



ÉCOLE SUPÉRIEURE DE LA STATISTIQUE ET DE L'ANALYSE
DE L'INFORMATION

UNIVERSITÉ DE CARTHAGE

Rapport du PFA : Atelier Statistique de deuxième année

Réalisé par :

ABDERRAHIM JRIDI

WASSIM BEN DALI

Gestion des commandes en magasin

Encadré par :

SIHEM MANSOUR

Année universitaire : 2020/2021

07/06/2021

Table des figures

1	Power AMC	5
2	SGBD ORACLE	5
3	Diagramme de cas d'utlisation	6
4	Diagramme de classe	8
5	Diagramme de séquence(Client)	10
6	Diagramme de séquence(Admin)	12
7	Le pilote JDBC	13
8	Classe Personne	14
9	Classe Facture	14
10	Classe Produit	14
11	Classe stock	14
12	JFreeChart	15
13	Java swing	15
14	Fenêtre "Bienvenu"	16
15	Fenêtre d'inscription	17
16	Fenêtre de connection	18
17	Fenêtre d'ajout stock	19
18	Fenêtre de selection produit	20
19	Composition du panier	21
20	Code	23

Table des matières

1	Introduction	4
2	Conception	5
2.1	Outils	5
2.2	Diagramme de cas d'utilisation	6
2.3	Diagramme de classe	8
2.4	Diagramme de séquences	10
3	Base de données	13
3.1	Outils	13
3.2	Imprimés écrans	14
4	Realisations	15
4.1	Outils	15
4.2	Imprimés écrans	16
4.3	Partie statistique	21
5	Conclusion	22
6	Annexe	23
7	Bibliographie	24

Remerciements

Nous voulons commencer dans notre rapport par des remerciements à ceux qui nous ont beaucoup appris au cours de ce projet de fin d'année et à ceux qui ont eu la gentillesse de faire de cet atelier un moment très profitable.

Nous tenons à remercier vivement notre encadrante de cet atelier statistique Mme Sihem mansour, professeur en informatique au sein de l'Ecole supérieure de la statistique et de l'analyse de l'information, pour sa gentillesse, sa patience, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à ses conseils et sa confiance que nous avons pu nous accomplir totalement dans nos missions.

1 Introduction

Ce module d'atelier statistique de deuxième année était une opportunité pour approfondir et appliquer les connaissances acquises en statistique et en programmation lors de notre formation à l'école supérieure de la statistique et de l'analyse de l'information.

Notre travail concerne la gestion des commandes et de stock en magasin.

Nous avons effectué l'analyse conceptuelle de notre système en générant les différents diagrammes (diagramme de cas d'utilisation, diagramme de classe et diagramme de séquence).

Nous avons également récupéré notre base de données en générant les différents tables en oracle. Nous avons manipulé aussi les interfaces graphiques en créant des fenêtres permettant la gestion des commandes par les différents acteurs de notre système (clients, responsable paiement, responsable stock...).

2 Conception

2.1 Outils

La partie conceptuelle consiste à effectuer les différents diagrammes :

- Diagramme de cas d'utilisation
- Diagramme de classes
- Diagramme de séquences

Ces diagrammes permettent de mieux comprendre le système, de visualiser son architecture et de modéliser ses besoins.

Pour générer ces diagrammes, nous avons utilisé le logiciel PowerAMC qui permet de modéliser les différents traitements de gestion de commandes en magasin et de récupérer les résultats en tables de base de données qui seront ensuite directement exploitable par le SGBD oracle.

