

Université : HIS
Module : Programmation Orientée Objet

Application Desktop de Gestion d'un Magasin de Peinture
--

projet tutoré présenté par :
Bouchene Mustapha
Koudri Yacine
Abdelkader Kheddaoui Abdelhak
Chehaba Choaib sofian
Benaoumeur Abdelmadjid

Sous la direction de :
prof : Soumia Zelagui

Année universitaire : 2022-2023

Table des matières

1	Introduction	2
1.1	Problématique	2
1.2	Solution Proposée	2
2	Analyse et Spécification de la Gestion	2
2.1	Objectif de l'application	2
2.2	Spécification du cahier des charges	2
2.3	Améliorations possibles	3
3	Conception de l'application	3
3.1	Diagramme de classe	3
3.2	Modèle de données	4
3.3	Interface utilisateur	5
3.4	Architecture de l'application	5
4	Développement de l'application	5
4.1	Introduction	5
4.2	Spécification logiciel	5
4.2.1	language de programmation	5
4.2.2	Outils logiciels	5
5	Présentation d'application	5
5.1	Page de connexion	5
5.2	Page d'inscription	6
5.3	page profil d'un auteur	6
5.4	page de liste des clients	7
6	Conclusion	7

1 Introduction

Ce rapport présente le travail effectué dans notre projet du module de Programmation Orientée Objet.

Au cours de notre étude, nous avons acquis de nombreuses connaissances en programmation en Java. Ce projet nous donnera l'occasion de mettre en pratique ces connaissances et de les développer.

Notre thème de projet est la création d'une application Desktop de gestion d'un magasin de peinture pour la soumission de produits, l'ajout, la modification et la suppression de produits.

Afin de rendre compte de notre projet, nous allons vous décrire les grands axes importants.

1.1 Problématique

D'après Ammi El Bachir, les problématiques sont :

- Ruptures de stock.
- Surplus de produits qui arrivent toujours.

1.2 Solution Proposée

La solution est de créer une application Desktop qui permet de faciliter la tâche d'Ammi El Bachir en offrant :

- La gestion des produits dans un tableau.
- Le codage du produit.

2 Analyse et Spécification de la Gestion

2.1 Objectif de l'application

1. **Pour l'auteur** : Cette application facilite à Ammi El Bachir la gestion de son magasin.

2.2 Spécification du cahier des charges

1. Fonctions :

- **Authentification** : Cette fonction permet à l'utilisateur de se connecter à l'application en utilisant des identifiants spécifiques pour accéder aux fonctionnalités de l'application.
- **Ajouter** : Cette fonction permet à l'utilisateur d'ajouter un nouveau produit à son inventaire de stock. L'utilisateur peut ajouter les détails du produit tels que le nom, la quantité, le code et d'autres informations pertinentes.
- **Modifier** : Cette fonction permet à l'utilisateur de modifier les informations d'un produit existant dans l'inventaire. L'utilisateur peut modifier les détails tels que la quantité, le code, le nom, etc.
- **Supprimer** : Cette fonction permet à l'utilisateur de supprimer un produit de l'inventaire de stock.
- **Actualiser** : Cette fonction permet à l'utilisateur de mettre à jour les informations de l'inventaire de stock. L'utilisateur peut actualiser les informations en récupérant les données les plus récentes de la base de données.
- **Produits hors stock** : est une fonctionnalité de l'application qui permet à l'utilisateur de consulter les produits qui ne sont actuellement pas disponibles dans le stock de magasin.

- **Client** : Cette fonction permet à l'utilisateur de visualiser les informations sur les clients qui ont acheté des produits de l'inventaire. L'utilisateur peut voir les détails tels que le nom, prénom, code client, Numero Telephone.
- **La recherche par mots-clés** : est une méthode de recherche qui permet à l'utilisateur de trouver des informations en tapant des termes spécifiques liés à sa recherche. Les résultats de la recherche peuvent inclure tous les éléments qui contiennent les mots-clés spécifiés, indépendamment de leur emplacement ou de leur catégorie.
- **La recherche par nom** : est une méthode de recherche qui permet à l'utilisateur de trouver des informations en saisissant le nom d'un élément spécifique. Les résultats de la recherche sont limités aux éléments qui portent exactement le nom saisi par l'utilisateur. Cette méthode est souvent utilisée lorsque l'utilisateur sait déjà ce qu'il recherche et souhaite le trouver rapidement sans avoir à parcourir une liste d'éléments.

