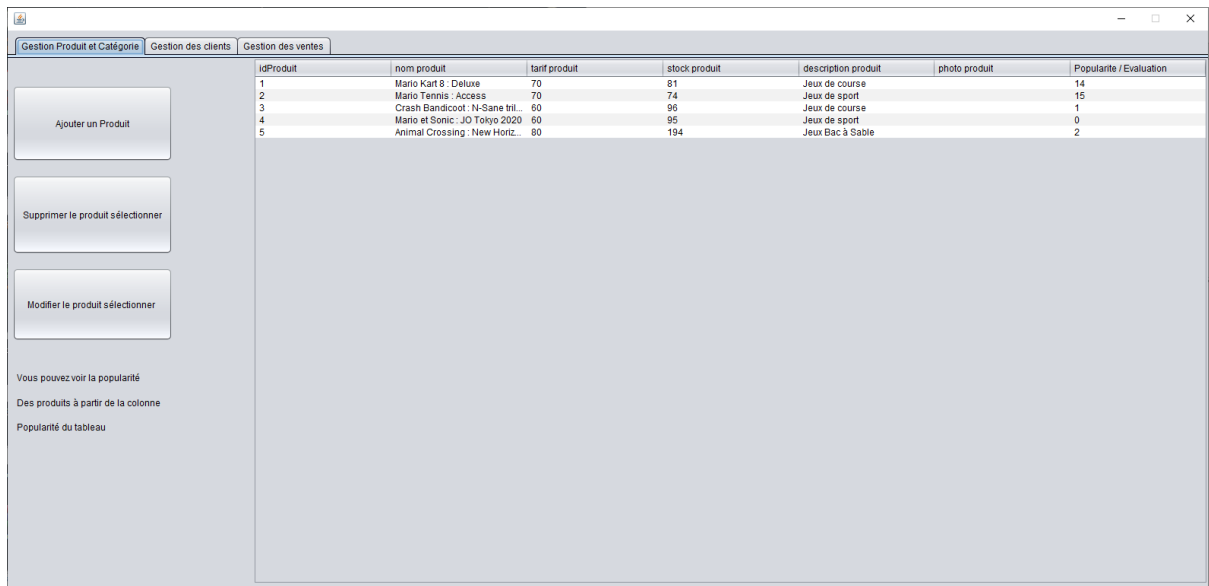
*afficherCatégorieDeProduit()* → Affiche au tableau java tous les produits et filtre les produits par catégories au niveau de l'onglet catégorie produit

## Fenêtre Administrateur

Création d'une JFrame (JFrameAdmin)

Implémentation d'un composant de type JTabbedPaneOnglet qui va nous permettre d'avoir plusieurs onglets (Premier Onglet → Gestion des produits, Second Onglet → Gestion des agents, Troisième Onglet → Gestion des ventes, Quatrième Onglet → Gestion des clients)

Premier Onglet :



idProduit	nom produit	tarif produit	stock produit	description produit	photo produit	Popularité / Evaluation
1	Mario Kart 8 : Deluxe	70	81	Jeux de course		14
2	Mario Tennis : Access	70	74	Jeux de sport		15
3	Crash Bandicoot : N-Sane tril...	60	96	Jeux de course		1
4	Mario et Sonic : JO Tokyo 2020	60	95	Jeux de sport		0
5	Animal Crossing : New Horiz...	80	194	Jeux Bac à Sable		2

