

Rapport Mini Projet Java :

GESTION D'UNE PÂTISSERIE

Réalisé par:

Sahar OUALI

Amina LACHHEB

Oumaima MILED

Table des matières

Introduction générale	1
1 Contexte général	3
1.1 Cadre général du projet	4
1.2 Étude de l'existant	4
1.3 Solution Proposée	5
2 Analyse des besoins et conception	6
2.1 Analyse des besoins	7
2.1.1 Besoins fonctionnels	7
2.1.2 Besoins non fonctionnels	7
2.1.3 Identification des acteurs	8
2.2 Modélisation des besoins fonctionnels	8
2.2.1 Diagramme de cas d'utilisation global	8
2.3 Diagramme de classe	9
3 Réalisation	11
3.1 Environnement de travail	12
3.1.1 Environnement matériel	12
3.1.2 Environnement logiciel	12
3.2 Réalisation des interfaces	13
3.2.1 Authentification	13
3.2.2 Espace Administrateur	14
3.2.3 Espace Gestionnaire	15
3.2.4 Espace Caissier	16
Conclusion générale	19

Table des figures

2.1	Diagramme de cas d'utilisation global	9
2.2	Diagramme de classes	10
3.1	Authentification	13
3.2	Interface d'ajout d'un utilisateur	14
3.3	Interface d'affichage des utilisateurs	14
3.4	Interface de la recherche	15
3.5	Interface d'ajout d'un produit	15
3.6	Interface d'afficher les produits	16
3.7	Interface d'accueil	16
3.8	Interface d'ajout d'un commande	17
3.9	Interface d'effectuer une vente	17
3.10	Interface d'imprimer une facture	18

Liste des tableaux

2.1 Identification des acteurs et leurs rôles 8

Introduction générale

La croissance rapide des technologies de l'information et de la communication ainsi que l'innovation dans les systèmes numériques sont à l'origine d'une révolution provoquée par le développement des techniques numériques, principalement l'informatique et l'Internet. Cette révolution, dite " révolution numérique " a incité les sociétés des deux secteurs publique et privé à s'informatiser et par conséquent d'établir des pâtisserie.

Une pâtisserie contient un ensemble des produits ,commandes et employées . En effet, la quantité des produits à gérer, leur éclatement au sein de l'entreprise souvent très étendue dans l'espace, les exigences de performance et de réactivité font que la gestion de pâtisserie est devenue un processus global, complet et indispensable. La gestion du pâtisserie recouvre non seulement la fonction d'inventaire de ces éléments (produits , employées) mais aussi celles concernant son suivi et son évolution.

Ces opérations peuvent être effectuées par une personne qualifiée, mais bien souvent ce travail dépasse ses compétences. Pour pallier à cela, il est nécessaire qu'un ou plusieurs outils soient mis en place au sein de la pâtisserie afin d'avoir un suivi régulier et parfois anticiper les défaillances de ses ressources.

C'est dans ce cadre que s'inscrit notre projet, dans lequel nous sommes amenés à implémenter une méthodologie pour instaurer une bonne gestion du pâtisserie.

Ce rapport est organisé en trois chapitres qui décrivent les différentes étapes par lesquelles nous sommes passés pour effectuer notre projet :

- Le premier chapitre « Contexte général » placera le projet dans son contexte en présentant l'objectif et en précisant l'étude de l'existant ainsi que le travail demandé.
- Le deuxième chapitre « Conception et spécification des besoins » comportera la conception de notre application, basée sur le langage de modélisation Unified Modelling Language UML et identifiera les besoins fonctionnels et non fonctionnels attendus du projet, ainsi que la modélisation de ces besoins par le recours aux diagrammes de cas d'utilisation.

- Le troisième chapitre « Réalisation » sera consacré à la description de la phase de l'implémentation de notre système. Avant de décrire quelques captures d'écran de notre application nous allons définir l'environnement matériel et logiciel utilisé au cours de cette phase.

Pour finir, une conclusion générale de tout le rapport sera nécessaire dans laquelle nous proposerons les éventuelles améliorations susceptibles d'être ajoutées ultérieurement.

