



Projet de Fin d'Etudes

Licence Sciences et Techniques Génie Informatique

Réalisation d'une Application de Gestion de Vente et de Stock de la Société Dar du Commerce Moderne



Lieu de stage : DAR DU COMMERCE MODERNE

Réalisé par :

TAMIMI Soufiane
CHABLI Bilal

Encadré par :

Pr. ZENKOUAR Khalid

Soutenu le 18/06/2015 devant le jury composé de :

Pr. ZENKOUAR Khalid

Pr. A. Begdouri

Pr. R. Benabbou



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَقُلْ أَطِيعُوا اللَّهَ وَأَطِيعُوا الرَّسُولَ
وَأَطِيعُوا أَرْوَاقَ الْإِسْلَامِ وَأَطِيعُوا
أَرْوَاقَ الْمَدِينَةِ وَأَطِيعُوا أَرْوَاقَ
الْمَدِينَةِ وَأَطِيعُوا أَرْوَاقَ الْمَدِينَةِ

اللَّهُ
صَلَّى
الْعَظِيمُ



SOMMAIRE

Sommaire	3
Dédicaces.....	5
Remerciement	6
Liste des Figures.....	7
Introduction Générale.....	8
CHAPITRE I : CONTEXTE GÉNÉRAL DU PROJET	9
Introduction :	10
1. présentation du lieu de stage :	10
2. Présentation du projet :.....	10
2.1. Cahier de charge :	10
2.2. Enoncé du problème :	12
2.3. Solutions proposées :	13
Conclusion :	14
CHAPITRE II : ANALYSE FONCTIONNELLE.....	15
Introduction :	16
1. Méthodologies d'analyse:.....	16
1.1. Le langage UML :	16
1.2. Le Modèle Incrémental et Itératif :	16
1.2.1. Avantage :	18
1.2.2. Inconvénients :	18
1.2.3. Incréments principal du projet	18
1.3. Le Modèle MVC (Modèle-Vue-Contrôleur) :	18
1.3.1. Définition MVC (Modèle-Vue-Contrôleur) :	19
1.3.2. Avantages du MVC :	20



UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES FES
DÉPARTEMENT D'INFORMATIQUE



2. Modélisation du contexte :	21
2.1. Les acteurs et leurs rôles :	21
2.2. Les messages émis et reçus :	22
3. Analyse et conception:	24
3.1. Diagramme de package :	24
3.2. Diagrammes des cas d'utilisation :	25
3.3. Diagrammes de séquences :	27
3.4. Diagramme de classe :	30
Conclusion :	31
CHAPITRE III : PRÉSENTATION DE L'APPLICATION	32
1. Outils et technologies de développement :	33
1.1. Langages utilisés :	33
1.1.1. C++ :	33
1.1.2. Qt :	34
1.1.3. SQLite :	36
1.2. Logiciels utilisés :	37
1.2.1. Qt Creator :	37
1.2.2. Entreprise Architect :	38
1.2.3. Inno Setup Compiler :	38
2. Schéma général de l'application :	39
3. Présentation de l'application :	40
Conclusion :	58
Conclusion générale	59
Webographie	60
Bibliographie	61



DÉDICACES

*Avant toute chose, nous tenons à dédier
notre travail à nos chers parents, avec amour et
respect vous avez su nous donner la force pour aller
aussi loin.*

*Nous dédions ce travail à notre famille, elle
qui a toujours été à nos côtés pour le meilleur et le
pire.*

*Nous tenons à le dédier aussi à nos amis, source
de folie au quotidien.*

*Merci à vous,
Du fond du cœur.*



REMERCIEMENT

Avant tout remerciement, louange à DIEU TOUT PUISSANT de nous avoir accordé la force d'accomplir cet humble travail.

*Au terme de notre stage effectué à la Société « **Dar du commerce moderne** », nous aimerions remercier vivement la société qui nous a accueillis, ainsi que **M. OMAR TOUATI**, pour son aide précieuse, pour les informations et notices techniques qui nous a fourni, ainsi que pour sa disposition.*

*Nous adressons pareillement nos remerciements à notre encadrant à la FSTF, **M. KHALID ZENKOUAR** qui a fait profiter de son aide, sa collaboration, et son orientation très utile pour la réussite de notre projet.*

*Nos remerciements s'adressent également à tous **les membres de jury** pour l'honneur qu'ils nous ont fait en acceptant d'examiner et juger notre travail.*

Enfin, nous tenons à remercier tous ceux qui ont contribué de près ou de loin pour accomplir ce projet, en prodiguant généralement leur aide accompagnée de sympathie et d'encouragements, qu'ils trouvent ici l'expression de notre sincère gratitude.

MERCI



LISTE DES FIGURES

Figure 1 : Cycle de vie Modèle Incrémental et Itératif.....	17
Figure 2 : Les incréments du Modèle Incrémental et Itératif.....	17
Figure 3 : Model MVC.....	19
Figure 4 : Les acteurs.....	21
Figure 5 : Diagramme de package.....	24
Figure 6 : Diagramme de cas d'utilisation d'agent.....	25
Figure 7 : Diagramme de cas d'utilisation de l'administrateur.....	26
Figure 8 : Diagramme de séquence pour gérer stock	27
Figure 9 : Diagramme de séquence pour consulter crédit	28
Figure 10 : Diagramme de séquence pour gérer chèque	29
Figure 11 : Diagramme de classe	30
Figure 12 : Schéma de l'application	39
Figure 13 : Fenêtre d'accueil	40
Figure 14 : Fenêtre choix client	41
Figure 15 : Fenêtre d'ajouter ou de sélectionner un client	42
Figure 16 : Fenêtre facture client	43
Figure 17 : Fenêtre facture invité	44
Figure 18 : Fenêtre d'authentification	45
Figure 19 : Fenêtre gérer client	46
Figure 20 : Fenêtre de gérer ou consulter stock.....	47
Figure 21 : Fenêtre pour gérer stock.....	48
Figure 22 : Fenêtre pour consulter stock	49
Figure 23 : Fenêtre modifier stock	50
Figure 24 : Fenêtre pour consulter chèque	51
Figure 25 : Fenêtre pour ajouter chèque	52
Figure 26 : Fenêtre pour modifier chèque	52
Figure 27 : Fenêtre pour l'échéant.....	53
Figure 28 : Menu accueil ajouter facture	54
Figure 29 : Menu accueil stock	55
Figure 30 : Menu mot de passe	56
Figure 31 : Menu à propos de Qt.....	57
Figure 32 : Fenêtre à propos de Qt	58



INTRODUCTION GÉNÉRALE

Le présent document est le fruit de notre travail qui s'inscrit dans le cadre du projet de fin d'étude effectué au sein du Dar du Commerce Moderne en vue d'obtenir la licence de la Faculté des Sciences et Techniques Fès.

Ce projet a pour but de réaliser une application pour la gestion des stocks, génération des factures et gestion des chèques.

Durant notre projet, nous aurons pour mission dans un premier temps de cerner le sujet. Après une analyse approfondie de la problématique, comme :

- Les conflits avec les clients.
- Perte de temps à calculer la somme de la facture manuellement.
- Oublier les dates des chèques à payer ou à retirer.

Nous avons élaboré les différents diagrammes. Ensuite, nous avons abordé la phase de la mise en œuvre et de l'implémentation de la solution. La dernière étape a fait l'objet du déploiement des tests et de validation.

Trois chapitres structurent ce rapport, et apportent progressivement les éléments suivants :

✓ Chapitre1 : le contexte général du projet

Présentation du lieu du stage, et l'étude du projet en définissant la problématique ainsi que la solution proposée.

✓ Chapitre2: l'analyse fonctionnelle du projet

En décrivant les fonctionnalités du système ainsi que l'étude conceptuelle qui constitue les différents diagrammes UML.

✓ Chapitre3: la présentation de l'application

Cette partie sera consacrée aux outils et langages de développement utilisés, à la réalisation du projet et la présentation de l'application.



CHAPITRE I : CONTEXTE GÉNÉRAL DU PROJET



Introduction :

Dans ce chapitre, nous allons donner un bref aperçu sur notre projet :

- Présentation du lieu de stage.
- Cahier des charges.
- Problématique et solutions.

1. PRÉSENTATION DU LIEU DE STAGE :

Cette société a été fondée depuis 2000 sous le nom de "Dar du commerce moderne ", leur siège se situe au (81 aqba Kaide Khamar Fès). L'objectif principal de la création de cette institution est d'avoir des investissements internationaux surtout en Chine, afin d'atteindre le maximum des importations et de renforcer les relations commerciales. Comme on avait déjà mentionné ci- dessus ; les activités de la société sont limitées uniquement au commerce selon le plan suivant :

- Le commerce de gros ou commerce inter – entreprises
- Le commerce saisonnier spécifiquement en trois saisons: La rentrée scolaire, Aïd Al Adha, Achoura, où il se concentre seulement sur les produits les plus utilisés en une période précise.
- Contrairement aux trois saisons précédentes les activités se centrent surtout sur le financement des bibliothèques.

2. PRÉSENTATION DU PROJET :

2.1. Cahier de charge :

Une société de vente de gros désire mettre en place un système informatisé pour ses services de ventes et d'achats et gestion de crédit et de chèque.

Quand les produits arrivent, l'agent vérifie manuellement la quantité et l'état de chaque produit. Après la vérification, l'agent saisit la quantité des éléments déjà existant, ajoute les nouveaux éléments (désignation, prix d'achat, quantité, commentaire). Le système doit modifier la quantité des éléments qui existent déjà, et ajouter les nouveaux produits.



