

I. Introduction

Après avoir établi une étude conceptuelle de notre système, nous passons à l'implémentation de l'application définis et détaillée au chapitre précédent tout en présentant les outils utilisés et en expliquant les écrans de notre logiciel. Cette implémentation est basée sur une architecture 2-tiers et réalisée avec le langage de programmation Java en utilisant l'IDE (l'environnement de développement intégré) NetBeans et le SGBD Derby.

II. Architecture du système

Nous commençons tout d'abord par définir l'architecture 2-tiers et nous passons après à la présentation de l'architecture de notre système.

II.1 Qu'est ce qu'une architecture 2-tiers ?

L'architecture à deux niveaux (aussi appelée architecture 2-tiers) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.[11]

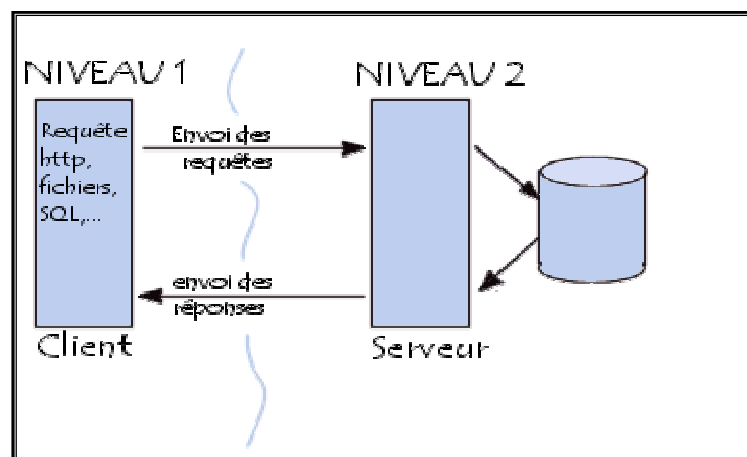


Figure III.1 : Architecture 2-Tiers

- Ce type d'application permet de tirer partie de la puissance des ordinateurs déployés en réseau pour fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données, qui restent gérées de façon centralisée.
- La gestion des données est prise en charge par un SGBD centralisé, s'exécutant le plus souvent sur un serveur dédié.

- Ce dernier est interrogé en utilisant un langage de requête qui, le plus souvent, est SQL.
- Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes. Ce dialogue nécessite l'instauration d'une communication entre client et serveur.

II.2. architecture du système GML

L'architecture de notre système est basée sur l'architecture 2-tiers, plus précisément l'application cliente utilise JDBC pour parler directement avec le SGBD Derby qui gère notre base de données (magasin). Cette architecture est illustré dans la figure suivante (figure III.2) .