CONTEXTE GÉNÉRAL

Plan

1	Cadre général du projet	4
2	Étude de l'existant	4
3	Solution Proposée	5

Introduction

Dans ce chapitre, nous commençons par présenter le sujet que nous a été affecté, l'étude de l'existant et nous terminons par exposer la solution proposée.

1.1 Cadre général du projet

Notre projet consiste à développer une application qui permet de gérer une pâtisserie en Java. L'application va devoir permettre de gérer les ventes, les produits, les employées, les commandes des gâteaux et le suivi d'activité...

1.2 Étude de l'existant

L'étude de l'existant est une phase importante dans le cycle de développement de l'application afin de bien se positionner sur le marché.

L'existant présente beaucoup de défaillances tels que les informations circulant dans une pâtisserie sont de nature diverses : les employées, le stock, les achats, les fournisseurs...

La gestion de stock et de commande deviennent de plus en plus dure. Les erreurs sont fréquemment commises, avec le nombre d'informations à sauvegarder qui augmente de jour en jour, de plus avec la procédure manuelle et faute de bon archivage des documents ces derniers sont facilement perdus, et la recherche des informations n'est pas chose évidente ni facile.

Tous ces points faibles ont un impact négatif sur le mécanisme de ce processus, et par la suite dégradent la performance. Suite à ces conditions, notre projet a été une occasion pour faciliter cette tâche et automatiser le processus de la gestion d'une pâtisserie.

Les insuffisances constatées peuvent être récapitulées comme suit :

- Stocker les produits
- Passer les commandes d'une manière automatique
- Faire l'achat et la vente
- Stocker les informations des employées
- .

1.3 Solution Proposée

Notre solution proposée consiste à gérer la pâtisserie d'une manière informatisée avec une application qui permet aux utilisateurs d'assurer ainsi une bonne gestion. En effet on aura un administrateur de système qui a une idée sur tout les employés du pâtisserie .

L'objectif essentiel de ce travail se résume à la conception et la réalisation d'une application pour la gestion d'une pâtisserie, une application fiable et maniable afin de faciliter la tâche de gestion de stock, des commandes et des fournisseurs. Ainsi que d'améliorer la qualité du service en un temps très réduit et surtout limiter les erreurs de stockage des informations.

Conclusion

Ce chapitre nous a servi à mettre le projet dans son cadre. Les défaillances constatées au niveau d'une pâtisserie nécessitent la mise en place d'une solution de gestion d'une pâtisserie par une application java pour gérer les produits , les commandes et l'action de la vente d'une manière automatique. Nous entamons dans le chapitre suivant une Conception et spécification des besoins.

ANALYSE DES BESOINS ET CONCEPTION

Plan

1	Analyse des besoins	7
2	Modélisation des besoins fonctionnels	8
3	Diagramme de classe	9

Introduction

L'ingénierie des besoins est l'une des activités les plus importantes puisqu'elle permet de déterminer les besoins de l'utilisateur de l'application et définir les contraintes envisagées. Dans ce chapitre, nous étudions les exigences de notre solution d'une manière plus formelle et plus spécifique en présentant en premier lieu les besoins fonctionnels et non fonctionnels et en second lieu en spécifiant les différents cas d'utilisation de notre application.

2.1 Analyse des besoins

Nous allons identifier dans cette partie les acteurs et leur rôles. Ensuite, nous allons expliquer les besoins fonctionnels et non fonctionnels.

2.1.1 Besoins fonctionnels

Les besoins fonctionnels représentent le comportement du système ainsi que les services offerts aux différents intervenants. Le système permet :

A l'administrateur de

- S'authentifier : saisir son login et son mot de passe.
- Gérer les employés : ajouter, consulter, modifier et supprimer des utilisateurs.

Au gestionnaire de :

- S'authentifier : saisir login et mot de passe.
- Gérer produits : consulter, modifier et supprimer des tickets.

Au caissier de :

- S'authentifier : saisir nom d'utilisateur et mot de passe.
- Passer une commande d'un gâteaux.
- Faire l'action de vente d'une manière automatique .

2.1.2 Besoins non fonctionnels

Le système doit répondre à certains besoins qui ne sont pas indispensables pour son fonctionnement mais qui sont importants car ils agissent de façon indirecte sur le résultat, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

- Ergonomie de l'application : L'application doit présenter des interfaces simples.
- Fiabilité : L'application doit fonctionner de façon cohérente sans erreurs.