➤ **L'agent peut :**

- Consulter la liste des produits soit par ordre alphabétique ou bien par quantité.
- Consulter la liste des chèques qui doivent être payés et la liste des chèques qui doivent être retirés.
- Ajouter, modifier ou supprimer les chèques à retirer ou à payer.
- Rechercher les chèques par date.

➤ **L'administrateur peut :** (En sus des actions d'agent)

- Modifier ou supprimer un produit.
- Consulter, modifier ou supprimer les informations sur le client.

Lorsqu'un client arrive l'agent saisit sa commande en sélectionnant le nom du produit et en mentionnant la quantité et le prix, après la validation, le système vérifie la quantité et le prix :

- Si la quantité désirée est supérieure à la quantité dans le stock, le système l'informe de la quantité existante.
- Si le prix est inférieur au prix d'achat le système l'informe et lui demande soit de le modifier ou de l'ignorer.

Après la validation, le système ajoute le produit à la liste de la facture, l'agent peut modifier ou supprimer un élément de cette dernière.

A chaque action le montant total de la facture est initialisé.

On génère la facture et un fichier qui sauvegarde la facture avec la date, et le système modifie les quantités des éléments de la facture.

Chaque facture est titrée par le nom de la personne ou de la société ayant fait les achats.

Il y a deux sortes de clients pour la société, un client fidèle et un client normal :

- Le client fidèle a le droit à un crédit, un total de ses crédits et un historique de ses factures.



2.2. Enoncé du problème :

Dar du commerce moderne accueille quotidiennement un grand nombre de clients de la région de Fès ou d'autres régions. Ceci pose un véritable problème de gestion des stocks, historique des factures et La transaction avec le client. Actuellement, la société est gérée d'une manière manuelle, car elle n'utilise que des factures manuscrites et des registres pour enregistrer les produits entrants Cette méthode est très frustrant, car ils ne peuvent pas bénéficier seulement après des heures d'examen et de rechercher dans une pile de papier.



Cette méthode de travail possède un nombre important de problèmes tels que :

- problèmes de gestion des stocks : L'absence d'informations sur ce qui existe et la quantité en stock.
- Problèmes de gestion des factures : l'impossibilité de conserver tous les factures
- Problèmes de la relation factures-stock : la facture contient souvent des produits inexistants ou de quantités inférieure à la demande.
- Problèmes de création des factures :
 - La lisibilité : Parce qu'il est écrit à la main.
 - Calcul du montant : Le besoin de plus de temps pour faire.



- La modification : la facture Peut contenir des terribles erreurs (comme des prix) doivent être corrigées en ce qui nécessite de réécriture et de recalculer cette facture.
- Problèmes de la notation et souvenir des crédits de chaque client.
- Problèmes de Respect des dates de paiement des chèques

Ces problèmes engendrent :

- Des frustrations des agents et des clients.
- Une anarchie dans le travail.
- Une perte de temps.
- Nous avons proposé la mise en place d'une application informatique pour générer la facture, gestion automatique de stock et poser la main sur les crédits et les cheque à payer et à retirer.

2.3. Solutions proposées :

La résolution de ce problème consiste à développer une application pour améliorer la gestion au sein de la société. L'application va être développée en C++ selon l'architecture MVC.

Cette application fera gagner un temps colossal et rendra le travail plus organisé. On va transformer la méthode de travail classique et statique en une autre dynamique.

L'application va garantir un traitement automatisé de ces procédures en utilisant des interfaces graphiques simples et faciles à comprendre et qui va en particulier:

- ✓ Organiser le travail de l'agent, ainsi que de l'administrateur.
- ✓ Permettre aux 2 acteurs de rechercher l'information dont ils ont besoin en un temps réduit.
- ✓ Organiser la gestion des stocks.
- ✓ Faciliter la génération de la facture.

Les solutions proposées seront détaillées dans les chapitres qui suivent.



Conclusion :

Nous avons vu dans ce chapitre un aperçu sur notre projet, dans le chapitre qui suit nous allons voir l'analyse fonctionnelle du projet, les acteurs, leurs rôles, et les différents diagrammes UML.



CHAPITRE II : ANALYSE FONCTIONNELLE



Introduction :

Dans ce deuxième chapitre, nous allons présenter l'analyse fonctionnelle de notre projet (la méthodologie d'analyse). Nous montrerons ensuite les acteurs de l'application, leurs rôles, ainsi que les différents diagrammes UML.

1. MÉTHODOLOGIES D'ANALYSE:

1.1. Le langage UML :

UML ou Langage de Modélisation Unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML est couramment utilisé dans les projets logiciels. Les différents éléments sont :

- Activité d'un objet/logiciel.
- Acteurs.
- Processus.
- Schéma.
- Composants logiciels.
- Réutilisation de composants.

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie code, par exemple en langage Java, à partir des divers documents réalisés.

1.2. Le Modèle Incrémental et Itératif :

La phase d'étude est la partie la plus importante pour tout projet réussi. On s'est basé durant la réalisation de notre application à des normes universelles durant la conception, en particulier le respect des principes du Modèle Incrémental.

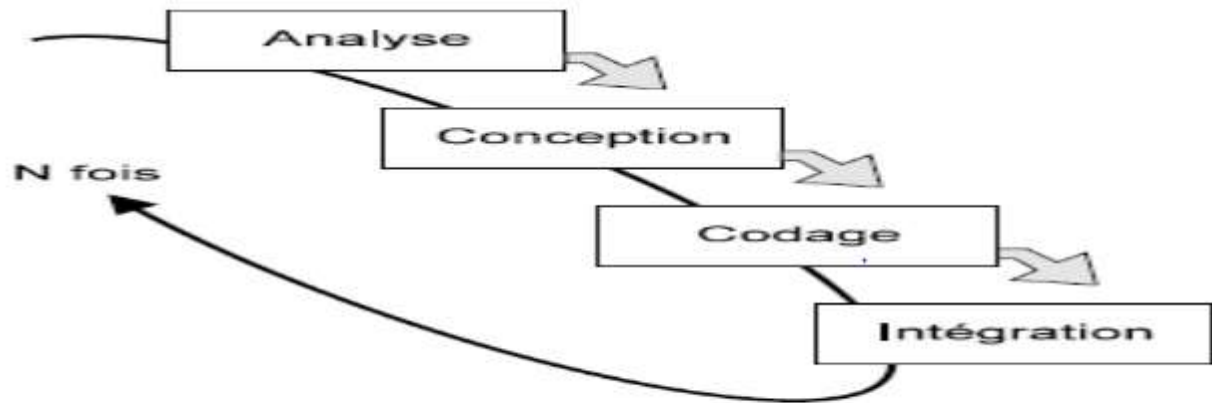


Figure 1 : Cycle de vie Modèle Incrémental et Itératif

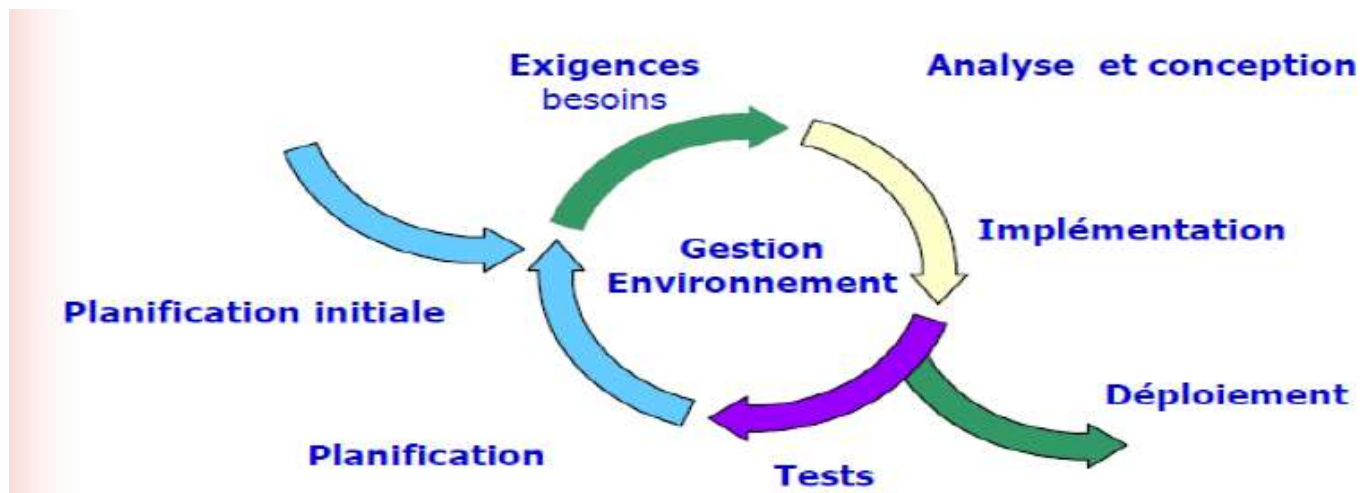


Figure 2 : Les incréments du Modèle Incrémental et Itératif

- Le projet de développement est découpé en plusieurs petits projets.
- Chaque projet représente une itération qui:
- Donne lieu à un incrément (version du produit).
- Prend en charge une partie des besoins.
- Répond à un ensemble de risques.
- Le développement se déroule en plusieurs itérations.
- Le projet est décomposé en un noyau et plusieurs incréments.
- Chaque incrément est développé séparément ou en parallèle.