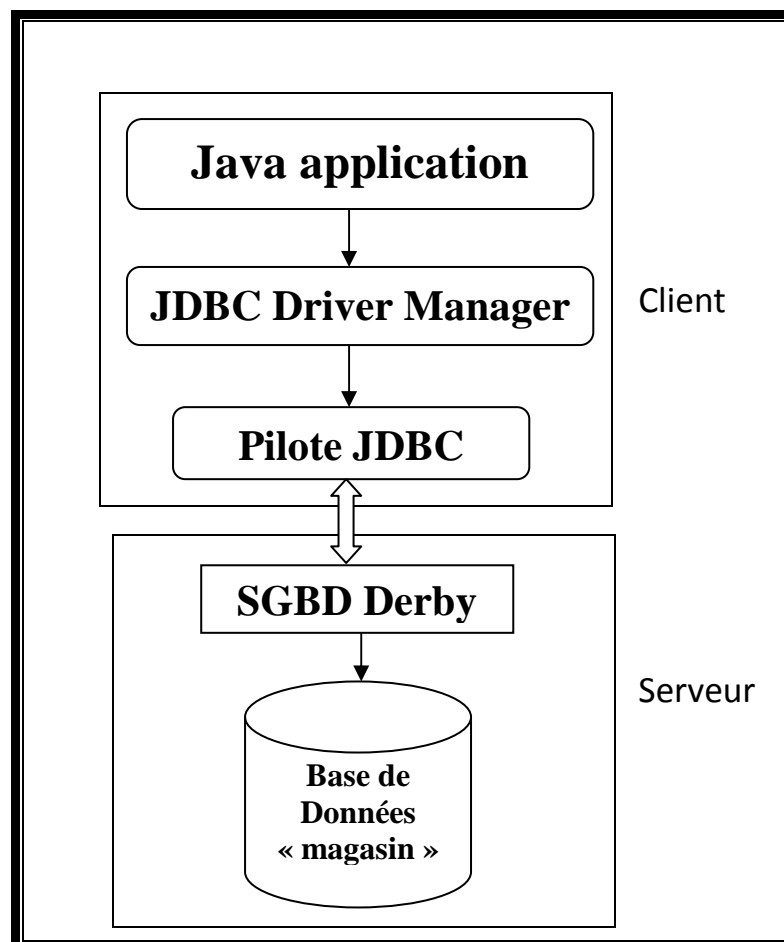


Figure III.2 : Architecture du système GML

III. Les outils utilisés pour réaliser l'application

De nos jours il existe de nombreux environnements de programmation et des pilotes pour les accès aux données, plus au moins dédiés à tel ou tel type d'applications

particulières. Parmi eux, notre choix s'est focalisé vers le langage JAVA avec l'IDE NetBeans et le pilote JDBC pour communiquer avec la base de données.

III.1. Présentation de l'environnement JAVA

❖ Qu'est ce que Java ?

Apparu fin 1995 début 1996 et développé par Sun Microsystems Java s'est très rapidement taillé une place importante en particulier dans le domaine de l'internet et des applications client-serveur.

Les objectifs de Java sont d'être multiplateformes et d'assurer la sécurité aussi bien pendant le développement que pendant l'utilisation d'un programme Java. Il est en passe de détrôner le langage C++ dont il hérite partiellement la syntaxe mais non ses défauts. Comme C++ et Delphi, Java est algorithmique et orienté objet à ce titre il peut effectuer comme ses compagnons, tout les taches d'un tel langage (bureautiques, graphiques, multimédias, base de données, environnement de développement, etc....) .son point de fort qui le démarque des autres est sa portabilité due(en théorie) à ses bibliothèques de classes indépendantes de la plate-forme ,ce qui est le point essentiel de la programmation sur internet ou plusieurs machines dissemblables sont interconnectées. [7][8]

❖ point faible de Java

- Java est jeune
- Java est gourmande en mémoire
- Java est un langage normalisé mais propriétaire (Sun)

❖ Caractéristiques

Java est un excellent candidat pour le développement d'applications de bases de données en fait il est :

- Orienté objet : Tout est classe (pas de fonctions) sauf les types primitifs (int, float, double,...) et les tableaux.
- Portable : Java supporte un code source écrit en Unicode
- Robuste : Pas d'accès direct à la mémoire.

III.2. Définition du pilote JDBC

❖ Qu'est ce que JDBC ?

JDBC est une API Java (ensemble de classes et d'interfaces défini par SUN et les acteurs du domaine des BD) permettant d'accéder aux bases de données à l'aide du langage Java via des requêtes SQL. JDBC est conçue pour que le développeur puisse se

concentrer au maximum sur son application, et perde le moins d'énergie possible pour traiter des problèmes techniques de liens avec la base de données.

❖ *Que peut-on faire avec JDBC ?*

Après le chargement d'un pilote JDBC et la définition de l'URL de connexion l'API JDBC permet de :

- 1) L'établissement d'une connexion avec le SGBD.
- 2) L'envoi de requêtes SQL au SGBD, à partir du programme java.
- 3) Le traitement, au niveau du programme, des données retournées par le SGBD.
- 4) Le traitement des métadonnées de la connexion, de l'instruction ou des résultats

III.3. Les interfaces de l'application

Pour mieux comprendre le fonctionnement de l'application, les figures qui suivent décrivent la chronologie des principales fonctionnalités de l'application.

La figure ci-dessous autorise le client et le libraire à s'introduire à toutes les fenêtres de l'application et d'accéder aux différents services (la recherche, la gestion du panier, le lancement de la commande, la mis à jour des livres), soit par le menu ou par des raccourcis buttons, c'est la fenêtre principale de notre application (figureIII.3).