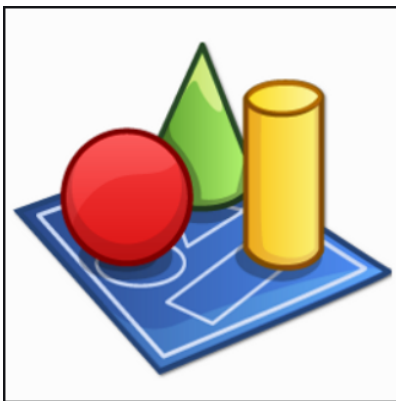


FIGURE 1 – Power AMC



FIGURE 2 – SGBD ORACLE

2.2 Diagramme de cas d'utilisation

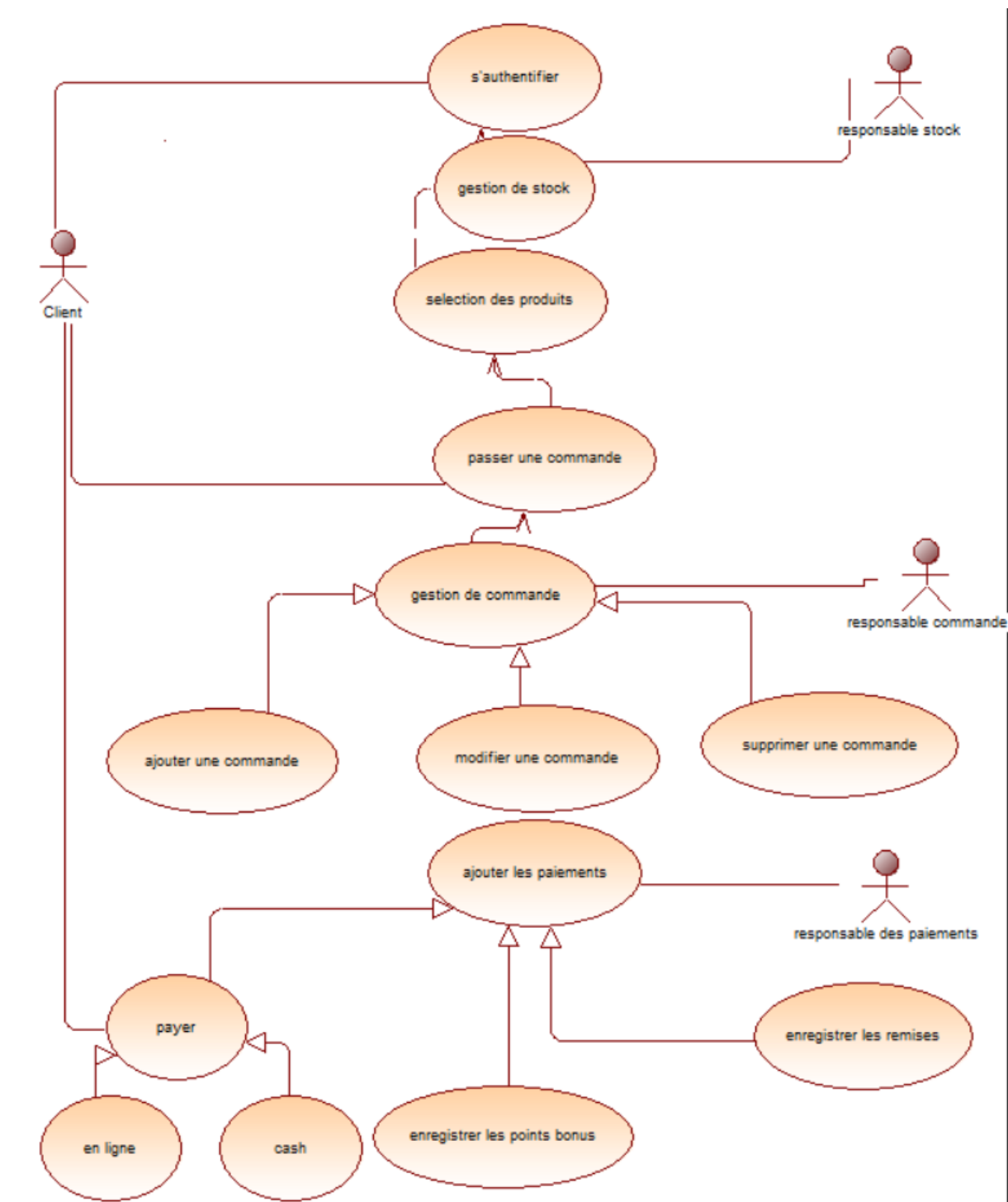


FIGURE 3 – Diagramme de cas d'utilisation

*** Les acteurs :**

Acteur principal : client

Acteurs secondaires :

-Responsable stock

-Responsable paiement

-Responsable commandes

*** Description du diagramme :**

Le client qui veut passer des commandes en magasin doit tout d'abord se connecter au système de gestion de commandes ensuite il peut sélectionner les produits qu'il veut acheter et passer sa commande au système.

Un responsable de commandes permet de gérer les différents commandes. Il s'occupe de vérifier la commande, si elle est valide il l'ajoute au système sinon il peut la modifier ou la supprimer.

Une fois la commande est validée et ajoutée au système par le responsable commande, le responsable de paiement enregistre la commande et détermine le montant à payer, il enregistre aussi les points bonus pour les clients fidèles qui disposent d'une carte de fidélité. Le client a le choix de payer sa commande cash ou bien en ligne.

De plus, le système dispose d'un responsable stock qui permet de gérer les différents produits stockés en magasin.