1.2.1. Avantage :

- Flexibilité (agilité) vis à vis de nouveaux besoins ou des changements.
- Pas de blocage en cas de spécifications incomplètes.
- Meilleure testabilité.
- Découverte de malentendu assez tôt pour les corriger.
- Répartition de l'effort dans le temps.
- Objectifs réduits et clairs.
- Utilisation de l'approche «diviser pour régner».
- Le client rentre en relation avec le produit très tôt.

1.2.2. Inconvénients :

- Difficultés de gestion du projet.
- Difficultés de contrôle qualité.
- Exigence d'une bonne planification et d'une bonne conception.
- Exigence d'une vision sur le produit fini pour bien diviser en incréments.

1.2.3. Incréments principal du projet

Notre projet est constitué de 10 incréments principaux :

- Incrément gestion des factures.
- Incrément gestion de stock.
- Incrément gestion des clients.
- Incrément gestion des chèques.

1.3. Le Modèle MVC (Modèle-Vue-Contrôleur) :

L'architecture MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur), ce qui assure la clarté de l'architecture et simplifie la tâche du développeur responsable de la maintenance et de l'amélioration du projet. Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure 3.

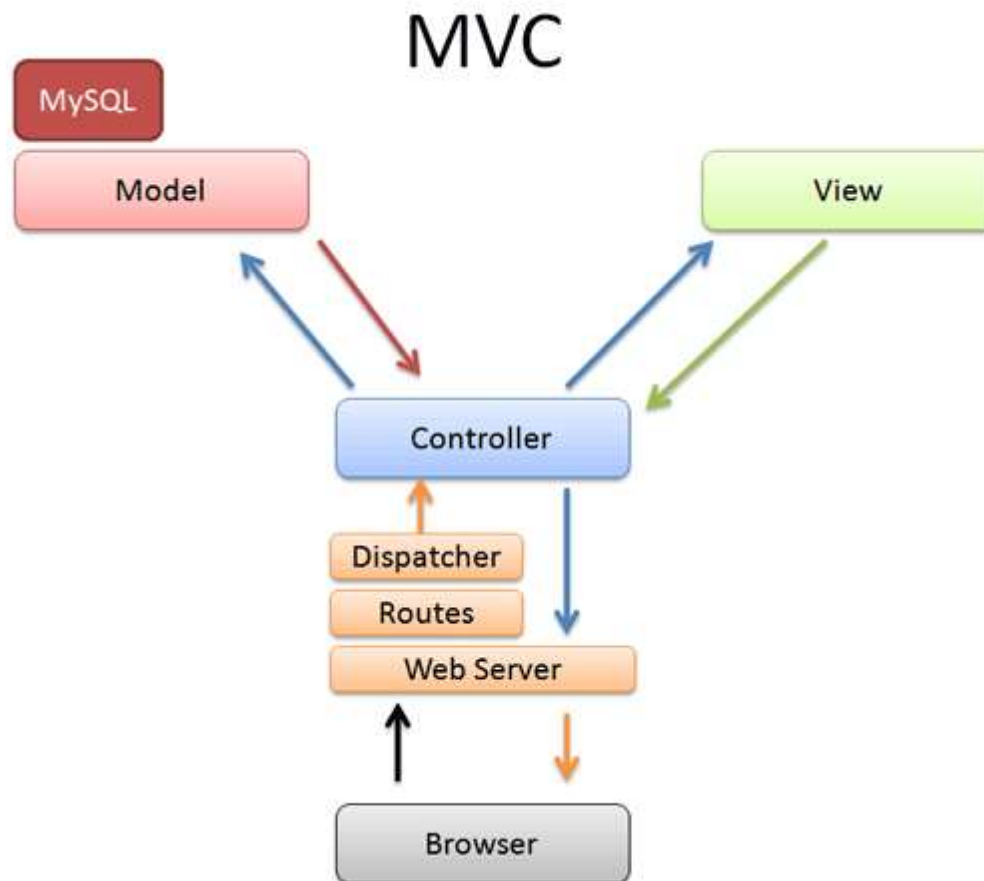


Figure 3 : Model MVC

1.3.1. Définition MVC (Modèle-Vue-Contrôleur) :

➤ **Le Modèle :**

Le modèle représente le cœur de l'application : traitements des données, interactions avec la base de données. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation, le modèle ne contient aucun lien direct vers la vue.



➤ La Vue :

C'est avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle, sa seconde tâche est de recevoir toute action de l'utilisateur (clic de souris, sélection d'un bouton radio, coche d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur.

La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

➤ Le Contrôleur :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et déclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ce dernier notifie la vue que les données ont changé pour qu'elle se mette à jour. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est « observable ».

Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée, il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

1.3.2. Avantages du MVC :

- ✓ Une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur.
- ✓ Un gain de temps de maintenance et d'évolution du site.
- ✓ Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l'affichage et des actions).



2. MODÉLISATION DU CONTEXTE :

2.1. Les acteurs et leurs rôles :

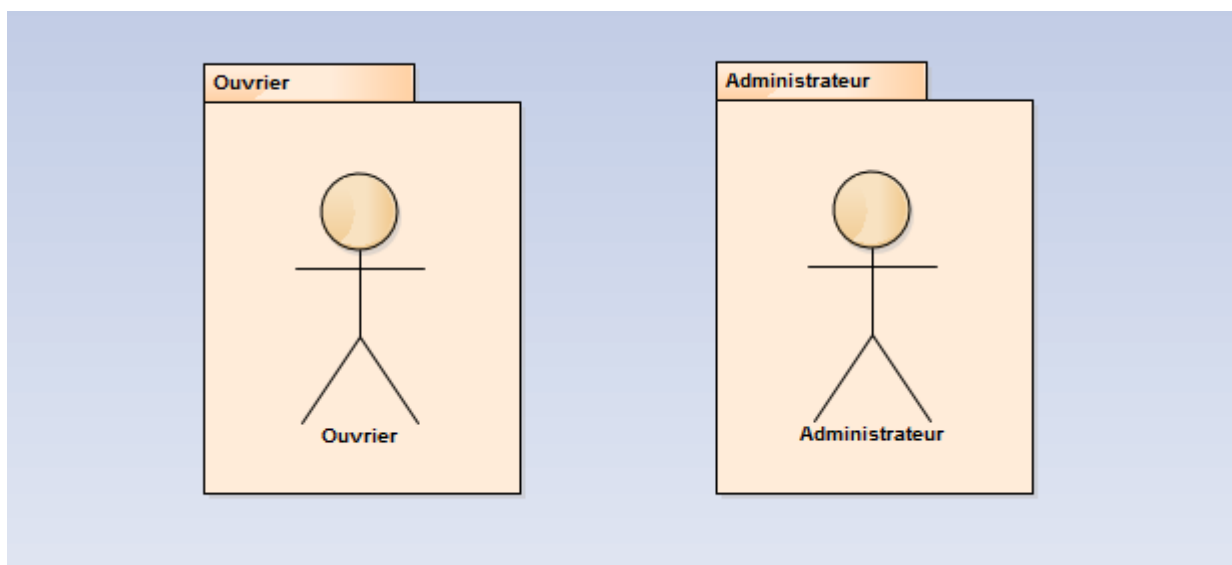


Figure 4 : Les acteurs

Acteurs	Rôles
Agent	- ajoute un nouveau client - générer la facture - ajoute un nouveau produit - lister les produits par alphabétique ou par quantité - consulter les chèques
Administrateur	En sus des actions d'agent il : - modifie /supprime un client - modifie /supprime un crédit - modifie /supprime un produit - consulte les crédits des clients



2.2. Les messages émis et reçus :

Cas d'utilisation	Acteurs	Messages émis /Messages reçus
Ajouter un produit	- Administrateur - Agent	Emis: Ajouter informations d'un nouveau produit. Reçus: Confirmation.
Ajouter une quantité	- Administrateur - Agent	Emis: Ajouter la nouvelle quantité du produit Reçus: Confirmation.
Modifier un produit	Administrateur	Emis: choisir le produit à modifier Reçus: demande de spécifier les changements et validation.
Supprimer un produit	Administrateur	Emis: choisir le produit à supprimer. Reçus: confirmation.
Ajouter un client	- Administrateur - Agent	Emis: Ajouter informations d'un nouveau client. Reçus: Confirmation.
Modifier un client	- Administrateur - Agent	Emis: choisir le client à modifier Reçus: demande de spécifier les changements et validation.
Supprimer un client	- Administrateur - Agent	Emis: choisir le client à supprimer. Reçus: confirmation.
Liste Alphabétique	- Administrateur - Agent	Emis: demande de listage alphabétique. Reçus: la liste des produits par ordre alphabétique.
Liste Quantité	- Administrateur - Agent	Emis: demande de listage par quantité. Reçus: la liste des produits par ordre croissante de la quantité.
Consulter chèque	- Administrateur - Agent	Emis: demande d'afficher historique des chèques. Reçus: affichage de résultats.



UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES FES
DÉPARTEMENT D'INFORMATIQUE



Ajouter chèque	• Administrateur • Agent	Emis: Ajouter informations d'un nouveau chèque. Reçus: Confirmation.
Modifier chèque	• Administrateur • Agent	Emis: choisir chèque à modifier Reçus: demande de spécifier les changements et validation.
Supprimer chèque	• Administrateur • Agent	Emis: choisir le chèque à supprimer. Reçus: confirmation.
Chèque à venir	• Administrateur • Agent	Emis: Entrer la date. Reçus: affichage de résultats.
Authentification	• Administrateur	Emis: authentification et accès au compte. Reçus : demande d'authentification et connexion.