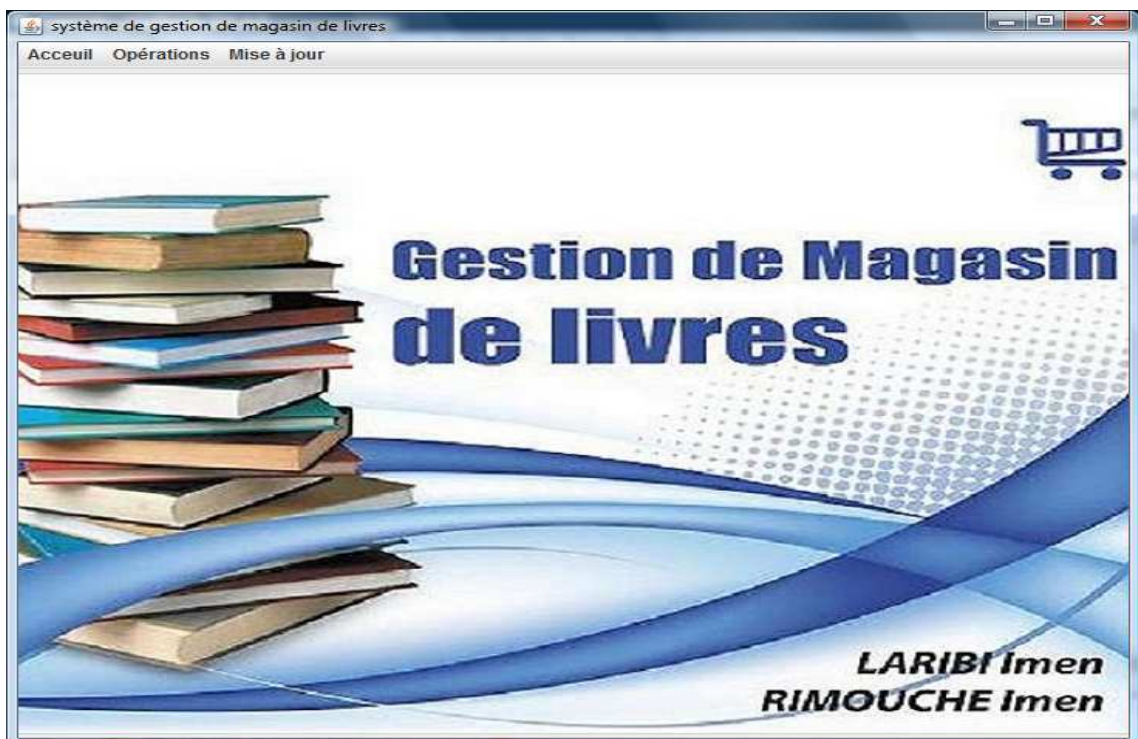
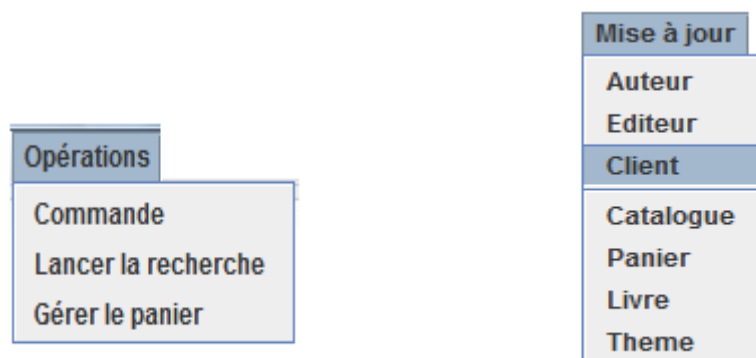


Figure III.3 : Fenêtre principale

Le menu « mis à jour » permet la mis à jour de différente table (ajouter, modifier, supprimer) et le menu « opérations » autorise l'accès aux services.



L'écran ci-dessous (la figure III.4) permet au libraire soit un ajout, une modification ou encore une suppression d'auteur.