- Rapidité : L'application doit être rapide.
- Sécurité : Une authentification est exigée lors du démarrage pour accéder, effectuer les opérations désirées et gérer des droits pour les utilisateurs.

2.1.3 Identification des acteurs

Un acteur définit un rôle qu'une entité extérieure assume lors de son interaction avec le système. Les acteurs qui interagissent avec le système sont :

Tableau 2.1: Identification des acteurs et leurs rôles

Acteurs	Rôles
Administrateur	<ul style="list-style-type: none">— Ajouter, consulter, modifier et supprimer les employées .
Gestionnaire	<ul style="list-style-type: none">— Consulter, ajouter, modifier et supprimer les produits.
Cassier	<ul style="list-style-type: none">— Passer une commande .— Faire la vente .

2.2 Modélisation des besoins fonctionnels

2.2.1 Diagramme de cas d'utilisation global

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet.

Nous présentons, dans la figure 2.1, le diagramme de cas d'utilisation général de notre application précisant tous les besoins et les rôles de chaque acteur.



Figure 2.1: Diagramme de cas d'utilisation global

2.3 Diagramme de classe

Cette partie présente une modélisation de la vue statique de notre projet. Elle s'intéresse en général à la structure statique du système par l'intermédiaire des objets, des classes, des attributs, des opérations et des relations. Nous décrivons cette vue par le biais d'un diagramme de classes.

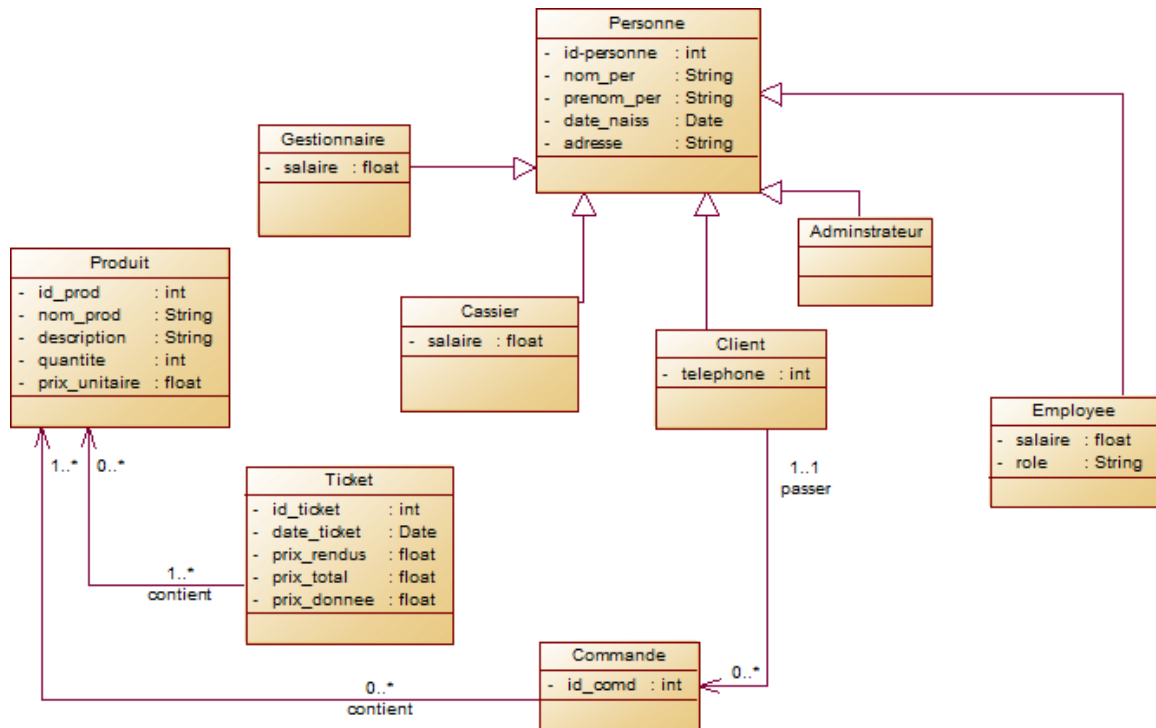


Figure 2.2: Diagramme de classes

Conclusion

Dans ce chapitre, nous avons présenté l'analyse des besoins fonctionnels et non fonctionnels. Aussi, nous avons défini les acteurs puis nous avons détaillé les cas d'utilisations. D'autre part, nous avons mis en place la méthodologie Scrum et la planification des sprints. Cette étude facilitera la tâche de la conception que nous détaillerons dans ce chapitre, et lorsque la phase de conception est terminée, nous sommes prêts donc à entamer la réalisation de notre projet, cette implémentation sera détaillée dans le prochain chapitre.