3. ANALYSE ET CONCEPTION:

Cette étape consiste à formaliser et à détailler les besoins exprimés lors de l'étude préliminaire, celle-ci sera réalisée principalement à l'aide des cas d'utilisation qui permettent de capturer la fonctionnalité du système au point de vue utilisateur.

3.1. Diagramme de package :

C'est un moyen pour regrouper les différents éléments de la modélisation. Il permet de représenter les relations entre les différents profils de l'application. Il rassemble les cas d'utilisations propre à chaque acteur de façon cohérente.

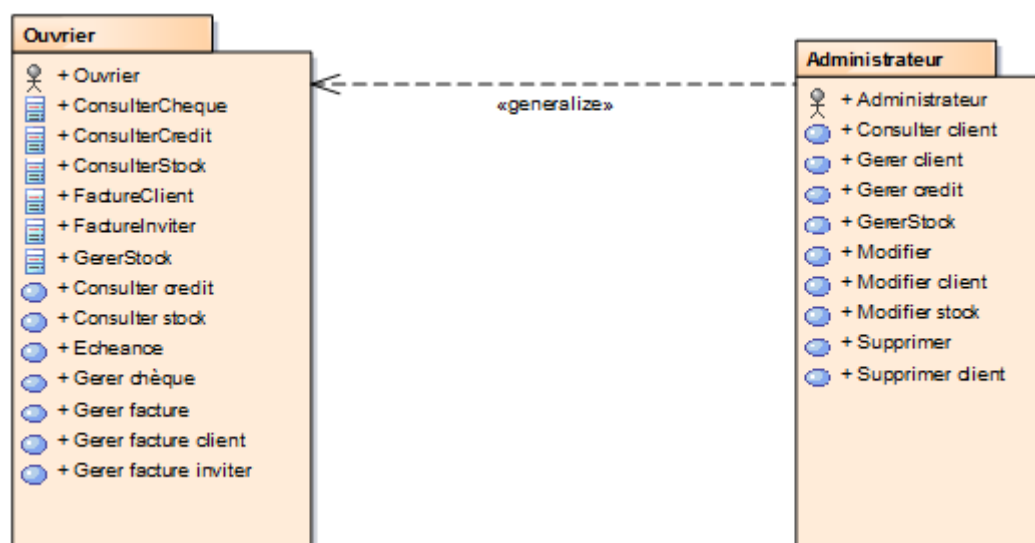


Figure 5 : Diagramme de package



3.2. Diagrammes des cas d'utilisation :

Il modélise un service rendu par le système utilisé afin de donner une vision globale du comportement fonctionnel d'un système logiciel.
Il représente une séquence d'actions réalisée par le système.

Agent :



Figure 6 : Diagramme de cas d'utilisation d'agent



Administrateur :

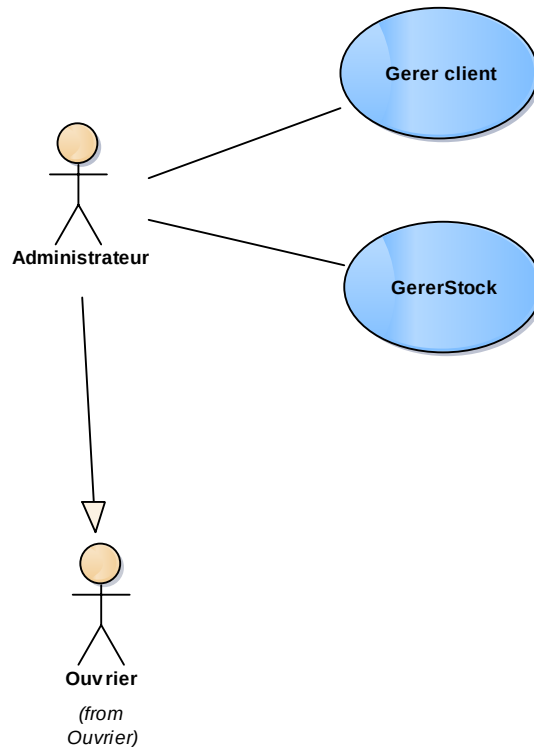


Figure 7 : Diagramme de cas d'utilisation de l'administrateur



3.3. Diagrammes de séquences :

Le diagramme de séquence permet d'illustrer les cas d'utilisation et de représenter les interactions dans le temps entre les objets du système.

Gérer stock :

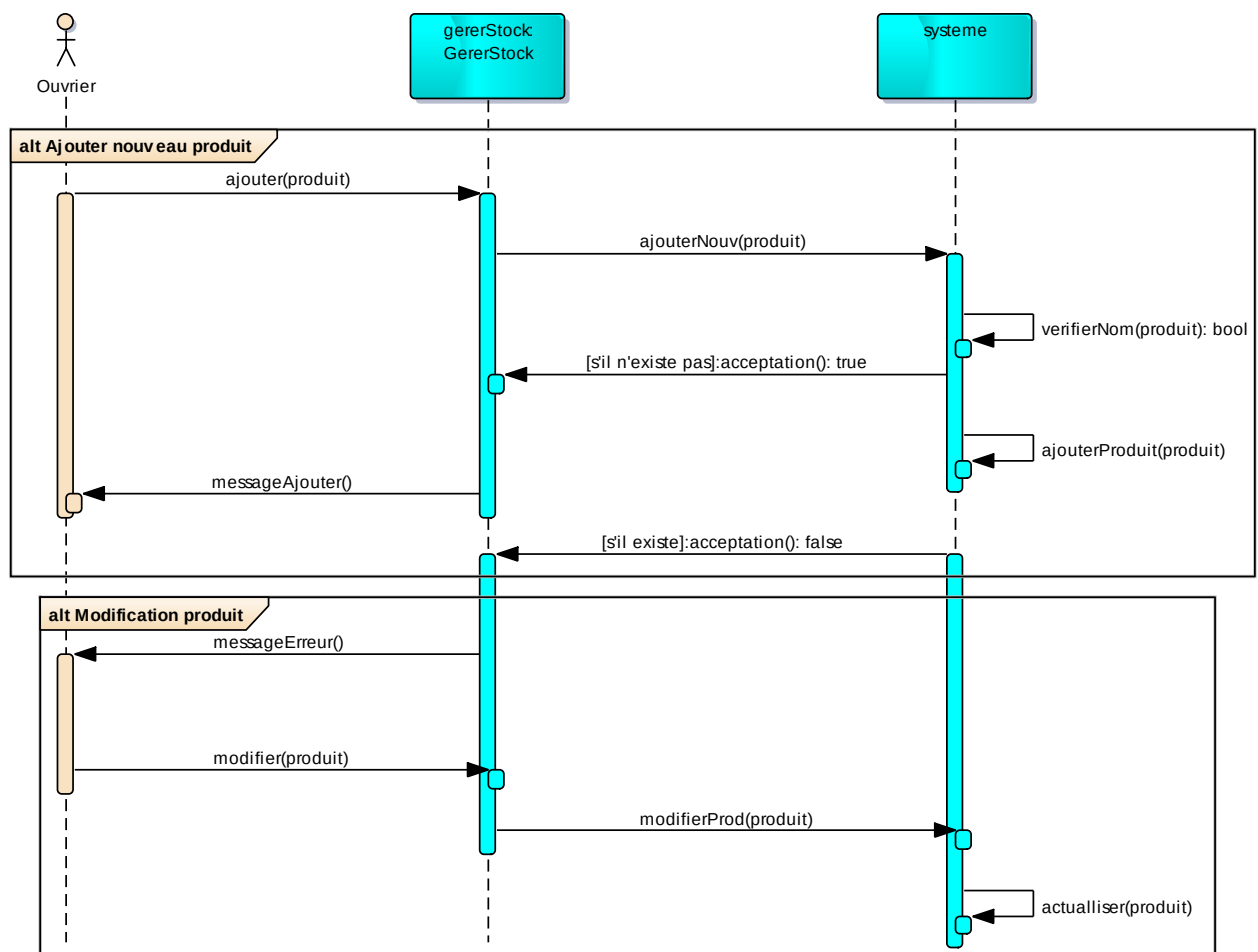


Figure 8 : Diagramme de séquence pour gérer stock



Consulter Crédit :