2.3 Diagramme de classe

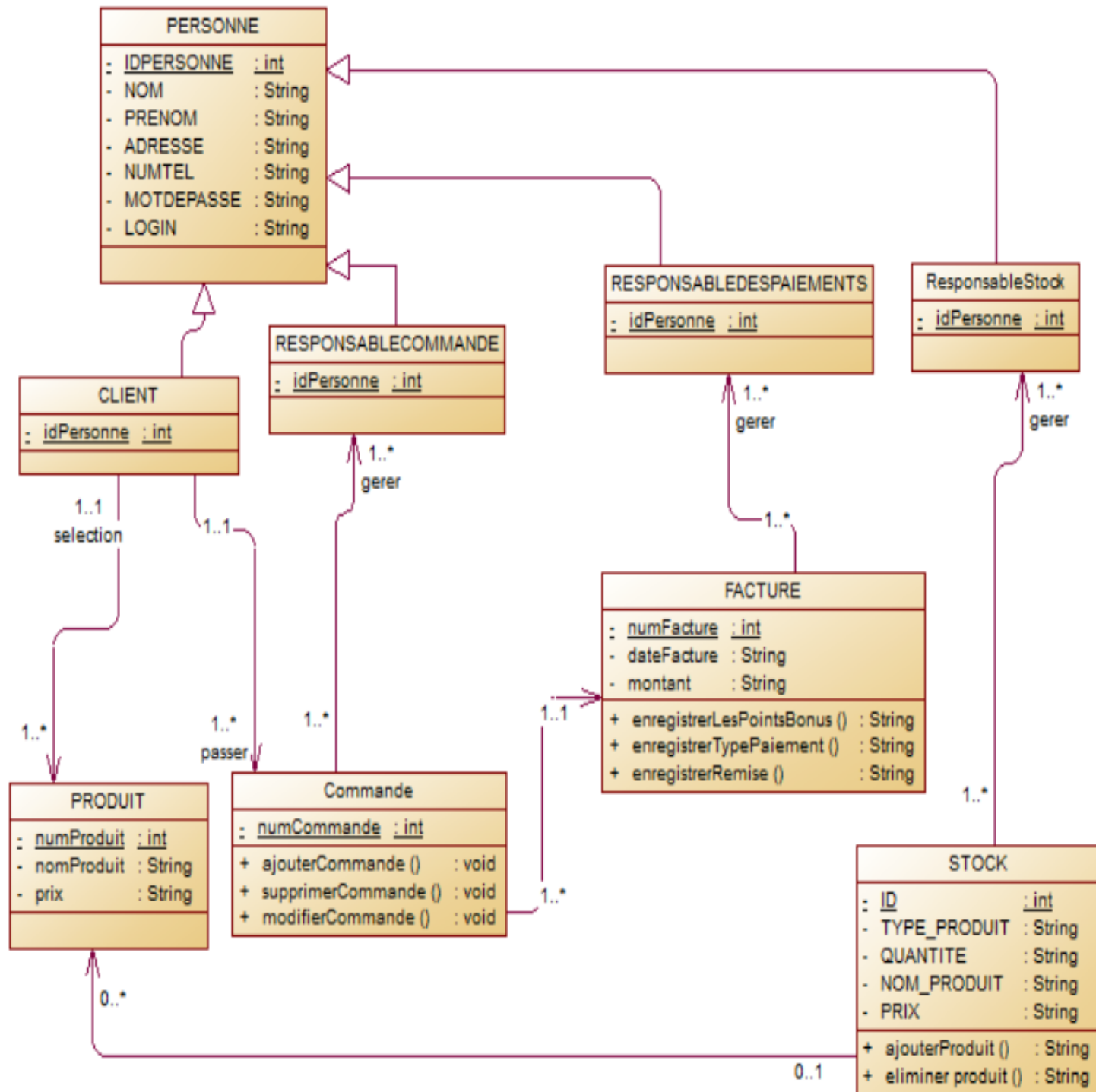


FIGURE 4 – Diagramme de classe

*** Description du diagramme :**

Le diagramme de classe est modélisé par une classe Personne qui peut être un client, un responsable commande, un responsable de paiement ou bien un responsable stock.

Une Personne est déterminée par son identité (idPersonne), son nom, son prénom, son adresse, son numéro de téléphone, son mot de passe et son login.

Un client peut sélectionner un ou plusieurs produits. un produit est caractérisé par son numéro (numProduit), son nom et son prix.

Chaque produit peut être stocké dans un dépôt. il est caractérisé par son identité (idStock), son type et sa quantité. il peut être ajouté ou éliminé du stock.

Chaque stock est géré par un ou plusieurs responsable stock.

A un client peuvent être associées une ou plusieurs commandes, chacune est caractérisée par un numéro (numCommande) et peut être ajoutée, modifiée ou supprimée du système.

Chaque commande est associée à une ou plusieurs factures. une facture est caractérisée par son numéro (numFacture), une date et un montant, et pour chaque facture on peut enregistrer les points bonus, la remise et le type de paiement.

2.4 Diagramme de séquences

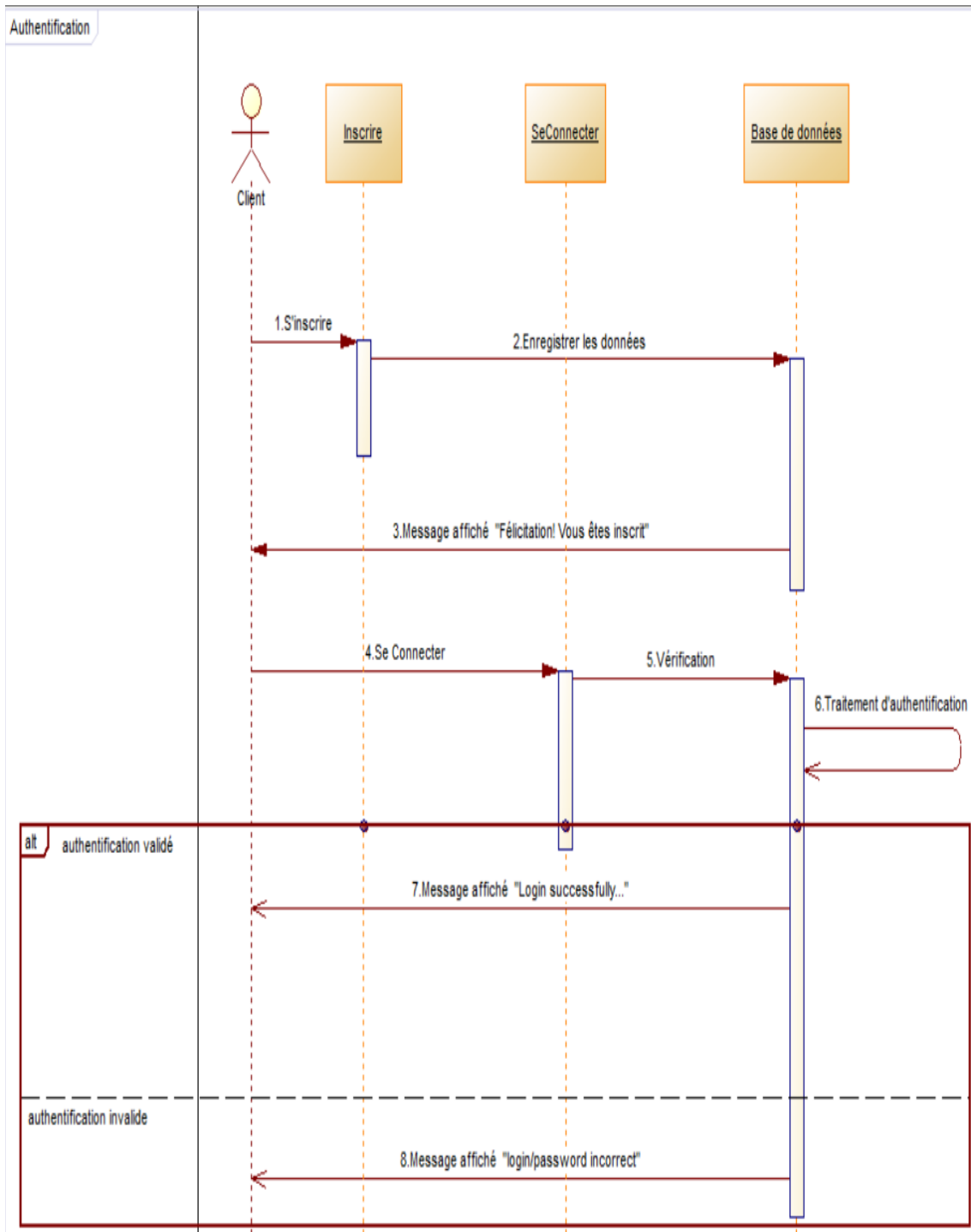


FIGURE 5 – Diagramme de séquence(Client)

*** Description du diagramme :**

Dans la phase de l'authentification, le client doit tout d'abord s'inscrire en écrivant tous les données nécessaires. S'il y a une information que le client a oublié de l'écrire, un message d'erreur apparait. Sinon l'inscription sera terminée avec succès et ces informations vont être enregistrées automatiquement dans la base de données.