Ajouter un Produit

Supprimer le produit sélectionné

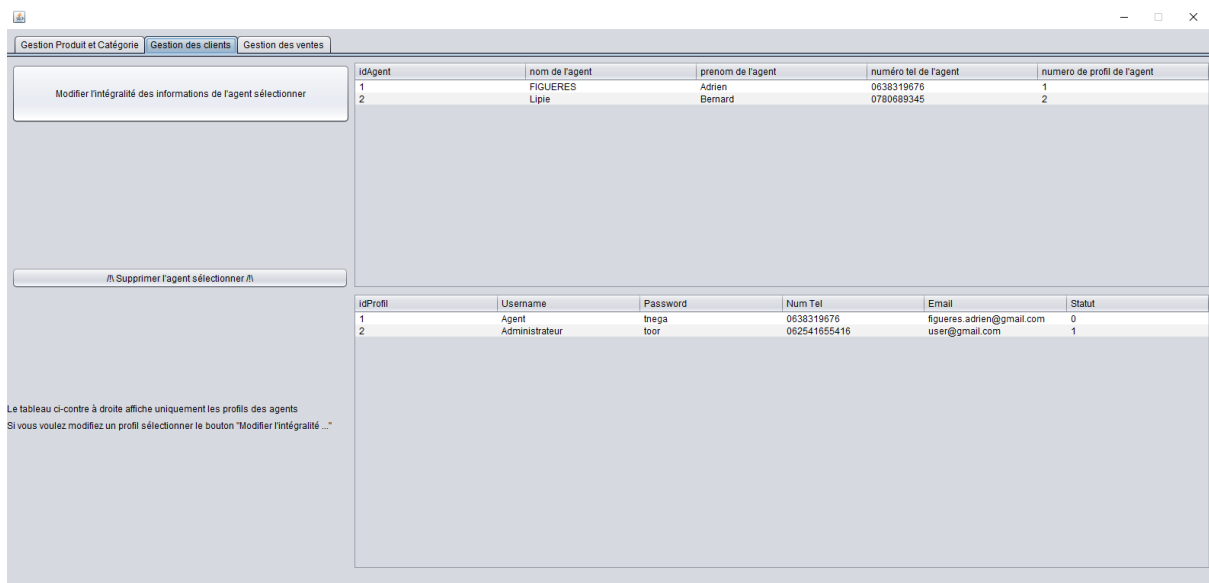
Modifier le produit sélectionné

Vous pouvez voir la popularité

Des produits à partir de la colonne

Popularité du tableau

Second Onglet :



idAgent	nom de l'agent	prenom de l'agent	numéro tel de l'agent	numero de profil de l'agent
1	FIGUERES	Adrien	0638319676	1
2	Lipie	Bernard	0780689345	2

Modifier l'intégralité des informations de l'agent sélectionné

Supprimer l'agent sélectionné

idProfil	Username	Password	Num Tel	Email	Statut
1	Agent	thega	0638319676	figueres.adrien@gmail.com	0
2	Administrateur	toor	062541655416	user@gmail.com	1

Le tableau ci-contre à droite affiche uniquement les profils des agents  
Si vous voulez modifier un profil sélectionnez le bouton "Modifier l'intégralité ..."

### Troisième Onglet :

Ajouter un Produit

Supprimer le produit sélectionner

Modifier le produit sélectionner

Vous pouvez voir la popularité

Des produits à partir de la colonne

Popularité du tableau

IdProduit	nom produit	tarif produit	stock produit	description produit	photo produit	Popularite / Evaluation
1	Mario Kart 8 : Deluxe	70	81	Jeux de course		14
2	Mario Tennis : Access	70	74	Jeux de sport		15
3	Crash Bandicoot: N-Sane tril...	60	96	Jeux de course		1
4	Mario et Sonic : JO Tokyo 2020	60	95	Jeux de sport		0
5	Animal Crossing: New Horiz...	80	194	Jeux Bac à Sable		2

### Quatrième Onglet :

Création d'un client

Modifier le client sélectionner

Supprimer le client sélectionner

IdClient	nomClient	prenomClient	Email	num_telephone
1	Loupion	Bernard	l.bernard@gmail.com	768684893
2	Hylami	Yusuka	h.yusuka45@outlook.com	555777975

*Les différentes méthodes utilisées (Vous pouvez vous référer à la JavaDoc de l'application pour en savoir plus) :*

`afficherProduit()` → Affiche tous les produits dans la table appartenant à l'onglet gestion des produits

`afficherAgent()` → Affiche tous les agents dans la première table appartenant à l'onglet gestion des agents

`afficherProfil()` → Affiche tous les profils dans la seconde table appartenant à l'onglet gestion des agents

`afficherVente()` → Affiche toutes les ventes dans la table appartenant à l'onglet gestion des ventes

`recupProfil()` → Permet de récupérer le profil de l'agent en question que l'on souhaite modifier

`afficherClient()` → Affiche tous les clients dans le tableau appartenant à l'onglet Gestion des clients

# Manuel d'Utilisation

## Comment utiliser le logiciel en tant que agent

Vous êtes un membre du personnel ? un commercial ? un agent ? vous ne savez pas comment utiliser l'application ? voici une notice d'utilisation qui vous permettra de comprendre comment fonctionne le logiciel.

Premièrement, lorsque vous lancez l'application **on vous demande** de vous connectez comme ceci :

Perpi&Co App Project  
Vente et Gestion de produit

Application développée par Adrien F. ©

Se connecter

Nom de profil :

Agent

Mot de passe :

\*\*\*\*\*

Connexion

Si vous êtes une des personnes citées précédemment, vous pouvez si c'est la première fois que vous lancez le logiciel de **saisir les identifiants suivants** :