RÉALISATION

Plan

1	Environnement de travail	12
2	Réalisation des interfaces	13

Introduction

Après avoir achevé la phase conceptuelle, nous passons maintenant à la phase d'implémentation de la solution envisagée. De ce fait, nous consacrons ce chapitre à mettre en oeuvre les environnements matériels et logiciels dans lesquels nous avons réalisé notre travail. Ensuite, nous montrons quelques interfaces graphiques réalisées à travers des captures d'écran.

3.1 Environnement de travail

Nous décrivons dans cette section les environnements matériel et logiciel adoptés pour l'implémentation de notre application.

3.1.1 Environnement matériel

Nous mettons l'accent sur l'environnement matériel utilisé pour la réalisation de notre application.

Au cours du développement de notre projet nous avons utilisé un ordinateur portable ASUS ayant comme configuration : Un système d'exploitation Windows 10, un processeur Intel(R) Core(TM) i5 CPU @ 2.2GHz , un disque dur de 1 To, et une mémoire vive de 8 Go.

3.1.2 Environnement logiciel

Nous mettrons l'accent dans ce paragraphe sur les différents outils logiciels utilisés :

Environnement de développement intégré (IDE) : Netbeans

NetBeans est un environnement de développement intégré (EDI), placé en open source. Compilé en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)).

Base de données : XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local. XAMPP est très pratique pour créer un serveur local de développement : il fournit au néophyte un serveur de test en quelques clics, sans avoir à choisir ni à connaître les différents services à

installer.

Il est multi-plateforme et existe donc pour Linux, Windows et OS X

Modélisation UML :PowerDesigner

PowerDesigner (anciennement PowerAMC) est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

3.2 Réalisation des interfaces

Dans cette partie nous allons présenter à travers un enchaînement de captures d'écran, les principales interfaces donnant un aperçu général sur les différentes fonctionnalités de l'application.

3.2.1 Authentification

Afin de profiter des fonctionnalités de la plateforme, l'utilisateur doit tout d'abord s'authentifier. Pour ce fait, il doit nécessairement saisir son nom, ainsi que son mot de passe comme le montre la figure 3.1

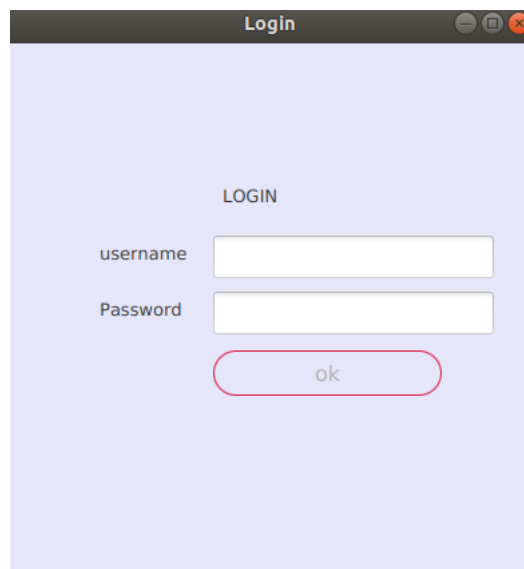


Figure 3.1: Authentification

3.2.2 Espace Administrateur

Interface d'ajout d'un utilisateur

La figure 3.2 montre la page d'ajout d'un utilisateur à notre application.