2.3 Améliorations possibles

Pour améliorer l'application dans le futur, nous pourrions envisager les améliorations suivantes :

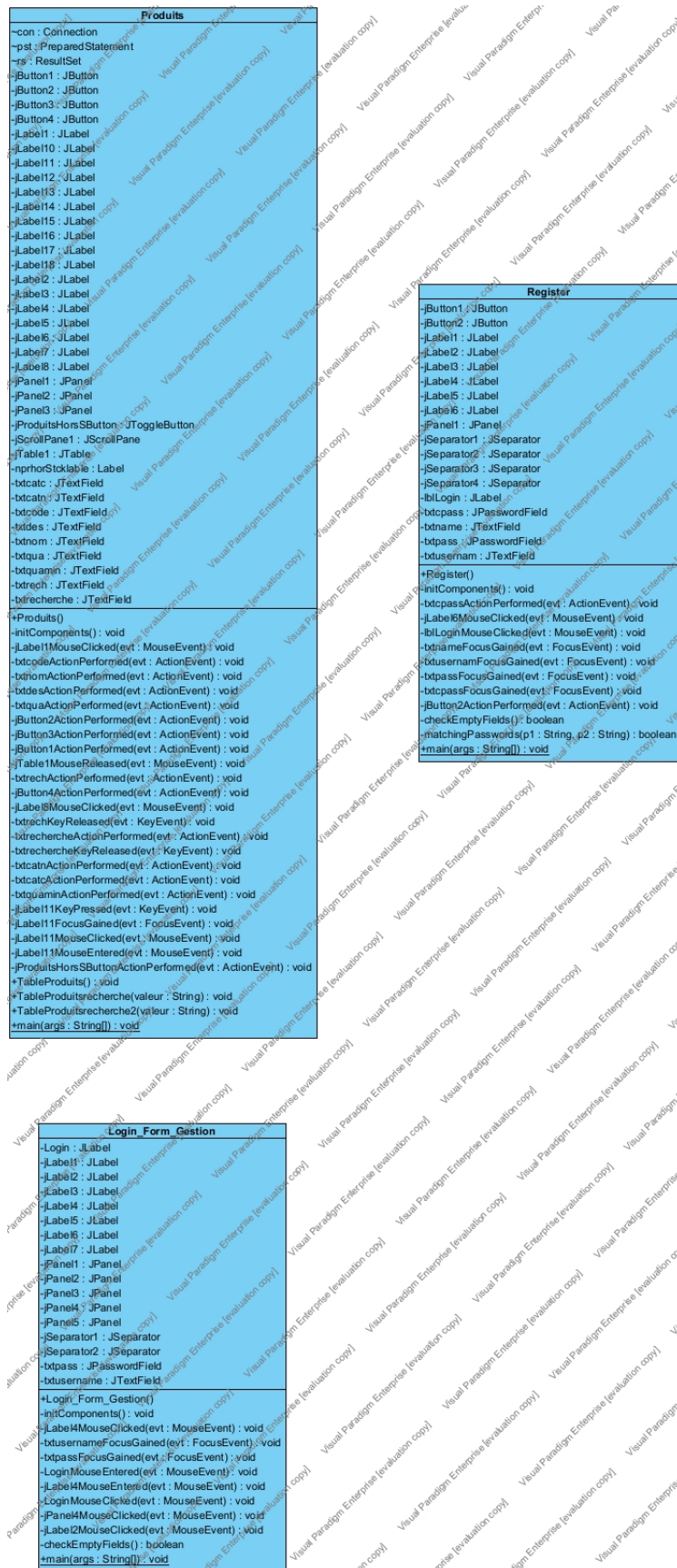
- Ajouter un module de gestion de stock pour suivre les niveaux de stock et éviter les ruptures ou les surplus ;
- Ajouter un module de gestion de clients et de facturation pour permettre aux clients de passer des commandes et de générer des factures directement depuis l'application ;
- Ajouter un module de reporting pour générer des rapports sur les ventes, les stocks, etc. ;
- Améliorer l'interface utilisateur pour la rendre plus conviviale et intuitive ;
- Ajouter des fonctionnalités de sécurité telles que la gestion des utilisateurs et les autorisations d'accès.

3 Conception de l'application

Dans cette section, nous allons décrire les différentes parties de la conception de l'application.

3.1 Diagramme de classe

Le diagramme de classe ci-dessous montre les principales classes de l'application et leurs relations :



3.2 Modèle de données

Le modèle de données de l'application se compose de deux entités principales : les produits et l'auteur. Chaque produit est décrit par un identifiant unique, un nom, une description, une quantité

disponible et un prix. L'auteur est décrit par un identifiant unique, un nom d'utilisateur, un mot de passe.

3.3 Interface utilisateur

L'interface utilisateur de l'application est conçue pour être simple et intuitive. La page d'accueil comprend un formulaire de connexion pour permettre à l'utilisateur de se connecter à l'application. Une fois connecté, l'utilisateur est redirigé vers la page principale, qui affiche la liste des produits disponibles et permet à l'utilisateur de les gérer (ajouter, modifier ou supprimer un produit).

3.4 Architecture de l'application

L'application est conçue selon une architecture client-serveur. Le serveur héberge la base de données et fournit des API pour permettre aux clients d'interagir avec la base de données. Le client est une application desktop qui utilise les API pour effectuer des opérations sur la base de données.

4 Développement de l'application

4.1 Introduction

Pour réaliser l'application, il faut adapter un environnement logiciel adéquat pour faire un bon travail

4.2 Spécification logiciel

4.2.1 langage de programmation

Le Java est le langage de base pour apprendre à créer des applications de desktop.

MySQL est un système de gestion de bases de données relationnelles, on l'a choisi car il est le plus utilisé

4.2.2 Outils logiciels

Netbeans Plateforme de programmation en Java et aussi pour le développement des applications de desktop, on a choisi ce logiciel car il est facile à coder et il permet de modifier le design de l'application plus facilement.