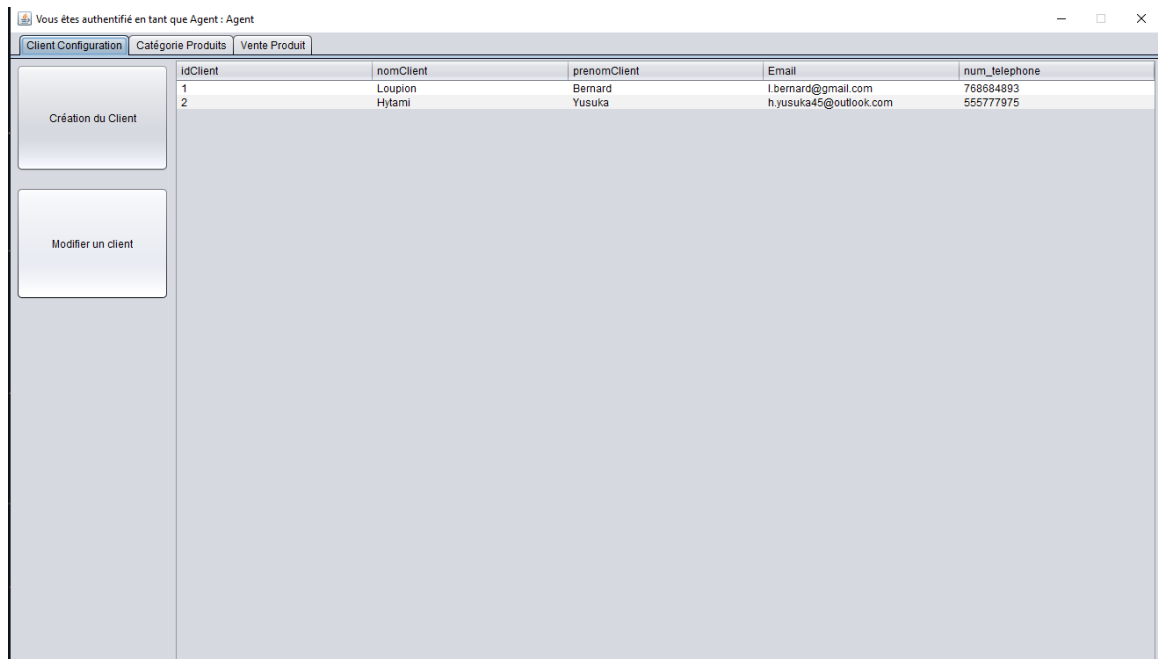
- **Nom de profil** : Agent
- **Mot de Passe** : tneg

(Remarque : Identifiant par défaut fournis par le logiciel, il est recommandé de modifier ces identifiants après vous être connecté par des identifiants plus forts et sécurisés mélangeant chiffres et lettres et au mieux avec des caractères spéciaux)

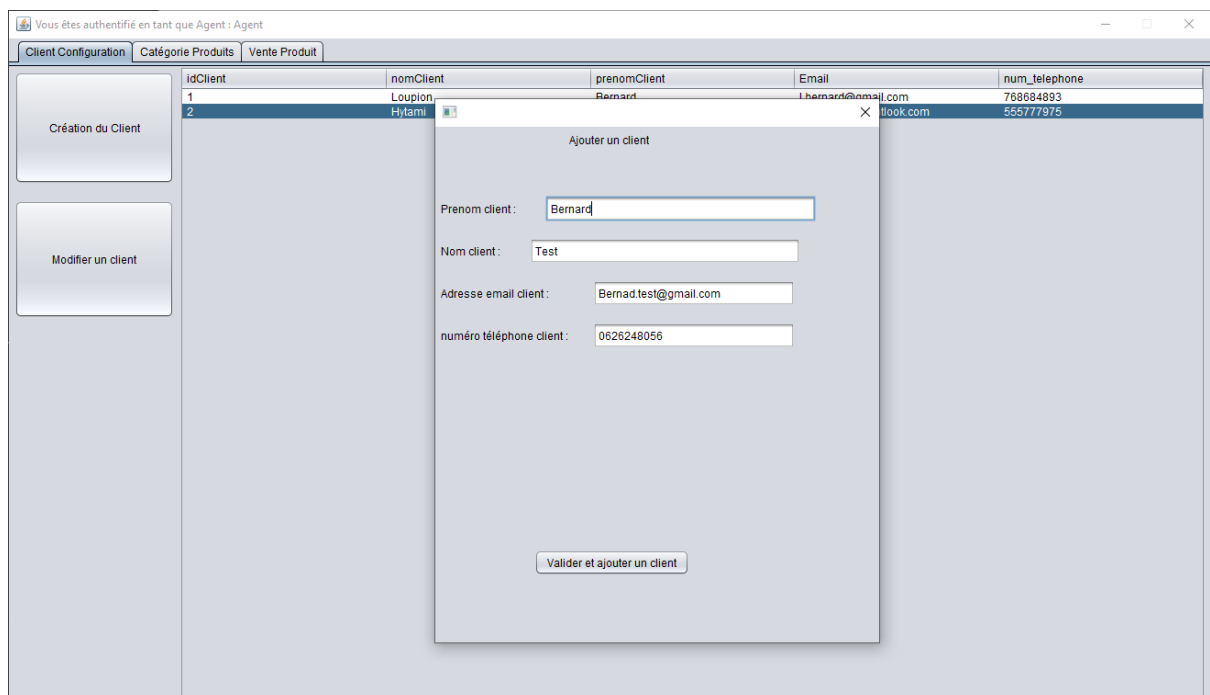
Puis lorsque vous avez saisi les identifiants **appuyez sur** le bouton « **Connexion** » .