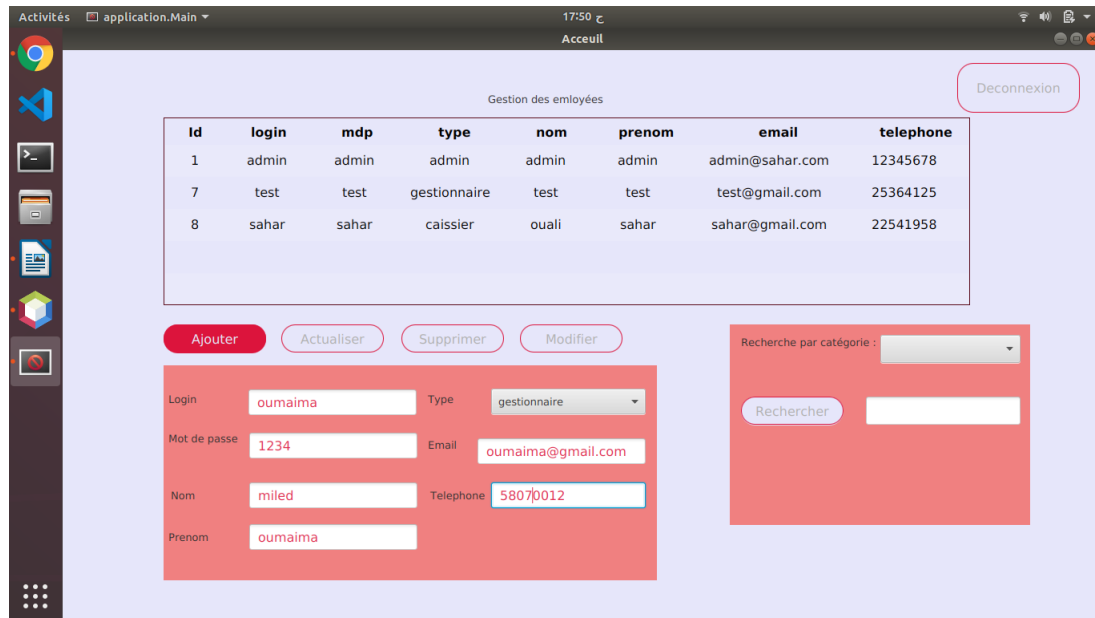


Figure 3.2: Interface d'ajout d'un utilisateur

Interface d'affichage des utilisateurs

La figure 3.3 montre la page d'affichage des utilisateurs à notre application.

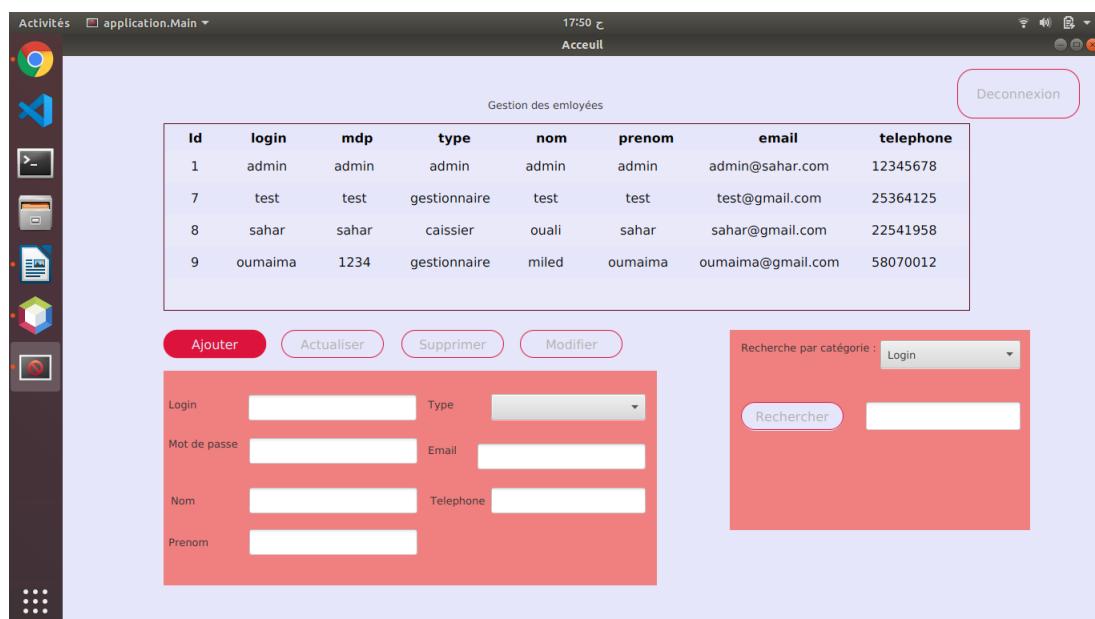


Figure 3.3: Interface d'affichage des utilisateurs

Interface de la recherche selon un critère

La figure 3.4 montre la recherche d'un utilisateur par un type (par exemple 'gestionnaire')