Une fois le client est inscrit, il peut se connecter en introduisant son login et son mot de passe.

On vérifie si ces informations existent déjà dans la base de données et si ces données sont valides, un message sera affiché "Login successfully..." et le client peut consulter l'interface Produit.

Sinon un message d'erreur sera affiché "login/password incorrect".

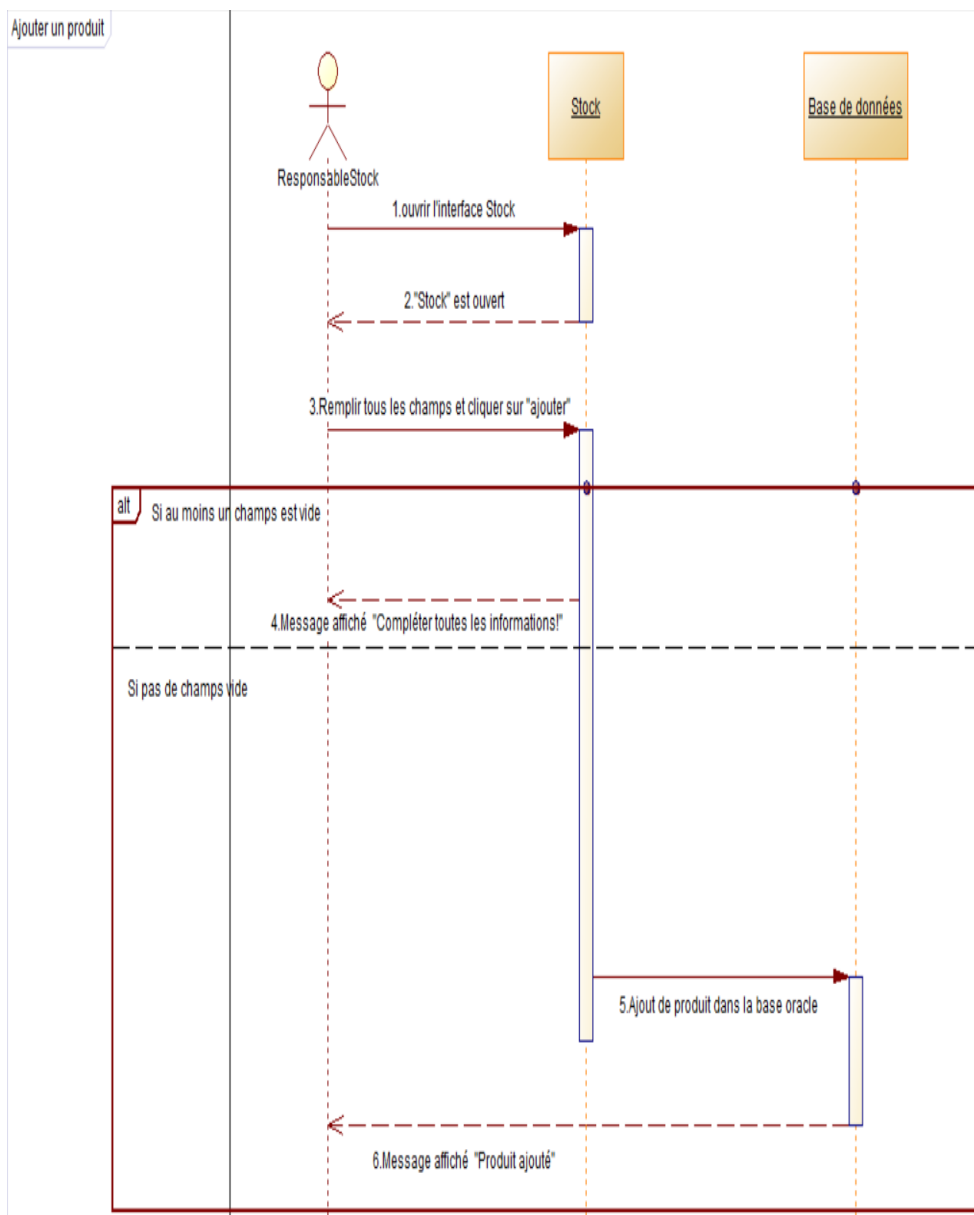


FIGURE 6 – Diagramme de séquence(Admin)

* Description du diagramme :

Dès que le responsable de stock ouvre l'interface stock, il peut ajouter un produit en cliquant sur le bouton ajouter. S'il n'a pas remplis tous les champs, un message d'erreur s'affiche. Sinon, ce produit sera ajouté dans la base de données et un message sera affiché au responsable stock "Produit ajouté".

3 Base de données

3.1 Outils

Nous avons utilisé le SGBD oracle g10 pour générer les tables de la base de données. Ces tables doivent ensuite être connectées aux différentes fenêtres créés en Java. Nous avons ainsi installé le pilote Oracle.jdbc pour faire la connection entre Java et oracle.

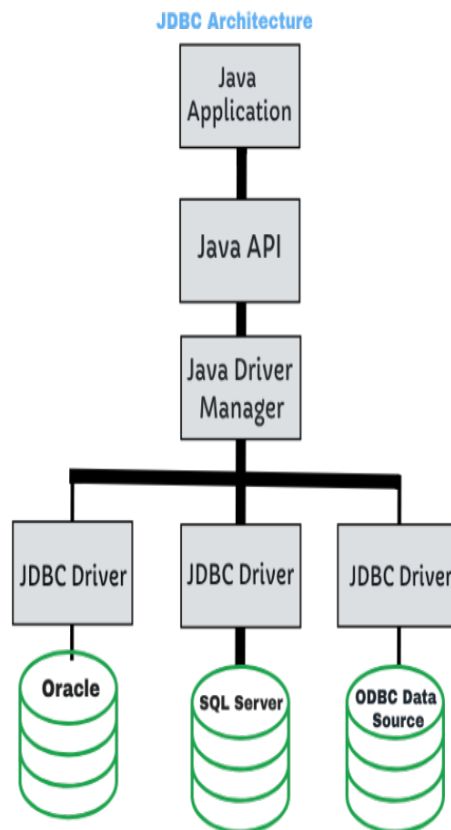


FIGURE 7 – Le pilote JDBC

3.2 Imprimés écrans

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
IDPERSONNE	NUMBER	No	-	1
NOM	VARCHAR2(254)	Yes	-	-
PRENOM	VARCHAR2(254)	Yes	-	-
ADRESSE	VARCHAR2(254)	Yes	-	-
NUMTEL	VARCHAR2(254)	Yes	-	-
MOTDEPASSE	VARCHAR2(254)	Yes	-	-
LOGIN	VARCHAR2(254)	Yes	-	-
				1 - 7