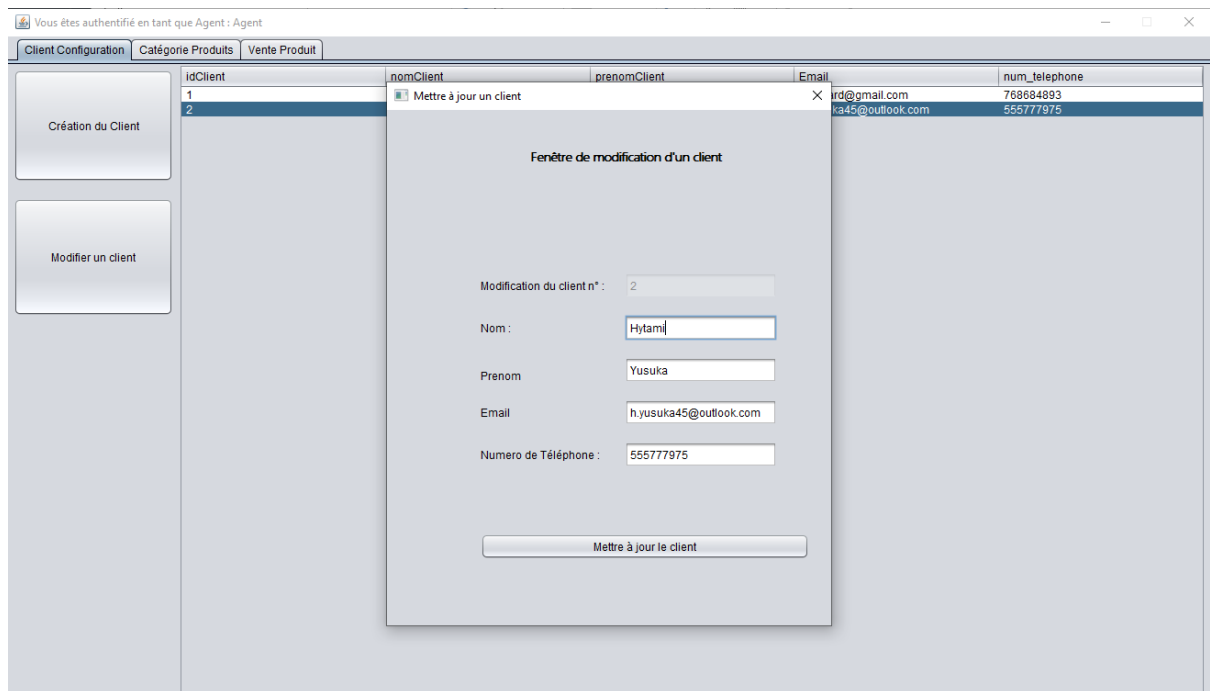
Si vous avez correctement saisi vos identifiants vous avez désormais accès aux outils des agents et arriverais à cette fenêtre au niveau du premier onglet « Client Configuration » c'est ici que vous gérerez tous vos clients.



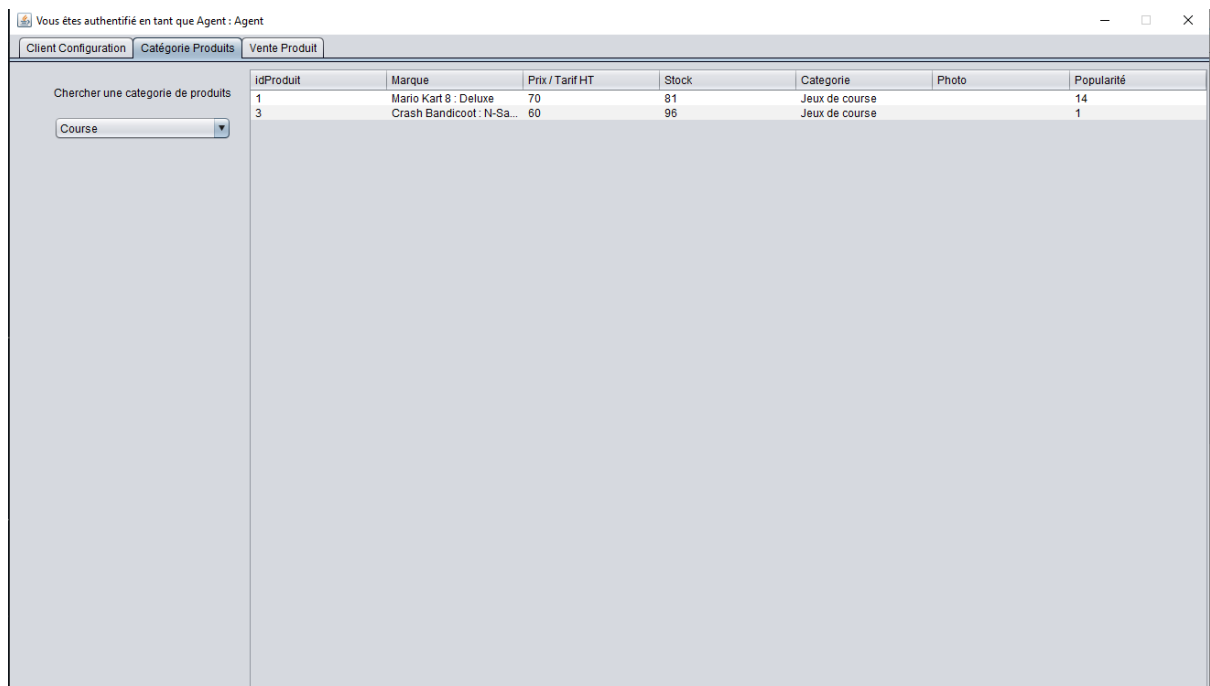
Vous avez la possibilité de créer un client (nom, prénom, email, numéro téléphone).