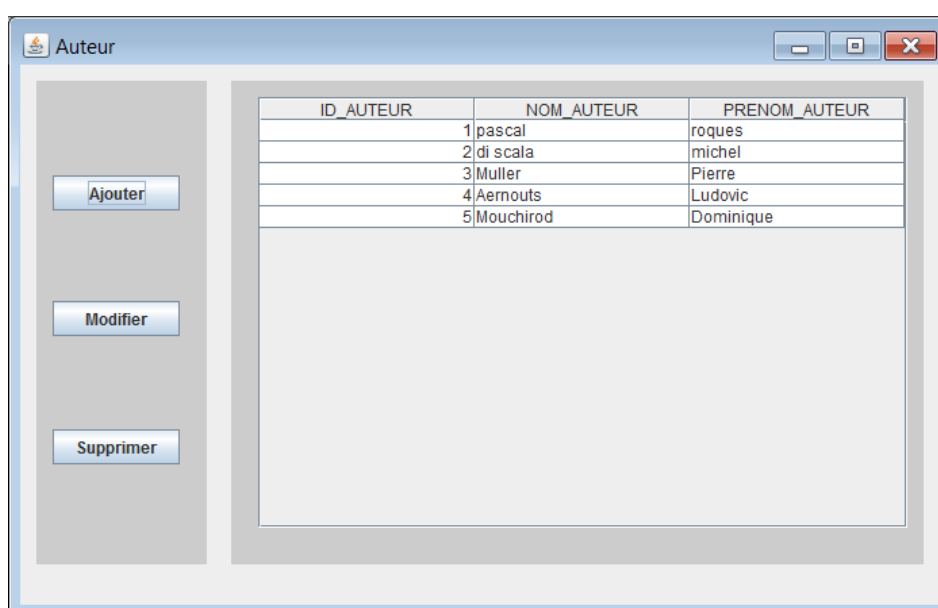
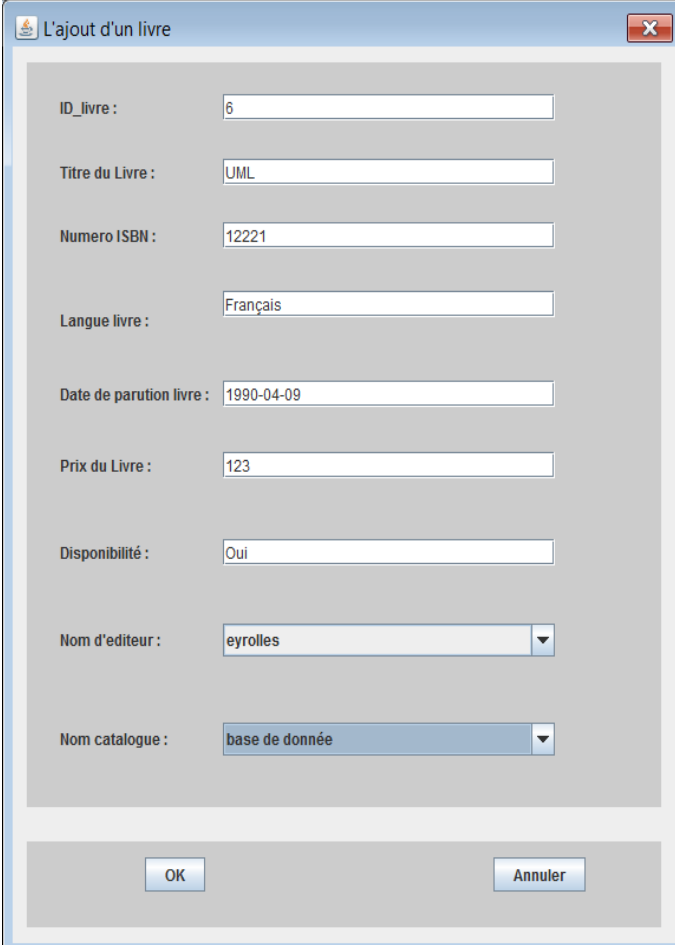


Figure III.4 : Mis à jour de l'auteur

L'écran suivant (figure III.5) nous permet de remplir les informations pour l'ajout d'un livre et de l'insérer dans la table.