FIGURE 8 – Classe Personne

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
numFacture	NUMBER	No	-	1
dateFacture	VARCHAR2(254)	Yes	-	-
montant	VARCHAR2(254)	Yes	-	-
				1 - 3

FIGURE 9 – Classe Facture

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
numProduit	NUMBER	No	-	1
nomProduit	VARCHAR2(254)	Yes	-	-
prix	VARCHAR2(254)	Yes	-	-
				1 - 3

FIGURE 10 – Classe Produit

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
ID	NUMBER	No	-	1
TYPE_PRODUIT	VARCHAR2(254)	Yes	-	-
QUANTITE	VARCHAR2(400)	No	-	-
NOM_PRODUIT	VARCHAR2(4000)	Yes	-	-
PRIX	VARCHAR2(4000)	Yes	-	-
				1 - 5

FIGURE 11 – Classe stock

4 Realisations

4.1 Outils

Nous avons manipulé les interfaces graphiques en créant des fenêtres permettant la gestion des commandes.

Ces fenêtres sont formées de plusieurs composantes graphiques : boutons, champs textuels, barre de menu..

Nous avons utilisé les packages javax.swing, java.sql et java.awt fournies en Java.

Pour la partie statistique, nous avons installé le pilote "JFreeChart" et après nous avons importé le package org.jfree..



FIGURE 12 – JFreeChart



FIGURE 13 – Java swing

4.2 Imprimés écrans

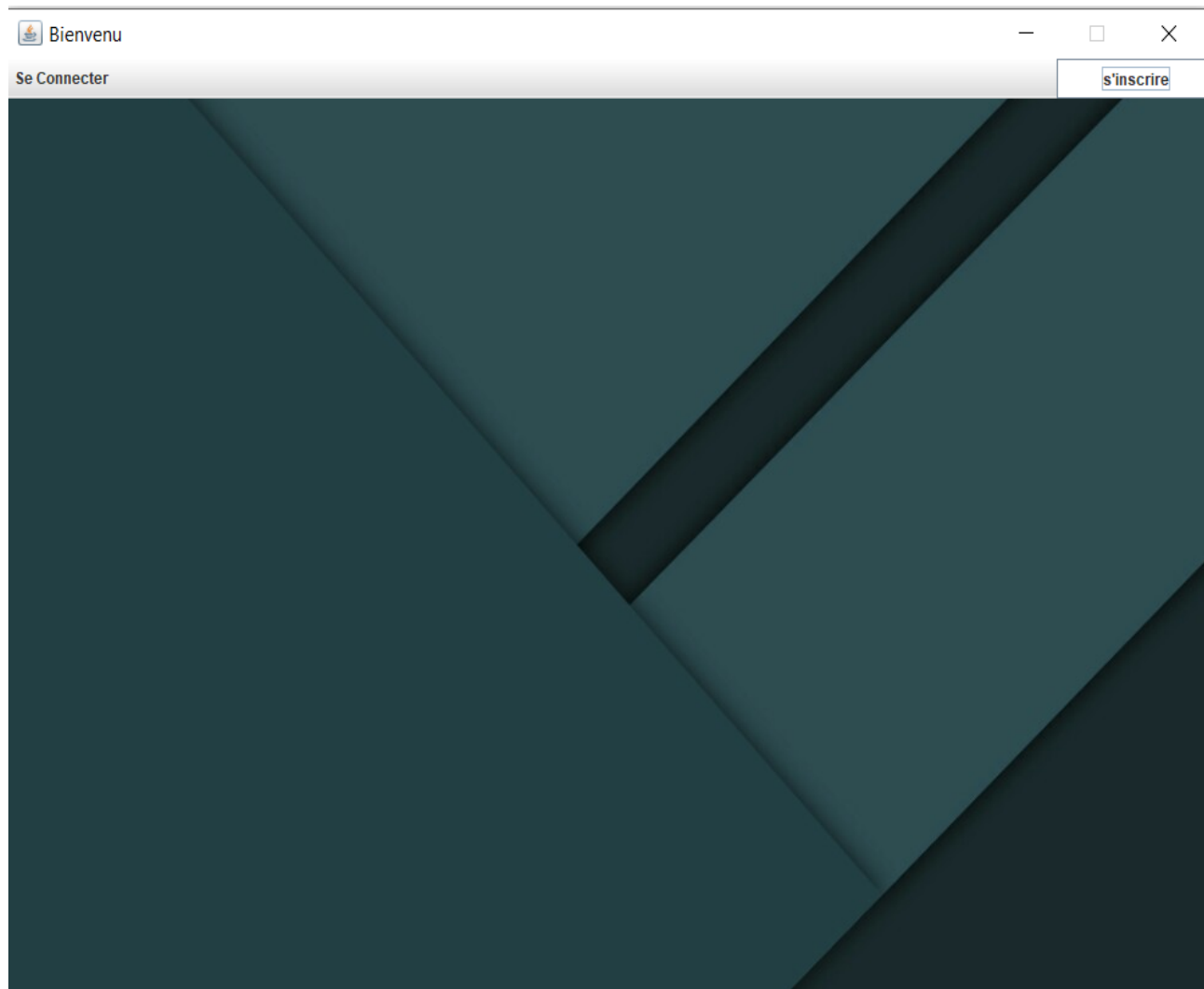


FIGURE 14 – Fenêtre "Bienvenu"

Nous avons créé cette première fenêtre "Bienvenu" qui permet au client de s'inscrire dans le système ou bien de se connecter s'il dispose déjà d'un compte.