De plus il vous sera possible de modifier un client déjà créer, par exemple à droite vous avez la liste de tous les clients déjà existants, si vous sélectionner (un clic) et que vous sélectionner à gauche le bouton « modifier un client » une nouvelle fenêtre va s’ouvrir (voir image, ci-dessous) et vous pourrez modifier les informations du client.



En tant qu’agent vous pouvez voir les produits et à quelles catégories appartiennent ces derniers en vous dirigeant dans l’onglet « Catégorie Produits ».



Vous pouvez filtrer grâce à la liste déroulante à gauche la catégorie de produit que vous voulez afficher uniquement.

De plus en tant qu'agent une partie importantes vous serez dédiés, la vente de(s) produit(s) à des clients, pour cela dirigez-vous dans l'onglet « Vente Produit ».

The screenshot shows a software window titled 'Vente Produit' with three tabs: 'Client Configuration', 'Catégorie Produits', and 'Vente Produit'. The 'Vente Produit' tab is active, displaying a table of products available for sale. Below the table is a large empty area for product details. To the right of the table is a summary section with three columns: 'Produit(s)', 'Prix', and 'Quantité'. At the bottom, there are two buttons: 'Ajouter le produit sélectionné au panier' and 'Effectuer la vente'. Below these buttons are two input fields: one for '0' and another for '0' labeled 'Montant HT'.

idProduit	nom produit	tarif produit	stock disponible	description pr...	photo produit	Popularite / Ev...
1	Mario Kart 8 : ...	70	81	Jeux de course		14
2	Mario Tennis : ...	70	74	Jeux de sport		15
3	Crash Bandic...	60	96	Jeux de course		1
4	Mario et Sonic ...	60	95	Jeux de sport		0
5	Animal Crossi...	80	194	Jeux Bac à Sa...		2

Si un client souhaite tel ou tel produit il vous suffit de sélectionner un produit dans la liste, de sélectionner la ligne (un seul clic suffit), puis de cliquer sur le bouton en dessous « Ajouter le produit sélectionné au panier », une fenêtre va s'ouvrir qui vous demandera la quantité que vous voulez pour le produit sélectionné (Voir ci-dessous) :

The screenshot shows a dialog box titled 'Regler la quantité que vous souhaitez soustraire du stock pour le produit : Mario et Sonic : JO Tokyo 2020'. It features a text input field containing the number '1', flanked by minus (-) and plus (+) buttons. Below the input field are two buttons: 'Annuler' and 'Valider'.

En sélectionnant le bouton [+] vous allez pouvoir ajouter +1 et en sélectionnant le bouton [-] vous allez pouvoir retirer -1 le minimum et 1.

Lorsque votre choix est fait sélectionner « Valider », un message vous préviendras que votre produit à bien été ajouter au panier dès lors vous faites « ok » et regardez dans les listes à droites (Voir ci-dessous surligner en jaune le produit ajouté) en effet votre produit a été ajouté :

The screenshot shows a software window titled 'Vente Produit' with three tabs: 'Client Configuration', 'Catégorie Produits', and 'Vente Produit'. The 'Vente Produit' tab is active, displaying a table of available products and a summary section.

**Liste des produits disponibles à la vente :**

idProduit	nom produit	tarif produit	stock disponible	description pr...	photo produit	Popularite / Ev...
1	Mario Kart 8 : ...	70	80	Jeux de course		14
2	Mario Tennis : ...	70	74	Jeux de sport		15
3	Crash Bandic...	60	96	Jeux de course		1
4	Mario et Sonic ...	60	95	Jeux de sport		0
5	Animal Crossi...	80	194	Jeux Bac à Sa...		2

Below the table is a large empty box. To the right, a summary table shows the selected item:

Produit(s)	Prix	Quantité
Mario Kart 8 : Deluxe	70	1

At the bottom, there are two buttons: 'Ajouter le produit sélectionner au panier' and 'Effectuer la vente'. Below the summary table, there is a small table showing the total amount:

	Montant HT
1	70

Vous pouvez voir en bas de la liste des prix le montant total en Euro Hors Taxe et à gauche dans la liste des produits le nombre de produits au panier.