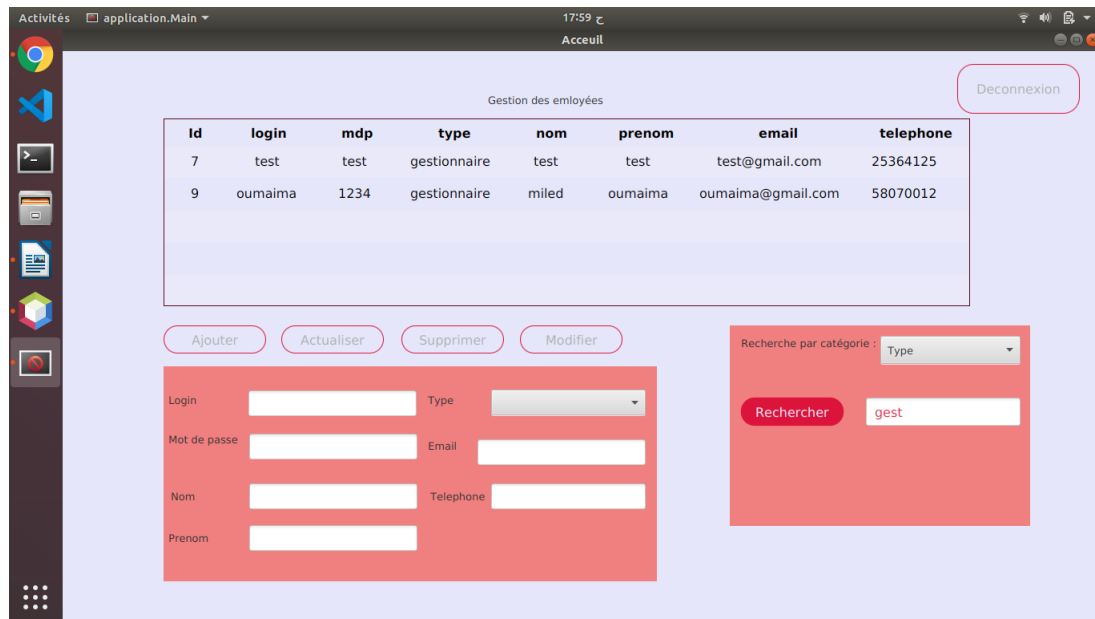


Figure 3.4: Interface de la recherche

3.2.3 Espace Gestionnaire

Interface d'ajout d'un produit

La figure 3.5 montre la page d'ajout un produit



Figure 3.5: Interface d'ajout d'un produit

Interface d'afficher les produits

La figure 3.6 montre la page d'affichage des produits



Figure 3.6: Interface d'afficher les produits

3.2.4 Espace Caissier

Interface d'accueil

La figure 3.7 montre la page d'accueil d'un caissier.

Elle permet au caissier de passer une commande des gâteaux et de passer une vente des produits.



Figure 3.7: Interface d'accueil

Interface d'ajout une commande

La figure 3.8 montre la page d'ajout une commande.