The image shows a graphical user interface for a registration process. The window has a title bar with the text 'Inscrire' and standard window controls (minimize, maximize, close). Below the title bar is a tab labeled 'Bienvenu'. The main area of the window has a dark teal background with a diagonal pattern. It contains six labels in a light blue font, each followed by a white rectangular input field:

- introduire votre nom
- introduire votre prenom
- introduire votre adresse
- introduire votre numero de telephone
- introduire votre mot de passe
- introduire votre login

At the bottom of the window, there are two buttons: 'valider' and 'annuler', both with a light blue gradient and a slight shadow.

FIGURE 15 – Fenêtre d'inscription

L'inscription se fait en remplissant toutes les informations concernant le nom, le prénom, l'adresse, le numéro de téléphone, le mot de passe et le login.

Une fois les champs remplis, l'utilisateur peut terminer son inscription en cliquant sur le bouton valider. Il peut aussi annuler la procédure à tout moment en cliquant sur le bouton annuler.

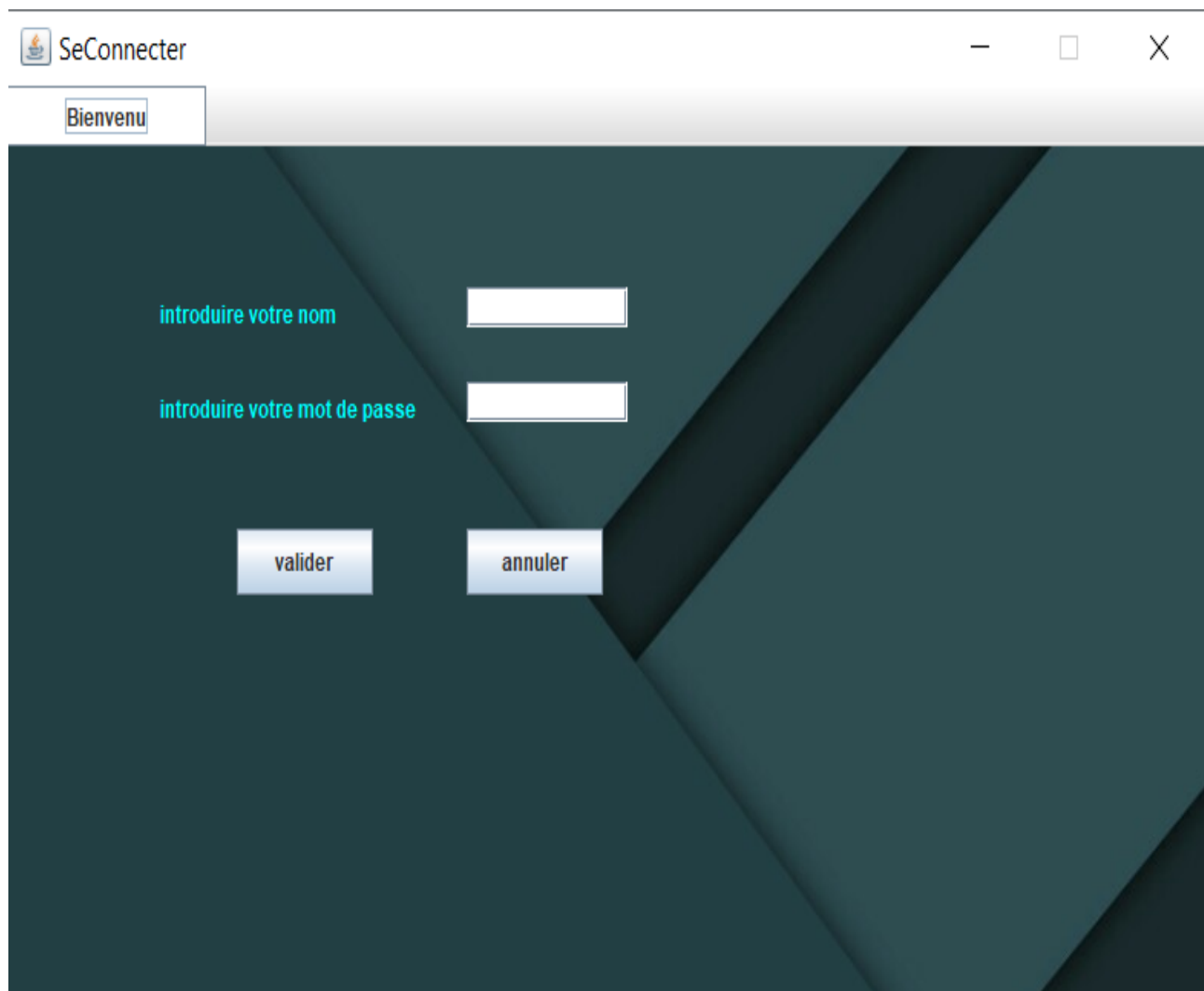


FIGURE 16 – Fenêtre de connection

Une fois inscrit, le client peut se connecter au système à tout moment en introduisant son nom et son mot de passe.

Cette fenêtre dispose aussi de deux boutons qui permettent de valider les champs remplis ou bien de les annuler et de commencer à nouveau le processus.

Stock

Bienvenu

Type Produit:

Téléviseur

Quantité:

1

Nom Produit:

LED HD + Récepteur Int Noir

Prix Produit:

829,000 DT

Ajouter

Supprimer

Modifier

Clear

Type produit	Quantité	Nom Produit	Prix
SmartPhone	2	SMARTPH...	239,000 DT
SmartPhone	3	SMARTPH...	269,000 DT
SmartPhone	1	Huawei Y7 ...	549,000 DT
Téléviseur	1	Téléviseur ...	549,000 DT
Téléviseur	1	Téléviseur ...	829,000 DT
Téléviseur	4	Téléviseur ...	989,000 DT
Video Proje...	4	Vidéo Proje...	999,000 DT
Video Proje...	2	Vidéo proje...	1 029,000 ...
Video Proje...	2	Vidéo Proje...	1 033,000 ...