Pour effectuer la vente sélectionner le bouton « Effectuer la vente » une fenêtre va s'ouvrir (voir ci-dessous)

The screenshot shows a window titled 'Effectuer une vente' with a close button (X) in the top right corner. The window is divided into three main sections for data entry.

**Information Client : (IMPORTANT)**

Selectionner un client parmi la liste

Nom du client :

Prenom du client :

Adresse Email du client :

Numéro de téléphone du client :

**Information De l'agent qui effectue la vente: (IMPORTANT)**

Selectionner un agent parmi la liste

Nom de l'agent :

Prenom de l'agent :

Num Tel de l'agent :

**Information de la vente : (IMPORTANT)**

Le numéro de vente :

Date de la vente qui va être effectué:

Pour effectuer la vente, il faut que le client soit déjà créé si ce n'est pas le cas créer le (Voir plus haut pour savoir comment créer un client) puis sélectionner le bouton « Sélectionner un client parmi la liste » au niveau de « information client » puis une liste avec tous vos clients va s'afficher.

idClient	nomClient	prenomClient	Email	num_telephone
1	Loupion	Bernard	l.bernard@gmail.c...	768684893
2	Hylami	Yusuka	h.yusuka45@outlo...	555777975

Valider la sélection du client

Sélectionner le client puis valider la sélection.

**Effectuer une vente**

**Information Client : (IMPORTANT)**

Sélectionner un client parmi la liste

Nom du client: Hylami

Prenom du client: Yusuka

Adresse Email du client: h.yusuka45@outlook.com

Numéro de téléphone du client: 555777975

**Information De l'agent qui effectue la vente: (IMPORTANT)**

Sélectionner un agent parmi la liste

Nom de l'agent:

Prenom de l'agent:

Num Tel de l'agent:

**Information de la vente : (IMPORTANT)**

Le numéro de vente: 0ea91eb0-187d-4eb2-8661-f8d78f730f73 Generer

Date de la vente qui va être effectué: 01-11-2020

Valider la vente

Faites de même pour l'agent à droite. (Pour savoir et garder la trace de quel agent effectue la vente) Lorsque cela est fait vous pouvez valider la vente.

Un bon de commande va s'afficher :

Client Configuration

Catégorie Produits

Vente Produit

Liste des produits disponibles à la vente :

idProduit	nom produit	tarif produit	stock disp
1	Mario Kart 8 : ...	70	80
2	Mario Tennis : ...	70	73
3	Crash Bandic...	60	96
4	Mario et Sonic ...	60	95
5	Animal Crossi...	80	194

Ajouter le produit sélectionner au panier

Facture vente n° 76b74348-b8ab-4f7a-9d7b-ec1855e95b86

Vente effectué avec succès

La vente à bien été enregistré et la commande à bien été validé.  
Annotation : Les tarifs sont aussi présent.

Information Client :

NOM : *Loupion*  
PRENOM : *Bernard*  
EMAIL : *Lbernard@gmail.com*  
NUMTEL : ~~768684893~~

Information Agent qui à effectuer la vente :  
NOM : FIGUERES  
PRENOM : Adrien

Votre facture à été générer localement automatiquement  
Si vous souhaitez la voir dirigez-vous vers votre disque C:\PPE3-facture  
La Facture à bien été enregistrer dans la base de données

Produit(s)  
Mario Tennis : Access

Montant Total Finale : 70

Une facture a été enregistrer sur votre PC localement (C:\\PPE3-facture) et sur le serveur dans la base de données.

(Ci-dessous un exemple de facture générer)

Facture n° : 76b74348-b8ab-4f7a-9d7b-ec1855e95b86

Vente effectué avec succès

La vente à bien été enregistré et la commande à bien été validé.

Information concernant le client :

Nom : Loupion  
Prenom : Bernard  
Numéro de téléphone : 768684893  
Adresse électronique (EMAIL) : lbernard@gmail.com

Information concernant l'agent qui vous as effectuer la vente :  
Nom de l'agent : FIGUERES  
Prenom de l'agent : Adrien