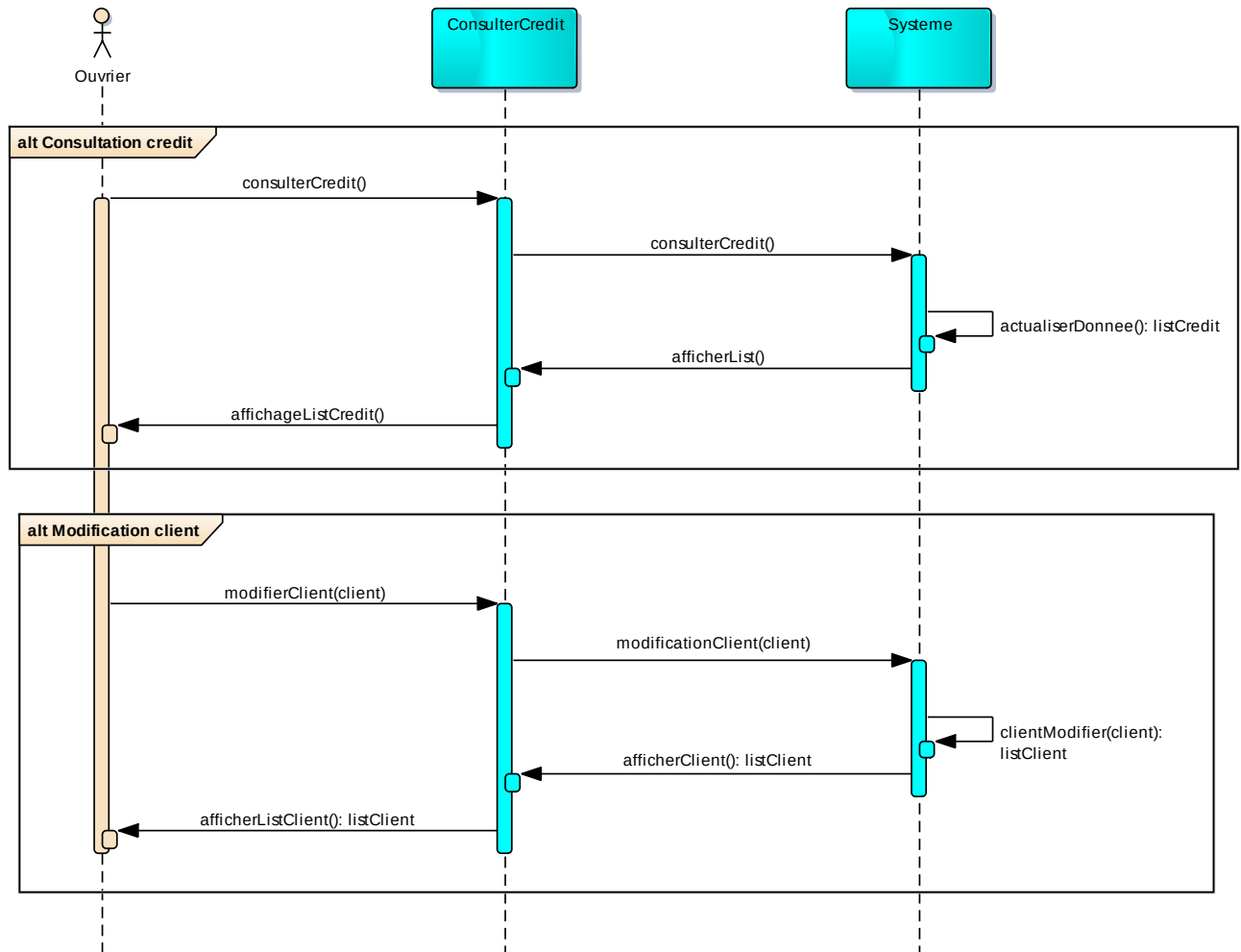


Figure 9: Diagramme de séquence pour consulter crédit



Gérer Chèque :

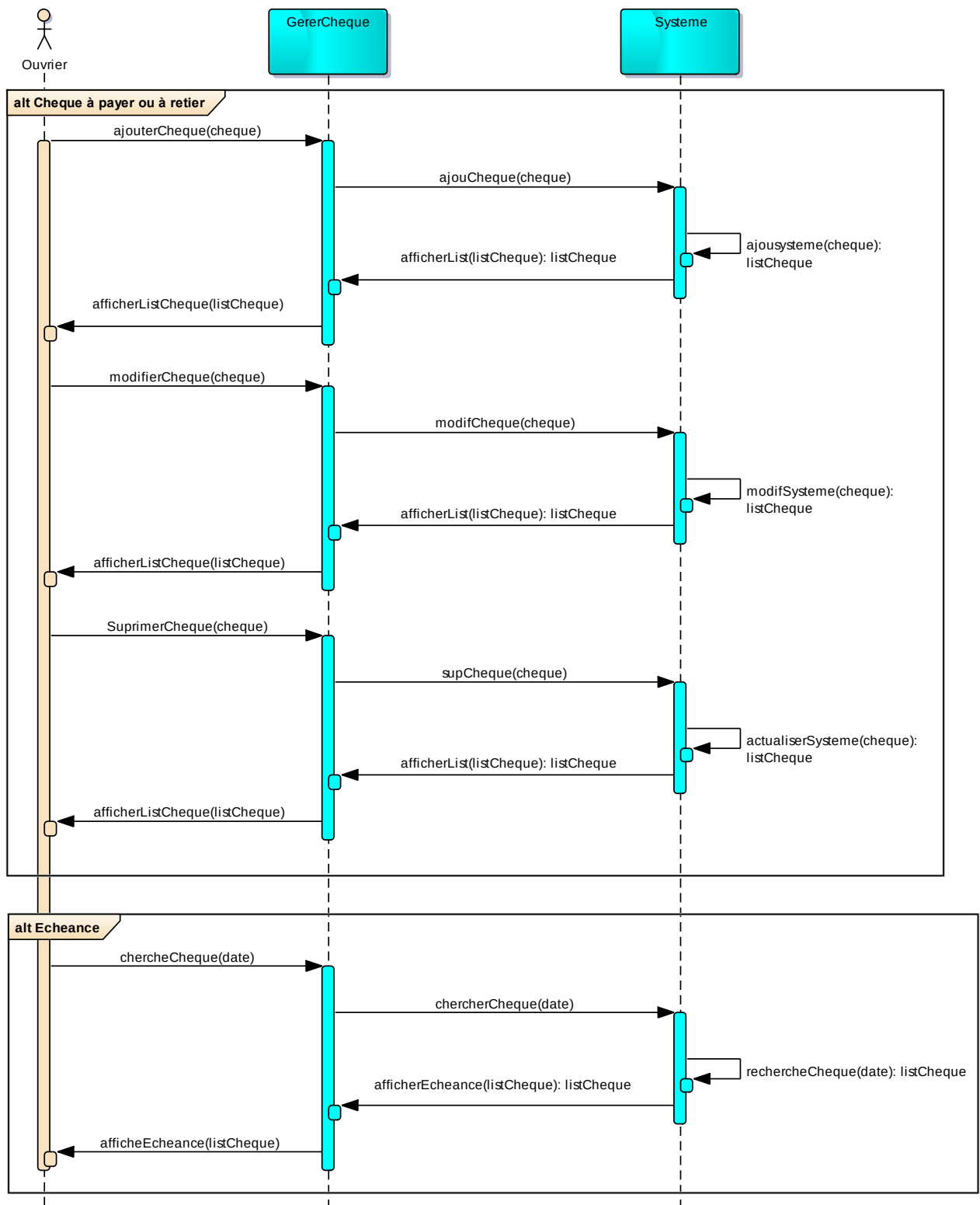


Figure 10 : Diagramme de séquence pour gérer chèque



3.4. Diagramme de classe :

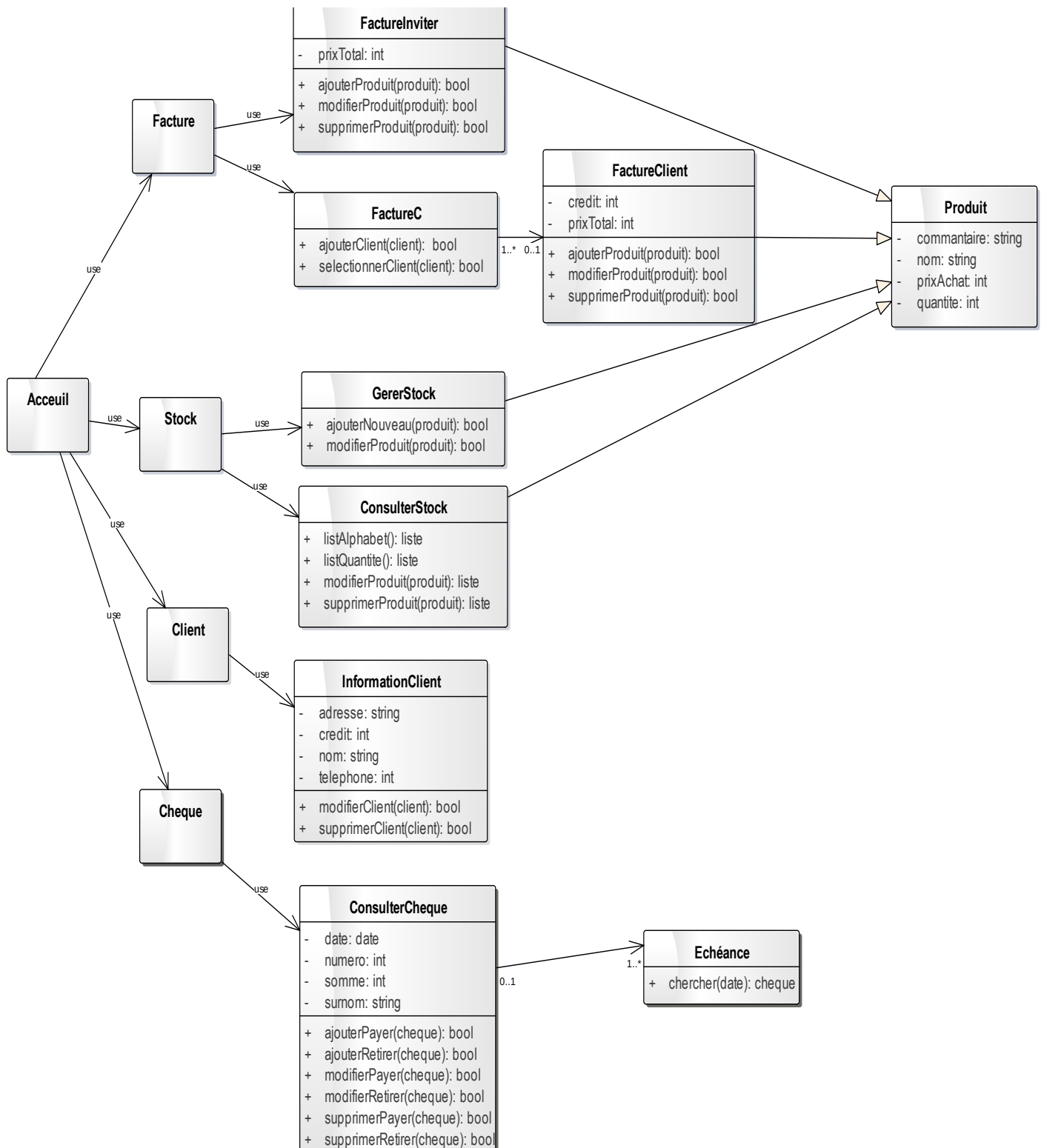


Figure 11 : Diagramme de classe



Conclusion :

Nous avons vu dans ce chapitre l'analyse fonctionnelle du projet, les acteurs, leurs rôles, et les différents diagrammes, dans le chapitre qui suit nous allons parler des outils de développement et les étapes de la réalisation.