L'ajout d'un livre

ID_livre : 6

Titre du Livre : UML

Numero ISBN : 12221

Langue livre : Français

Date de parution livre : 1990-04-09

Prix du Livre : 123

Disponibilité : Oui

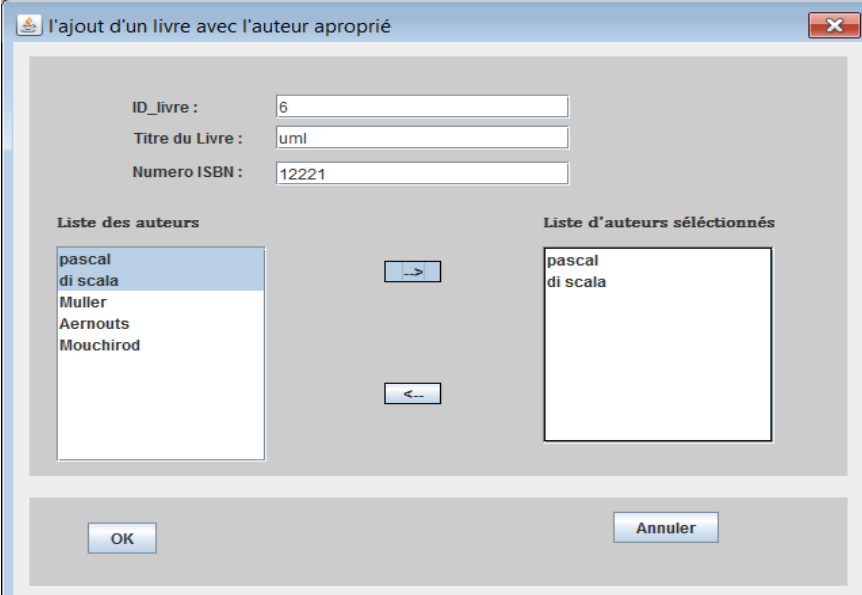
Nom d'éditeur : eyrolles

Nom catalogue : base de donnée

OK Annuler

Figure III.5 : L'ajout d'un livre.

La figure suivante (figure III.6) nous permet d'attribuer des auteurs pour chaque livres



l'ajout d'un livre avec l'auteur approprié

ID_livre : 6

Titre du Livre : uml

Numero ISBN : 12221

Liste des auteurs

- pascal
- di scala
- Muller
- Aernouts
- Mouchirod

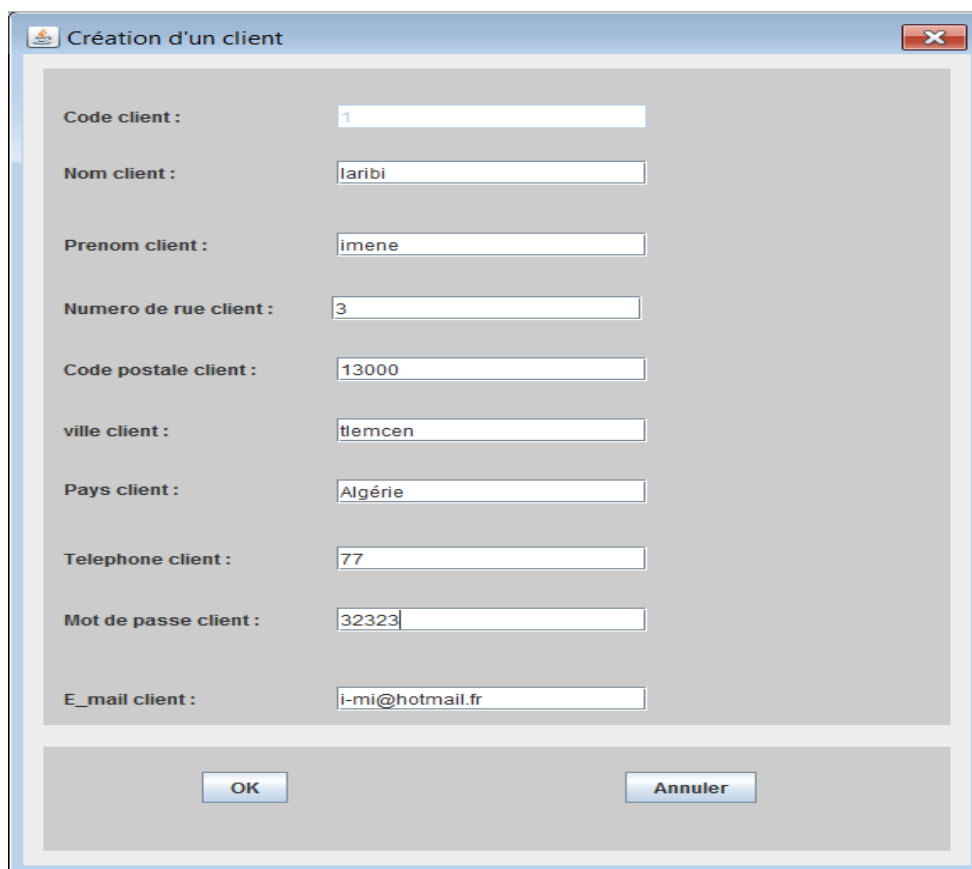
Liste d'auteurs sélectionnés

- pascal
- di scala

OK Annuler

Figure III.6 : ajout d'un livre avec l'auteur approprié

La fenêtre présentée par la figure III.7 permet de remplir des renseignements pour la création d'un client et de l'insérer dans la table.