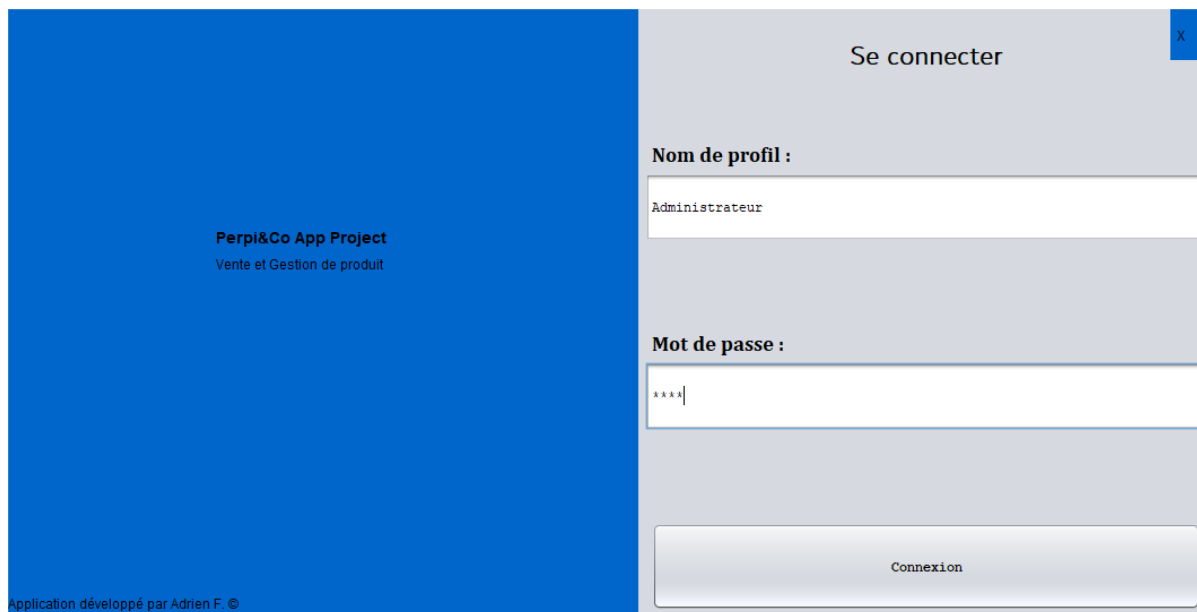
Liste des produits :  
[Mario Tennis : Access]

Montant Final à régler : 70€

## Comment utiliser le logiciel en tant que Administrateur

Vous êtes un membre des équipes Administratives ? Vous ne savez pas comment utiliser l'application ? voici une notice d'utilisation qui vous permettra de comprendre comment fonctionne le logiciel.

Premièrement, lorsque vous lancez l'application **on vous demande** de vous connectez comme ceci :



Perpi&Co App Project  
Vente et Gestion de produit

Application développé par Adrien F. ©

Se connecter x

Nom de profil :  
Administrateur

Mot de passe :  
\*\*\*\*

Connexion

Si vous êtes un administrateur, vous pouvez si c'est la première fois que vous lancez le logiciel de **saisir les identifiants suivants** :

- **Nom de profil** : Administrateur
- **Mot de Passe** : toor

(Remarque : Identifiant par défaut fournis par le logiciel, il est recommandé de modifier ces identifiants après vous être connecter par des identifiants plus fort et sécurisé mélangeant chiffre et lettre et au mieux avec des caractères spéciaux)

Puis lorsque vous avez saisis les identifiants **appuyez sur** le bouton « **Connexion** ».

Vous arriverais sur une nouvelle fenêtre dans le premier onglet « Gestion produit et Catégorie »

Dans cet onglet vous pouvez « Ajouter un produit », « supprimer un produit sélectionner » dans la liste à droite, ou bien « modifier le produit sélectionner » dans la liste à droite

Vous êtes authentifié en tant que Administrateur : Administrateur

Gestion Produit et Catégorie | Gestion des agents | Gestion des ventes | Gestion des clients

Ajouter un Produit

Supprimer le produit sélectionner

Modifier le produit sélectionner

Vous pouvez voir la popularité  
Des produits à partir de la colonne  
Popularité du tableau

IdProduit	nom produit	tant produit	stock produit	description produit	photo produit	Popul
1	Mario Kart 8 : Deluxe	70	80	Jeux de course		15
2	Mario Tennis : Access	70	73	Jeux de sport		16
3	Crash Bandicoot : N-Sane tril...	60	96	Jeux de course		1
4	Mario et Sonic : JO Tokyo 2020	60	95	Jeux de sport		0
5	Animal Crossing : New Horiz...	80	194	Jeux Bac à Sable		2

En tant qu'administrateur vous avez accès et la possibilité de modifier les agents existants pour ce faire dirigez-vous vers l'onglet « gestion des agents ». Pour modifier un agent sélectionner la ligne dans le tableau à droite correspondant à l'agent que vous voulez modifier puis cliquez sur le bouton à gauche « modifier l'intégralité des informations de l'agent sélectionner » une fenêtre va s'ouvrir

Vous êtes authentifié en tant que Administrateur : Administrateur

Gestion Produit et Catégorie | Gestion des agents | Gestion des ventes | Gestion des clients

Modifier l'intégralité des informations de l'agent sélectionner

Supprimer l'agent sélectionner

IdAgent	nom de l'agent	prenom de l'agent	numéro tel de l'agent	n
1	FIGUERES	Adrien	0638319676	1
2	Lipie	Bernard	0780689345	2

idProfil	Username	Password	Num Tel	Email
1	Agent	tnega	0638319676	figueres.adrien@gmail.com
2	Administrateur	toor	062541655416	user@gmail.com

Le tableau ci-contre à droite affiche uniquement les profils des agents  
Si vous voulez modifier un profil sélectionner le bouton "Modifier l'intégralité ..."