CHAPITRE III : PRÉSENTATION DE L'APPLICATION



Introduction :

Dans ce chapitre, nous allons aborder la partie réalisation du projet. Plus précisément nous parlerons des outils de développement, puis nous allons entamer les étapes de la réalisation et quelques captures d'écran pour expliquer le fonctionnement de l'application.

1. OUTILS ET TECHNOLOGIES DE DÉVELOPPEMENT :

1.1. Langages utilisés :

1.1.1. C++ :



Le C++ est un langage de programmation permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique. C++ est actuellement le 3e langage le plus utilisé au monde. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation.

Bjarne Stroustrup a développé C++ au cours des années 1980, alors qu'il travaillait dans le laboratoire de recherche Bell d'AT&T. Il s'agissait en l'occurrence d'améliorer le langage C. Il l'avait d'ailleurs nommé C with classes (« C avec des classes »). Les premières améliorations se concrétisèrent donc par la prise en charge des classes, ainsi que par de nombreuses autres fonctionnalités comme les fonctions virtuelles, la surcharge des opérateurs, l'héritage (simple ou multiple), les « templates », la gestion des exceptions, etc.



Le langage C++ est normalisé par l'ISO. Sa première normalisation date de 1998 (ISO/CEI 14882:1998). Le standard a ensuite été amendé par l'erratum technique de 2003 ISO/CEI 14882:2003. Le standard actuel a été ratifié et publié par ISO en septembre 2011 sous le nom de ISO/IEC 14882:2011. (Aussi appelé C++11).

En langage C, ++ est l'opérateur d'incrément, c'est-à-dire l'augmentation de la valeur d'une variable de 1. C'est pourquoi C++ porte ce nom: cela signifie que C++ est un niveau au-dessus du C. Il existe de nombreuses bibliothèques C++ en plus de la bibliothèque standard de C++ (C++ Standard Library) qui est incluse dans la norme. Par ailleurs, C++ permet l'utilisation de l'ensemble des bibliothèques C existantes.

Avantages :

- Grand nombre de fonctionnalités
- Facilité d'utilisation des langages objets
- Portabilité des fichiers sources
- C++ beaucoup plus rapide que java

1.1.2. Qt :



Qt est une **bibliothèque** multiplateforme pour créer des GUI (programme utilisant des fenêtres). Qt est écrite en C++ et elle est, à la base, conçue pour être utilisée en C++. Toutefois, il est aujourd'hui possible de l'utiliser avec d'autres langages comme Java, Python, etc.

Plus fort qu'une bibliothèque : un Framework

Qt (voir logo en figure suivante) est en fait... bien plus qu'une bibliothèque. C'est un *ensemble de bibliothèques*. Le tout est tellement énorme qu'on parle d'ailleurs plutôt de **Framework** : cela signifie que vous avez à votre disposition un ensemble d'outils pour développer vos programmes plus efficacement.



Qu'on ne s'y trompe pas : Qt est fondamentalement conçue pour créer des fenêtres, c'est en quelque sorte sa fonction centrale. Mais ce serait dommage de la limiter à cela.

Qt est donc constituée d'un ensemble de bibliothèques, appelées « modules ». On peut y trouver entre autres ces fonctionnalités :

- **Module GUI** : c'est toute la partie création de fenêtres. Nous nous concentrerons surtout, dans ce cours, sur le module GUI.
- **Module OpenGL** : Qt peut ouvrir une fenêtre contenant de la 3D gérée par OpenGL.
- **Module de dessin** : pour tous ceux qui voudraient dessiner dans leur fenêtre (en 2D), le module de dessin est très complet !
- **Module réseau** : Qt fournit une batterie d'outils pour accéder au réseau, que ce soit pour créer un logiciel de Chat, un client FTP, un client Bittorrent, un lecteur de flux RSS...
- **Module SVG** : Qt permet de créer des images et animations vectorielles, à la manière de Flash.
- **Module de script** : Qt prend en charge le Javascript (ou ECMAScript), que vous pouvez réutiliser dans vos applications pour ajouter des fonctionnalités, par exemple sous forme de plugins.
- **Module XML** : pour ceux qui connaissent le XML, c'est un moyen très pratique d'échanger des données à partir de fichiers structurés à l'aide de balises, comme le XHTML.
- **Module SQL** : permet d'accéder aux bases de données (MySQL, Oracle, PostgreSQL...).

Avantages :

- Une bibliothèque multiplateforme.
- Une documentation assez complète.
- Le nom des fonctions est explicite.
- Qt est écrite en C++ et elle est, à la base, conçue pour être utilisée en C++.
- Possible de l'utiliser avec d'autres langages comme Java, Python, etc.



1.1.3. SQLite :



SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL. SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID.

Contrairement aux serveurs de bases de données traditionnels, comme MySQL ou PostgreSQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme.

D. Richard Hipp, le créateur de SQLite, a choisi de mettre cette bibliothèque ainsi que son code source dans le domaine public, ce qui permet son utilisation sans restriction aussi bien dans les projets open source que dans les projets propriétaires. Le créateur ainsi qu'une partie des développeurs principaux de SQLite sont employés par la société américaine Hwaci.

SQLite est le moteur de base de données le plus distribué au monde, grâce à son utilisation dans de nombreux logiciels grand public comme Firefox, Skype, Google Gears, dans certains produits d'Apple, d'Adobe et de McAfee et dans les bibliothèques standards de nombreux langages comme PHP ou Python. De par son extrême légèreté (moins de 300 Ko), il est également très populaire sur les systèmes embarqués, notamment sur la plupart des smartphones modernes : l'iPhone ainsi que les systèmes d'exploitation mobiles Symbian et Android l'utilisent comme base de données embarquée. Au total, on peut dénombrer plus d'un milliard de copies connues et déclarées de la bibliothèque.

Choisir entre MySQL et SQLite

Les principaux "inconvénients" de SQLite par rapport à MySQL sont :

- écritures atomiques (avec SQLite, un seul processus peut écrire dans la base)
- pas de réplication



Pour quoi utiliser SQLite?

Ce qu'il faut bien garder à l'esprit, c'est que SQLite n'est pas vraiment un concurrent des serveurs de base de données relationnelles, c'est plus une extension de leur champ d'application à des applications offline (là ou avant elles n'étaient utilisé que sur des serveurs sur un modèle client/serveur).

SQLite peut aussi s'avérer très utile sur un site web pour créer des fonctionnalités qui ne sont pas directement liées au site (comme par exemple organiser un jeu concours avec un formulaire où les gens peuvent d'inscrire), évitant ainsi de polluer la base de données principale (fonctionnant sur MySQL par exemple) avec des données temporaires.

L'autre avantage est la simplicité : il n'y a aucune manipulation à faire, le fichier SQLite est créé automatiquement à la volée, toute la base est stockée dans un fichier unique qu'il est facile d'échanger en FTP...

En fait, SQLite peut être utilisé en remplacement de Microsoft Access, car il est bien plus performant, propre, portable et multiplateforme.

1.2. Logiciels utilisés :

1.2.1. Qt Creator :



Qt Creator est un environnement de développement intégré multi plate-forme faisant partie du Framework Qt. Il est donc orienté pour la programmation en C++

Il intègre directement dans l'interface un débogueur, un outil de création d'interfaces graphiques, des outils pour la publication de code sur Git et Mercuriale ainsi que la documentation Qt. L'éditeur de texte intégré permet l'auto complétion ainsi que la coloration syntaxique. Qt Creator utilise sous Linux le compilateur gcc et MinGW par défaut sous Windows.

Qt Creator a été traduit en français par l'équipe Qt de Developpez.com.



1.2.2. Entreprise Architect :



Entreprise Architect est un outil d'analyse de création UML, couvrant le développement du logiciels de rassemblement d'exigences, en passant par les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien.

Cet outil permet de bien schématiser notre application, pour passer de la conception vers la réalisation. Il facilite la représentation des diagrammes UML tels que le diagramme des cas d'utilisation, des séquences et des classes.

L'architecte d'entreprise est un outil conçu pour établir un logiciel facile à mettre à jour.

Il possède un outil de production de documentation souple et de haute qualité.

1.2.3. Inno Setup Compiler :



Inno Setup est un logiciel libre permettant de créer des installateurs pour Windows. Ceux-ci peuvent comporter des scripts programmés en Pascal.