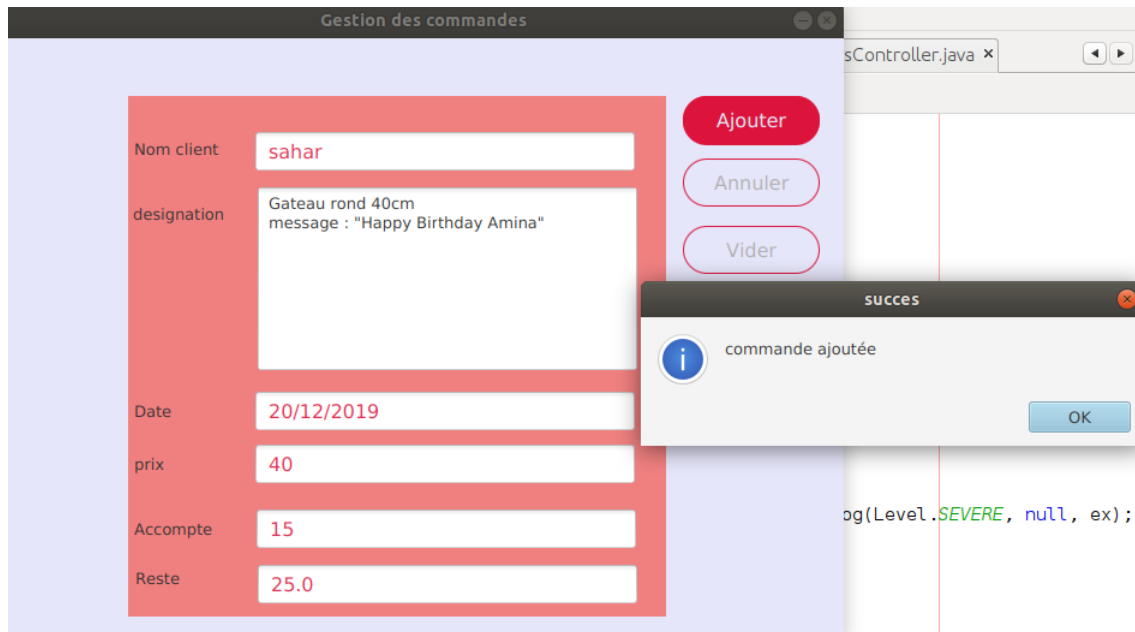


Figure 3.8: Interface d'ajout d'un commande

Interface d'effectuer une vente

La figure 3.9 montre la page d'effectuer une vente

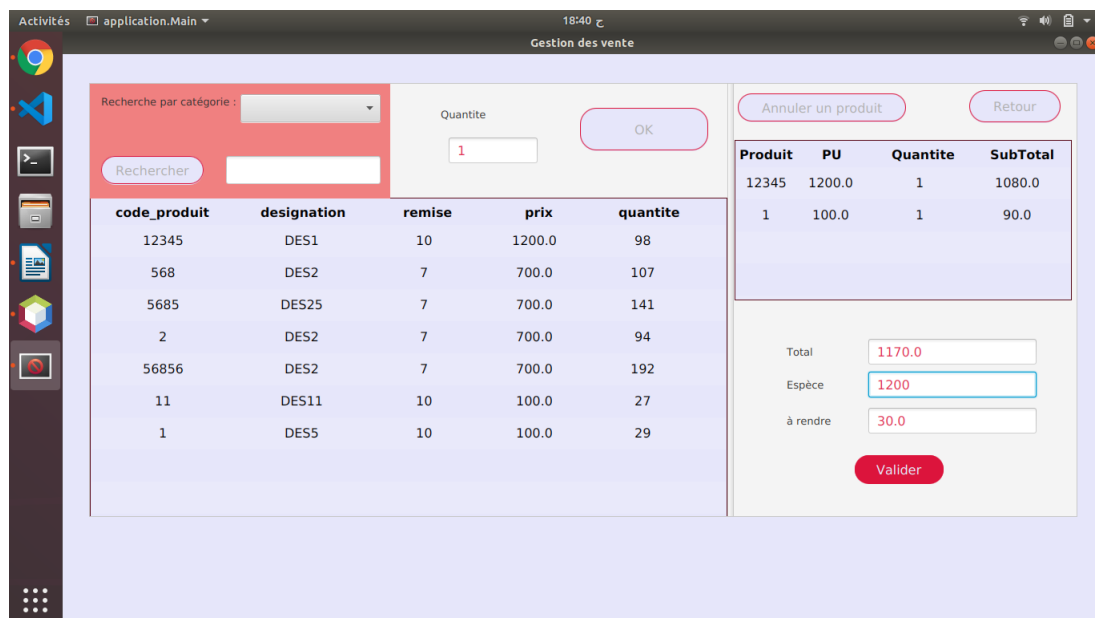


Figure 3.9: Interface d'effectuer une vente

Interface d'imprimer une facture

La figure 3.10 montre la page d'imprimer une facture

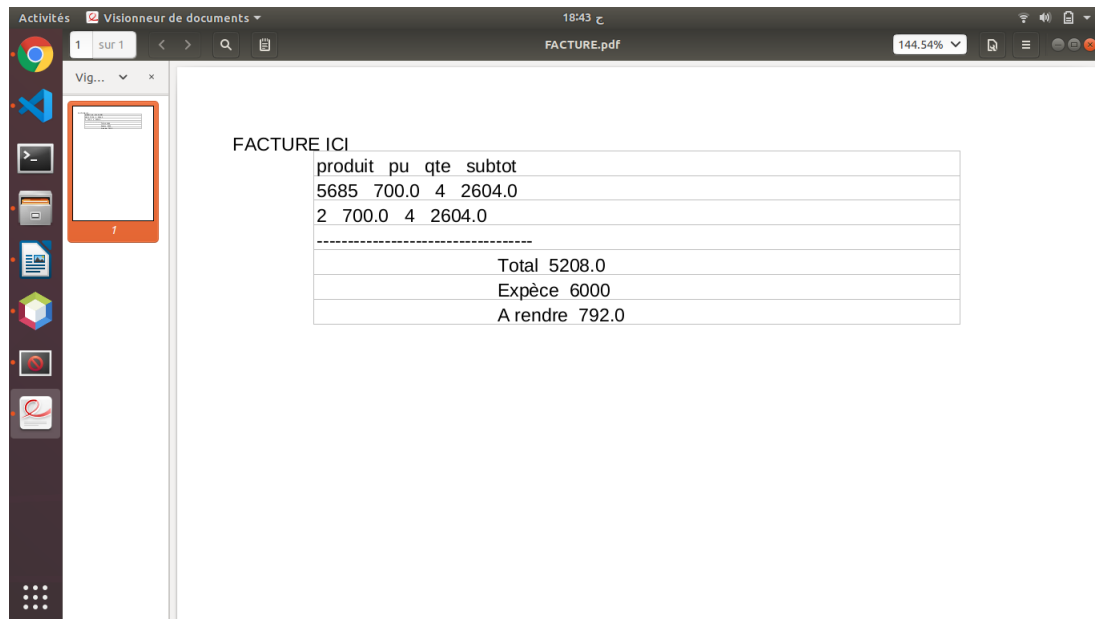


Figure 3.10: Interface d'imprimer une facture

Conclusion

Dans ce dernier chapitre, nous avons présentés les outils qu'on a utilisés. Puis, nous avons détaillé le travail réalisé en se basant sur les principales interfaces de notre application.

La conclusion générale retenue permettant de clôturer ce travail seront l'objet de la partie suivante.

Conclusion générale

Ce projet, a été, sous plusieurs aspects riches d'enseignements. Nous avons commencé dans un premier lieu par comprendre le contexte général de notre application et identifier les différentes exigences de notre futur système. Nous avons préparé par la suite notre planning de travail en respectant les priorités de nos besoins.

La première étape a été de présenter le cadre général en dégagant le contexte général et exposant l'étude de l'existant. Nous avons ensuite entamé dans le deuxième chapitre l'étude des besoins au bout de laquelle nous avons abouti à une définition précise des différentes fonctionnalités attendues de la solution envisagée à travers la description des besoins et des différents cas d'utilisation à envisagés. Partant de cette spécification, nous nous sommes intéressés à la description détaillée des différents modules à l'aide du diagramme de classes selon le formalisme UML. Nous avons par la suite concrétisé notre travail par la phase de réalisation au cours de laquelle nous avons présenté l'environnement matériel et logiciel puis nous avons exposé les différentes fonctionnalités mises en place à travers des imprimés écrans d'une application construite en se basant sur notre solution.

Ce travail nous a été très instructif de point de vue des connaissances acquises. Il nous a procuré une opportunité pour, d'une part aborder un domaine métier et d'autre part confirmer une fois de plus nos connaissances dans le développement orienté objet.

Par ailleurs, d'un point de vue technologique, le projet a été très enrichissant puisqu'il nous a donné l'occasion d'étudier et d'utiliser une panoplie de technologies java , netbeans et xampp.

Grâce à son caractère évolutif, notre projet a pu supporter d'autres fonctionnalités tel qu'un module de gestion des stocks des matière primaire et des fournisseurs .

Pour conclure, nous souhaitons que ce modeste travail apporte la satisfaction.