FIGURE 17 – Fenêtre d'ajout stock

La fenêtre "stock" est composée d'une table contenant le type de produit, la quantité, le nom et le prix. Le responsable du stock peut ajouter des nouveaux produits à stocker dans le dépôt. Il doit ainsi introduire le nom du produit, son type, sa quantité et son prix. Une fois ses champs remplis, il doit cliquer sur le bouton ajouter pour insérer ce nouveau produit dans la liste des produits stockés. Cette fenêtre de stock dispose aussi d'autres boutons qui permettent au responsable de stock de suspendre un produit de la liste, de modifier un produit ou de supprimer toute la liste.

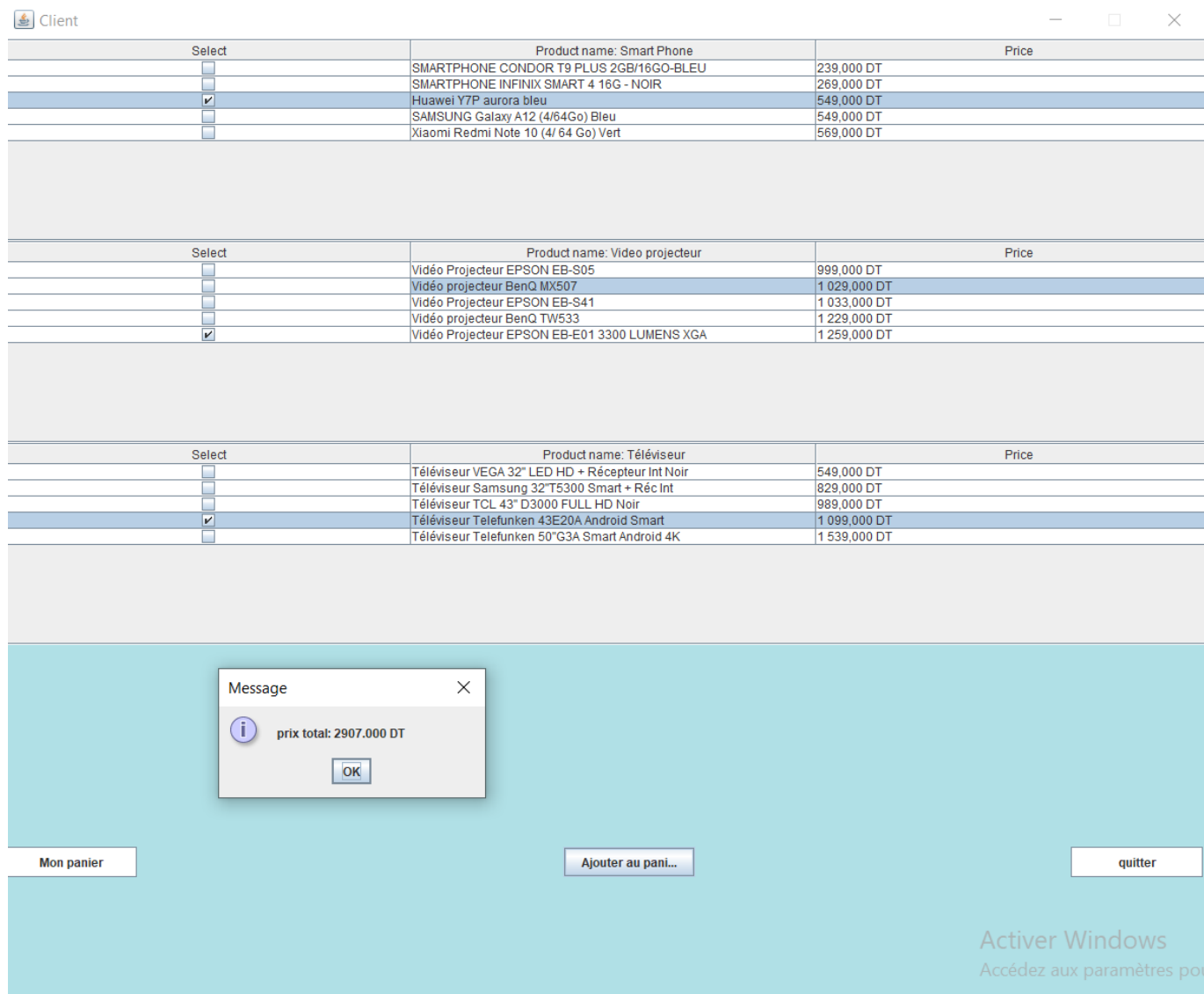


FIGURE 18 – Fenêtre de selection produit

Le client peut choisir des produits et les ajouter à son panier en cliquant sur le bouton 'ajouter au panier'. Une fois le panier rempli un message s'affiche à l'écran indiquant les noms des produits ajoutés et le prix total à payer. Cette fenêtre contient aussi un bouton 'quitter' qui permet au client d'annuler l'achat à tout moment ou bien de se connecter une autre fois en dirigeant vers la fenêtre "SeConnecter". Ce dernier peut aussi consulter à son panier en cliquant sur le bouton "Mon panier".