SQLite studio plateforme pour gérer la base de données, on a choisi ce logiciel car il est facile de relier la base de données avec le programme

Visual Paradigm est un outil de modélisation visuelle qui permet de créer des diagrammes de classe, de séquence, d'activité, de cas d'utilisation, etc. Il est utilisé pour la conception de logiciels et de systèmes d'information.

5 Présentation d'application

dans cette partie, on va présenter le développement de l'application en se basant sur les captures d'écran

5.1 Page de connexion

L'utilisateur a plusieurs choix dans la Page de connexion, le choix s'inscrire ou de connecter pour utiliser les fonctionnalités de l'application.

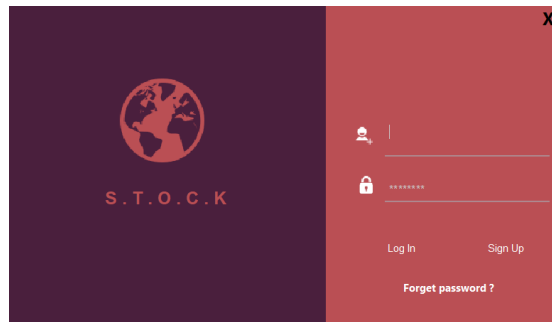


FIGURE 1 – Capture d'écran de la Page de connexion

5.2 Page d'inscription

Après que l'utilisateur remplit tous les champs, s'il clique sur "Register", il va être redirigé vers une autre page d'accueil, l'accueil de l'utilisateur connecté.

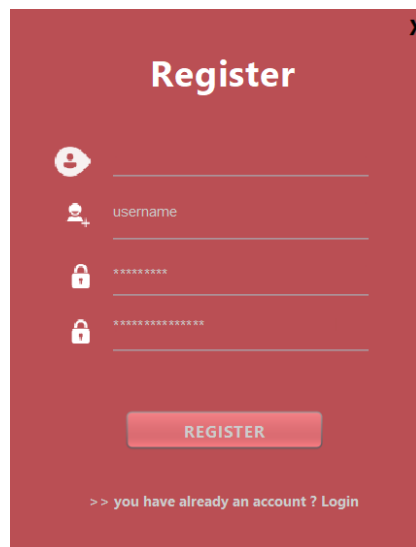


FIGURE 2 – Capture d'écran de la page d'inscription

5.3 page profil d'un auteur

Cette page contient plusieurs options pour l'auteur qui peut faire :

- ajouter un nouveau produit à son inventaire de stock.
- modifier les informations d'un produit existant dans l'inventaire.
- supprimer un produit de l'inventaire de stock.
- mettre à jour les informations de l'inventaire de stock.
- visualiser l'inventaire de stock existant dans l'application.
- visualiser les informations sur les clients qui ont acheté des produits de l'inventaire.
- La recherche par mots-clés et par nom.
- consulter les produits qui ne sont actuellement pas disponibles dans le stock de magasin.

CODE	NOM	DESCRIPTION	QUANTITE	QUANTITE_MIN	NOM_CATEGORIE	CODE_CATEGORIE
1	PeintBlanche	Peinture de q...	100	1	Intérieures	INT001
12	PeintRouge	Peinture de q...	50	12	Intérieures	INT001
14	PeintJaune	Peinture de q...	100	20	Intérieures2	INT002
55	PeintMétal	Peinture spéci...	500	100	Extérieures	EXT001
555	PeintRouges	pour les surfa...	50	12	Extérieures2	EXT002
605	PeintVerre	Peinture spéci...	30	20	Spéciales	SP001
2005	PeintRouges...	pour les surfa...	500	200	Spéciales	SP001
5455	PeintCarrelage	pour les surfa...	120	52	Extérieures2	EXT002
20055	PeintBois	pour les surfa...	7	30	Extérieures	EXT001

FIGURE 3 – Capture d’écran de profil d’un auteur

5.4 page de liste des clients

La page de la liste des clients est une fonctionnalité de l’application qui permet à l’utilisateur de consulter la liste de tous les clients enregistrés dans la base de données. Cette page affiche toutes les informations relatives à chaque client, y compris leur nom, leur prenom, leur numéro de téléphone, leur code. L’auteur peut également effectuer une recherche spécifique de clients en entrant un nom dans la barre de recherche.

CODE CLIENT	NOM	PRENOM	NUMTEL
5454	Brahim	Alouch	0777663311

FIGURE 4 – Capture d’écran de profil d’un auteur

6 Conclusion

Ce rapport a décrit le travail effectué dans le cadre de notre projet de programmation orientée objet.

Nous avons présenté notre objectif, notre analyse et spécification des besoins, notre conception de l’application et ses améliorations possibles. Nous avons également présenté notre diagramme de classe et de package, notre modèle de données, notre interface utilisateur et l’architecture de l’application. Grâce à ce projet, nous avons pu mettre en pratique les connaissances que nous avons acquises en programmation et nous avons appris de nouvelles compétences en matière de conception d’applications.