- Inno Setup supporte les systèmes d'exploitation : Windows 8, Windows 7, Windows Vista, Windows Server 2003, Windows XP (cela inclut les éditions x64), Windows 2000, Windows NT 4.0 et Windows 9x.
- Il supporte l'installation des applications 64 bit sur Windows XP et Windows Server 2003.
- Création des installateurs personnalisés.
- Création des raccourcis sur le Bureau ou dans le Menu démarrer.
- Possibilité d'installer des polices.
- Créations des clés dans le registre (par exemple pour associer un type de fichier à un logiciel).



- Possibilité d'exécuter des scripts écrits en Pascal.
- Création des installateurs multilingues.
- Possibilité de chiffrer (avec un mot de passe) l'installateur
- Code source disponible.
- Il supporte Unicode et les langues qui s'écrivent de droite à gauche.

2. SCHÉMA GÉNÉRAL DE L'APPLICATION :

Le schéma suivant présente l'architecture générale de l'application «Gestion d'inventaire d'un grossiste» :

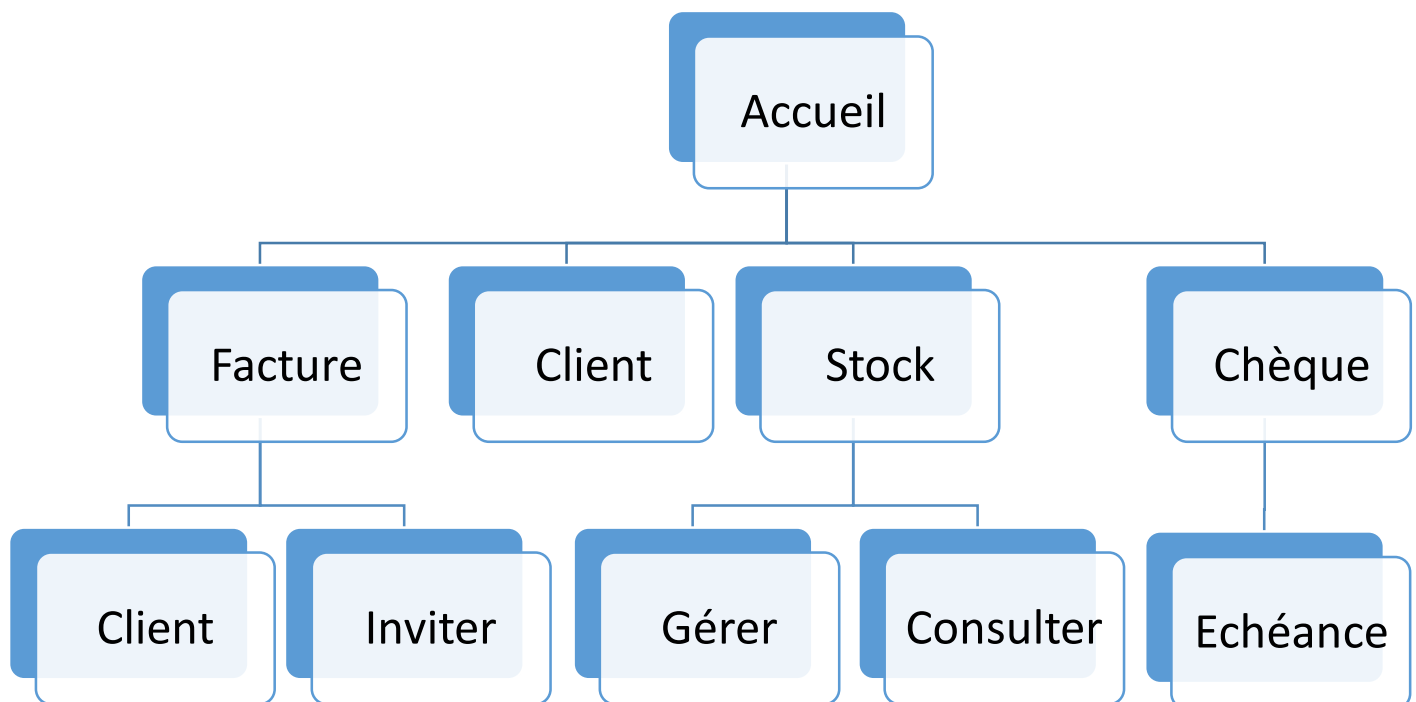


Figure 12 : Schéma de l'application



3. PRÉSENTATION DE L'APPLICATION :

➤ Présentation de la page d'accueil :

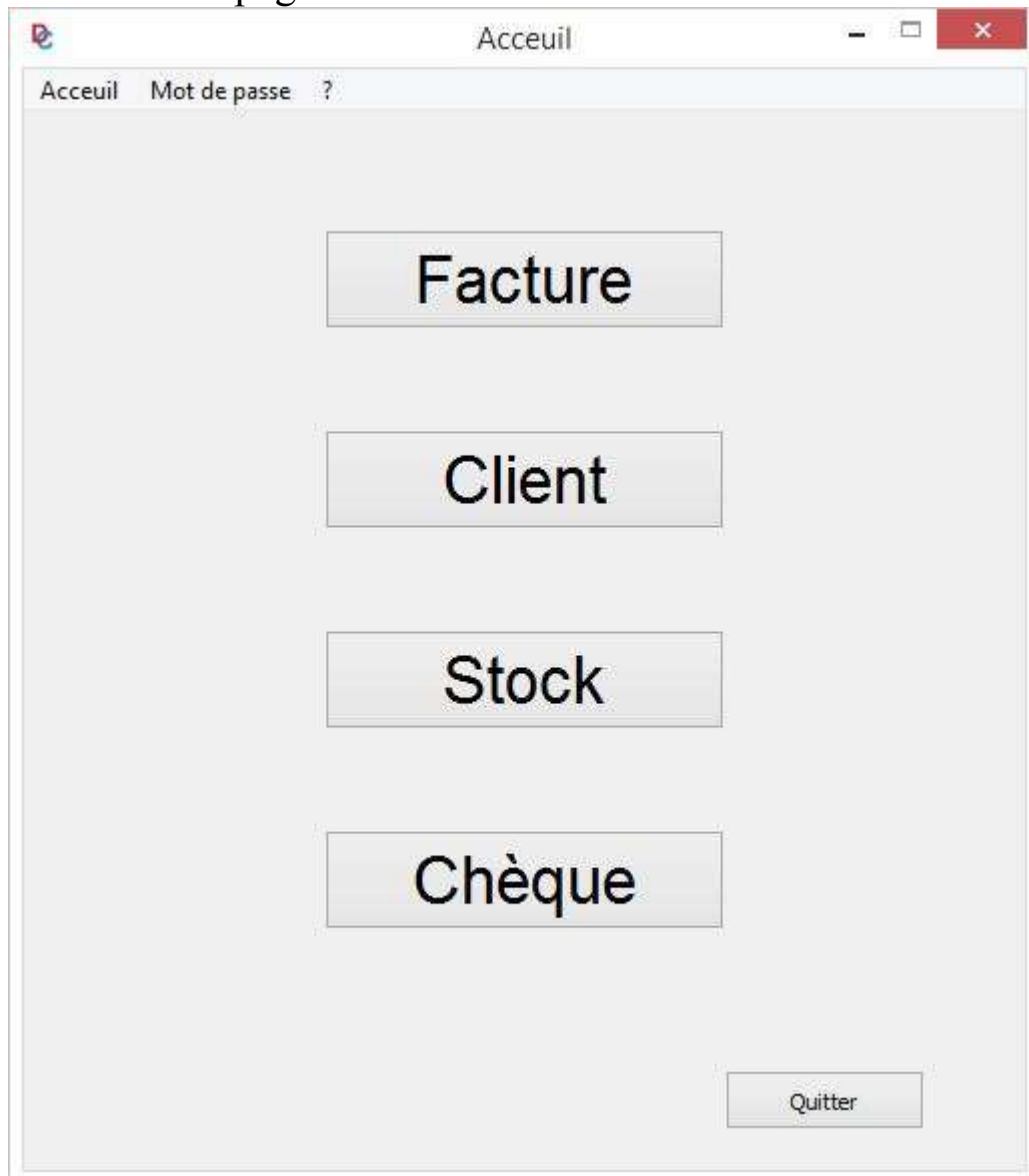


Figure 13 : Fenêtre d'accueil

Cette page est l'interface principale du programme où il est le couloir principal pour atteindre tous les cas d'utilisation.