4.3 Partie statistique

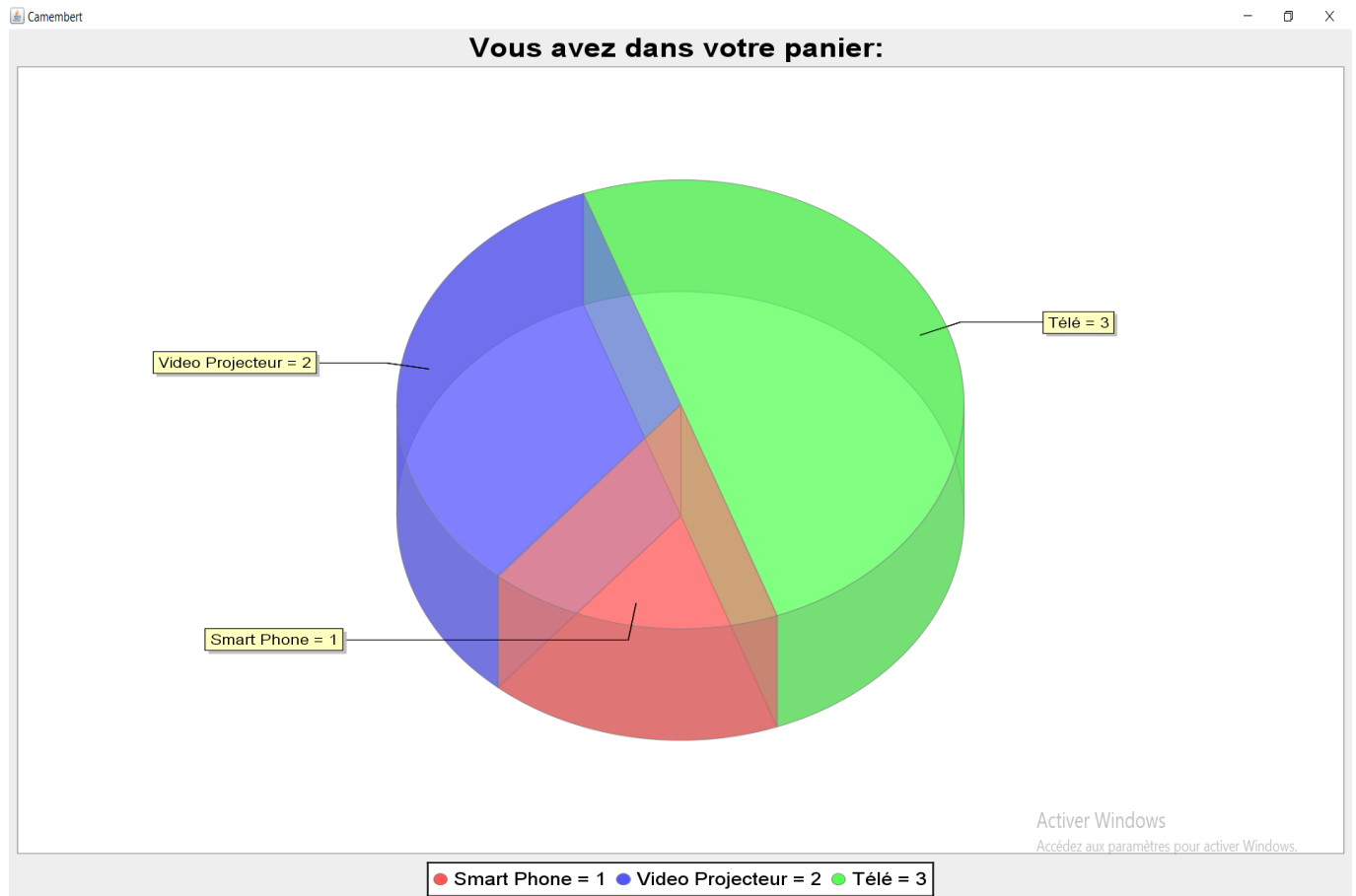


FIGURE 19 – Composition du panier

Nous avons réalisé un diagramme en camembert (diagramme en secteurs) qui indique le nombre des produits achetés pour chaque type.

5 Conclusion

Pour conclure, cet atelier statistique était une opportunité pour appliquer tous les acquis théorique en programmation orienté objet et les nouveaux concept du langage UML.

C'était aussi une occasion pour mieux programmer en Java et pour s'adapter aux interfaces graphiques tout en manipulant les bases de données avec le SGBD oracle.

Malgré tout ce qu'on a appris au cours de cet atelier statistique, nous aurions pu utlisier d'autres bibliothèques graphiques comme "JCalendar".

6 Annexe

```
JButton btnAdd = new JButton("Ajouter");
btnAdd.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (type.getText().equals("") || quant.getText().equals("") || nom.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Compléter toutes les informations!");
        } else {
            row[0] = type.getText(); row[1] = quant.getText(); row[2] = nom.getText(); row[3] = prix.getText();
            model.addRow(row);
            try {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "user", "pfa");
                String sql = "SELECT MAX(ID) FROM stock";
                PreparedStatement ps = con.prepareStatement(sql);
                ResultSet rs = ps.executeQuery();
                if (rs.next() == true) {
                    id = rs.getInt(1);
                } else {
                    JOptionPane.showMessageDialog(null, " ");
                }
                String sql1 = "insert into stock values(?,?,?,?)";
                PreparedStatement pStmt = con.prepareStatement(sql1);
                pStmt.setInt(1, (id + 1));
                pStmt.setString(2, type.getText());
                pStmt.setInt(3, Integer.parseInt(quant.getText()));
                pStmt.setString(4, nom.getText());
                pStmt.setString(5, prix.getText());
                pStmt.executeUpdate();
                con.close();
            } catch (SQLException se) {
                se.printStackTrace();
            } catch (Exception d) {
                d.printStackTrace();
            }
            type.setText(""); quant.setText(""); nom.setText(""); prix.setText("");
        }
    }
});
```

FIGURE 20 – Code

Cette partie du code concerne l'insertion des champs concernant un produit à ajouter dans le stock. les champs à insérer sont le nom du produit, son type, son prix et sa quantité qui doit être un entier.

chaque produit est caractérisé par un identifiant unique l' "id". l' "id" s'incrémente de 1 à chaque insertion d'un nouveau produit dans le stock.

A chaque fois le produit concerné est ajouté au stock, il sera automatiquement inséré dans la base de données.

7 Bibliographie

Références

- [1] <https://drive.google.com/file/d/1tjFXJ-ttgLu0OFMZtMivynkUXcoGgaln/view?ts=60a82003>
installation du pilote OJDBC
- [2] <https://www.jfree.org/>
installation du pilote JFreeChart
- [3] <https://stackoverflow.com/>
- [4] <https://java.developpez.com>
- [5] <https://www.overleaf.com/>
- [6] <https://www.jmdoudoux.fr/java/dej/chap-swing.htm>
- [7] https://en.wikipedia.org/wiki/Java_package