Id de l'agent : 2

Nom de l'agent : Lipie

Prenom de l'agent : Bernard

Numéro de tél. de l'age... : 0780689345

Id de profil associé : 2

Voir le profil de l'agent

Valider les modifications de l'agent

Vous pouvez modifier le profil correspondant à l'agent en cliquant sur le bouton « Voir le profil de l'agent » puis de nouvelles options seront disponibles et lorsque vous avez fini les modifications sélectionner le bouton « valider les modifications de l'agent »

Si vous souhaitez voir toutes les ventes qui ont été effectuées dirigez-vous vers l'onglet « gestion des ventes »

Vous êtes authentifié en tant que Administrateur : Administrateur

Gestion Produit et Catégorie Gestion des agents Gestion des ventes Gestion des clients

idvente	nom	prenom	numero_commande	date_vente	Facture PDF
2	Loupion	Bernard	fda76a1a-3204-4647-bd26-2a1660...	2020-10-31	
9	Loupion	Bernard	9b1d5c96-0262-4b7a-9eab-18493a...	2020-10-31	
13	Loupion	Bernard	21bc3373-9057-4beb-9341-62014c...	2020-10-31	
14	Loupion	Bernard	4f2a4b2e-7c8a-415d-9320-e28a13...	2020-10-31	
19	Loupion	Bernard	5d3be77a-ec45-4499-877b-c803e1...	2020-11-01	
20	Loupion	Bernard	76b74348-b8ab-4f7a-9d7b-ec1855...	2020-11-01	
21	Loupion	Bernard	76b74348-b8ab-4f7a-9d7b-ec1855...	2020-11-01	
1	Hytami	Yusuka	332c243d-3d2a-4190-ba9c-e9086c...	2020-10-31	
3	Hytami	Yusuka	069b2a16-6849-4365-b8e3-6107e4...	2020-10-31	
4	Hytami	Yusuka	54441bcc-7637-446f-83ec-25875ba...	2020-10-31	
6	Hytami	Yusuka	d7c400c9-d88f-4db7-ac5d-bd55f4b...	2020-10-31	
7	Hytami	Yusuka	d7c400c9-d88f-4db7-ac5d-bd55f4b...	2020-10-31	
8	Hytami	Yusuka	6026d999-6717-4db1-9358-674bd5...	2020-10-31	
10	Hytami	Yusuka	aabafbed-b083-4d57-9998-64892b4...	2020-10-31	
11	Hytami	Yusuka	a6597438-7f84-4c71-be25-74d720...	2020-10-31	
15	Hytami	Yusuka	39fbcdb0-c1ed-47cc-a172-122caac...	2020-10-31	
16	Hytami	Yusuka	5a6d3484-6485-4fb1-9b60-a06ce7...	2020-11-01	
17	Hytami	Yusuka	e832b669-4b69-48a1-bd40-3b8ed7...	2020-11-01	
19	Hytami	Yusuka	0ea91eb0-187d-4eb2-8661-f8d787...	2020-11-01	
2	Loupion	Bernard	fda76a1a-3204-4647-bd26-2a1660...	2020-10-31	
9	Loupion	Bernard	9b1d5c96-0262-4b7a-9eab-18493a...	2020-10-31	
13	Loupion	Bernard	21bc3373-9057-4beb-9341-62014c...	2020-10-31	

Annuler la vente (Suppression)

\*Si une vente est considérée comme annulée. Il est possible en sélectionnant la vente dans le tableau (Ligne du tableau) de la supprimer en cliquant sur le bouton "Annuler la vente"

Vous pouvez en sélectionnant une des ventes dans la liste supprimer la vente (l'annuler) pour ce faire cliquez sur le bouton « Annuler la vente (Suppression) »

Vous pouvez gérer les clients pour ce faire dirigez-vous vers la gestion des clients

Vous êtes authentifié en tant que Administrateur : Administrateur

Gestion Produit et Catégorie Gestion des agents Gestion des ventes Gestion des clients

Création d'un client

Modifier le client sélectionner

Supprimer le client sélectionner

idClient	nomClient	prenomClient	Email	num_telephone
1	Loupion	Bernard	t.bernard@gmail.com	768684893
2	Hytami	Yusuka	h.yusuka45@outlook.com	555777975

Vous pouvez comme les autres onglets, créer un client, modifier un client, et/ou supprimer un client

## Amélioration(s) envisageable(s)

A l'heure actuelle logiciel comme il est possible de constater n'est qu'à une version ALPHA avec beaucoup de chose à améliorer ou changer ou ajouter.

De ce fait une liste de ce qui est déjà penser à améliorer, ajouter ou modifier dans le futur est déjà faite :

- Amélioration de l'esthétique (Design, apparence)
- Amélioration de l'ergonomie (Prise en main pour un utilisateur lambda plus facile)
- Ajout d'un bouton déconnexion (Mécanique)