The image shows a Windows-style dialog box titled "Création d'un client". It contains several text input fields, each with a label to its left. The fields are filled with the following text: "1", "Iaribi", "imene", "3", "13000", "tlemcen", "Algérie", "77", "32323", and "i-mi@hotmail.fr". At the bottom of the dialog, there are two buttons: "OK" on the left and "Annuler" on the right.

Code client :	1
Nom client :	Iaribi
Prenom client :	imene
Numero de rue client :	3
Code postale client :	13000
ville client :	tlemcen
Pays client :	Algérie
Telephone client :	77
Mot de passe client :	32323
E_mail client :	i-mi@hotmail.fr

Figure III.7 : création d'un client

La fenêtre ci-dessous (figure III.8) est composée de deux parties : la première contient les critères de recherche et la deuxième contient la fiche détaillée du livre recherché.

Lancement de la recherche

Catalogue : base de donnée

Champ : Titre

Recherche : java

Resultat de votre recherche

ID_LIVRE	ID_EDITEUR	ID_CATALOGUE	TITRE_LIVRE	ISBN_LIVRE	LANGUE_LIVRE	DATEDEPARU...	PRIX_LIVRE	DISPONIBILITE
2	1	3	java	4469	Français	5 août 2009	569	oui

Figure III.8 : Lancement de la recherche.

La fenêtre suivante (figure III.9) permet d'insérer le livre sélectionné dans le panier, le bouton « calculer » permet de calculer le prix total en multipliant la quantité par le prix unitaire.

Ajout au panier

Id ligne : 1

Id panier : 2

titre livre : java

Quantité ligne : 2

Prix unitaire : 569.0

Prix total : 1138

calculer

OK Annuler

Figure III.9 : Ajout au panier

La fenêtre ci-dessous (figure III.10) illustre l'état du panier où le client peut commander des livres ou vider son panier.

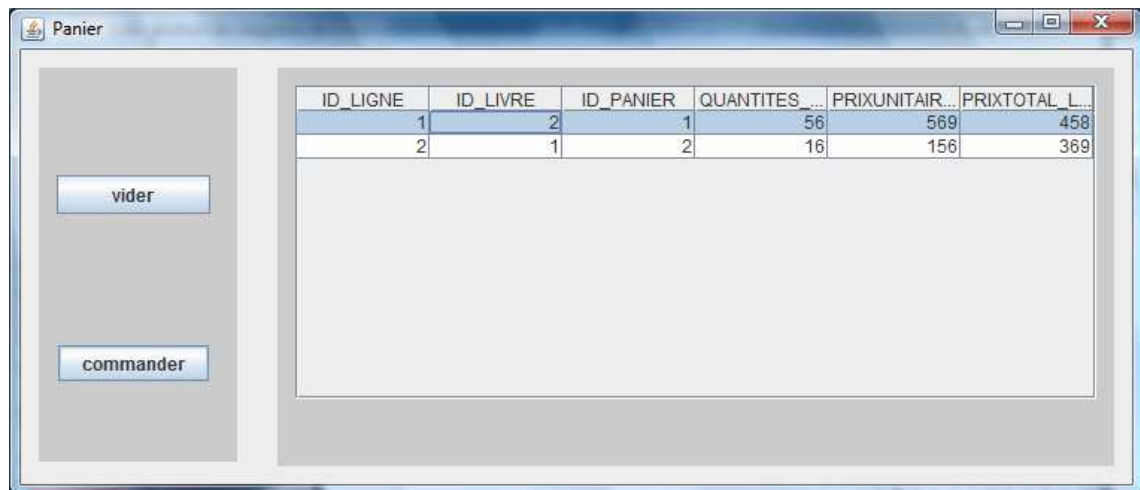


Figure III.10 : Liste des paniers.

L'écran suivant (figure III.11) présente la commande passé par le client.

The screenshot shows a window titled 'commande' with a form containing six input fields. The fields are labeled: Num commande, id panier, Date, Délai de livraison, frais de port, and Montant. The values entered in the fields are: 3, 1, 1994-07-12, 29, 15, and 458. At the bottom of the window are two buttons: 'OK' and 'Annuler'.

Figure III.11 : Commander panier.

IV. conclusion

Nous avons réalisé notre application de gestion de magasin de livres en se basant sur l'architecture 2-tiers et en utilisant le langage JAVA.

L'implémentation de cette application nécessite un environnement de développement NetBeans , l'API JDBC pour gérer les accès aux données, et la base de données « magasin » réalisé sous le SGBD Derby.