➤ Présentation d'accès « Facture » :

Après avoir cliqué sur Facture vient l'interface suivante :

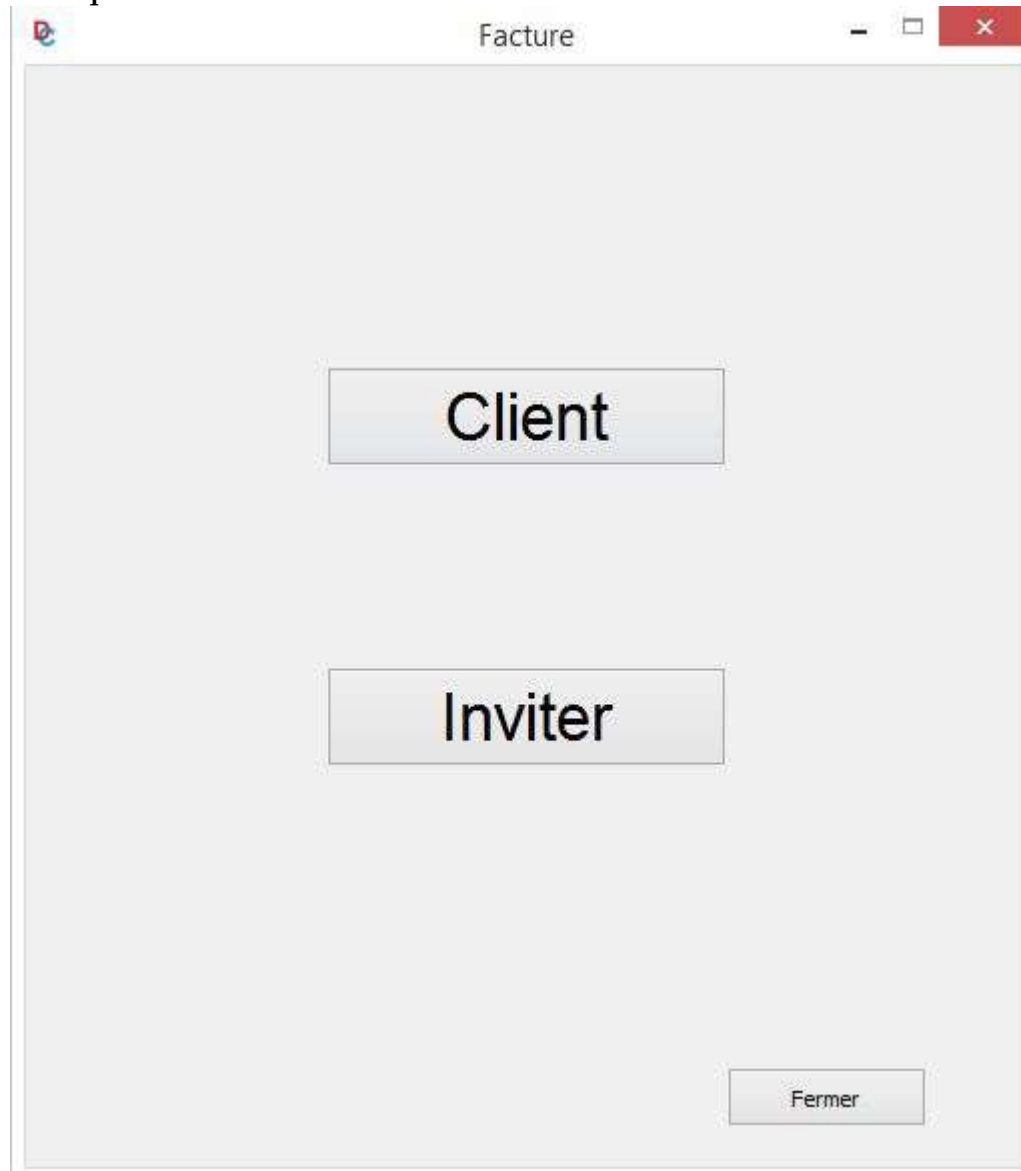


Figure 14 : Fenêtre choix client

Cette interface permet de distinguer entre deux types de client :

- client régulier : peut obtenir un crédit ou payer par chèque.
- client normal (invité).

Si on choisit client on accède vers l'interface suivante :



The screenshot shows a software window titled "Client" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains two main sections:

- Nouveau client:** This section is enclosed in a rounded rectangle. It contains three input fields: "Nom :" (Name), "Téléphone :" (Telephone), and "Adresse :" (Address). Below these fields is a button labeled "Ajouter" (Add).
- Selectionner client:** This section is also enclosed in a rounded rectangle. It contains a dropdown menu labeled "Nom :" with the text "hamza" selected. Below the dropdown is a button labeled "Générer" (Generate).

At the bottom right of the window, outside the rounded rectangles, is a button labeled "Fermer" (Close).

Figure 15 : Fenêtre d'ajouter ou de sélectionner un client

La première partie de cette interface permet d'ajouter un nouveau client en indiquant son Nom, son Téléphone et son adresse. Lorsqu'on clique ajouter le système crée un compte crédit initialisé par 0.

La deuxième partie permet de sélectionner un client déjà existé dans la base de données.

La dernière étape dans cette direction nous conduira à l'interface suivante :



Facture client

Facture N° : 61 Credit : 6000

Ajouter produit

Designation : blanco Quantité : Prix :

Ajouter

Liste produit

Designation	Quantité	Prix
-------------	----------	------

Modifier

Supprimer

Prix total : 0

Générer Fermer

Figure 16 : Fenêtre facture client

Là où on peut générer une facture-client en ajoutant à chaque fois un produit dans le tableau et on peut aussi modifier ou supprimer un élément de ce tableau lorsqu'il veut.



Si on choisit inviter dans la fenêtre "Facture" on accède vers l'interface suivante :

Facture N° : 62

Ajouter produit

Designation : blanco Quantité : Prix :

Ajouter

Liste produit

Designation	Quantité	Prix
-------------	----------	------

Modifier

Supprimer

Nom : Prix total : 0

Générer Fermer

Figure 17 : Fenêtre facture invité

La différence entre la facture client et la facture invité c'est qu'on doit ajouter le nom du client dans la facture.



➤ Présentation d'accès « Client » :

Si on clique sur le bouton client sur la fenêtre d'accueil, on doit s'authentifier, voici l'interface de l'authentification :

Figure 18 : Fenêtre d'authentification

Après l'authentification, on peut consulter ou modifier les informations sur les clients avec l'interface ci-dessous :



Gérer clients

Credit client

	nom	credit	adresse
1	hamza	6000	15 rue mohame..
2	said	2100	3 rue ibn khatib...
3	morad	7800	97 avenue des f...
4	mohamed	2500	15 rue rachidi fes

Modifier

Supprimer

Modification

Nom : Réglement : Credit : 0

Adresse : Téléphone :

Valider

Fermer

Figure 19 : Fenêtre gérer client



➤ Présentation d'accès « Stock » :



Figure 20 : Fenêtre de gérer ou consulter stock



D'après cette interface on peut :

- Gérer stock :

The screenshot shows a window titled "Gérer stock" with two main sections. The top section, "Nouveau produit", contains input fields for "Designation", "Quantité", and "prix", followed by a large text area for "Commentaire" and an "Ajouter" button. The bottom section, "Produit existant", features a "Designation" dropdown menu (currently showing "blanco"), "Quantité" and "prix" input fields, another large "Commentaire" text area, and a "Modifier" button. A "Fermer" button is located at the bottom right of the window.

Figure 21 : Fenêtre pour gérer stock

Ajouter un nouveau produit arrivé ou bien changer la quantité d'un produit existe dans la base des données.



- Consulter stock :

Consultation du stock

Liste alphabétique Liste quantité

	designation	quantite	prixAchat	commentaire
1	stilo bic	5000	1	50 dans chaque...
2	gaume	3000	1	5 * 6 dans chaq...
3	gillet blue 2	2000	5	12 * 10 * 6 dans ...
4	cahier 200	7000	5	10 * 5 dans cha...
5	cahier 100	6000	3	10 * 5 dans cha...
6	registre	5500	30	20 dans chaque...
7	blanco	3500	8	20 dans chaque...
8	papier	2000	20	4 * 12 dans cha...

Modifier

Supprimer

Fermer

Figure 22 : Fenêtre pour consulter stock

Là on peut lister nos produits par deux manières, soit par ordre alphabétique pour faciliter la recherche des produits, ou par quantité pour faciliter à l'administrateur d'effectuer une commande sur les produits à une quantité critique.



On peut aussi modifier un produit on cliquant sur modifier, et on tombe sur l'interface suivante :

Modifier stock

Designation : stilo bic Quantité : 5000 Prix : 1

Commentaire : 50 dans chaque packet

Modifier Annuler

Figure 23 : Fenêtre modifier stock



➤ Présentation d'accès « Chèque » :

Consultation des chèques

A payer

	surnom	numero	date
1	aziz	123456789	20/09/2015
2	younes	987654321	31/08/2015
3	ali	684618354	03/07/2015
4	samir	987654321	11/11/2015
5	achraf	456789123	12/08/2016

Ajouter
Modifier
Supprimer

A retirer

	surnom	numero	date
1	hamid	1597536548	29/05/2016
2	rachid	6846156454	26/12/2015
3	khalid	4098765412	16/11/2015
4	anoir	356184646	12/12/2015
5	adam	147258369	06/09/2015

Ajouter
Modifier
Supprimer

Echéance Fermer

Figure 24 : Fenêtre pour consulter chèque

L'interface ou on peut gérer les chèques à payer ou à retirer



On peut ajouter un chèque à l'aide de cette boîte de dialogue :

Ajouter chèque à p... - [] [X]

Surom :

Numero :

Date :

Somme :

Ajouter Annuler

Figure 25 : Fenêtre pour ajouter chèque

On peut modifier un chèque à l'aide de cette boîte de dialogue en cliquant sur le chèque concerner puis sur modifier :

Modifier chèque à ... - [] [X]

Surom :

Numero :

Date :

Somme :

Modifier Annuler

Figure 26 : Fenêtre pour modifier chèque



Et on cliquant sur "Echéant" on peut chercher les chèques dans une date précise :

Echéance

Recherche

Le :

Rechercher

A payer

A retirer

Fermer

Figure 27 : Fenêtre pour l'échéant



➤ Les menus :

On peut utiliser les menus pour accéder rapidement aux interfaces souhaiter, voici des captures d'écran de ces derniers :

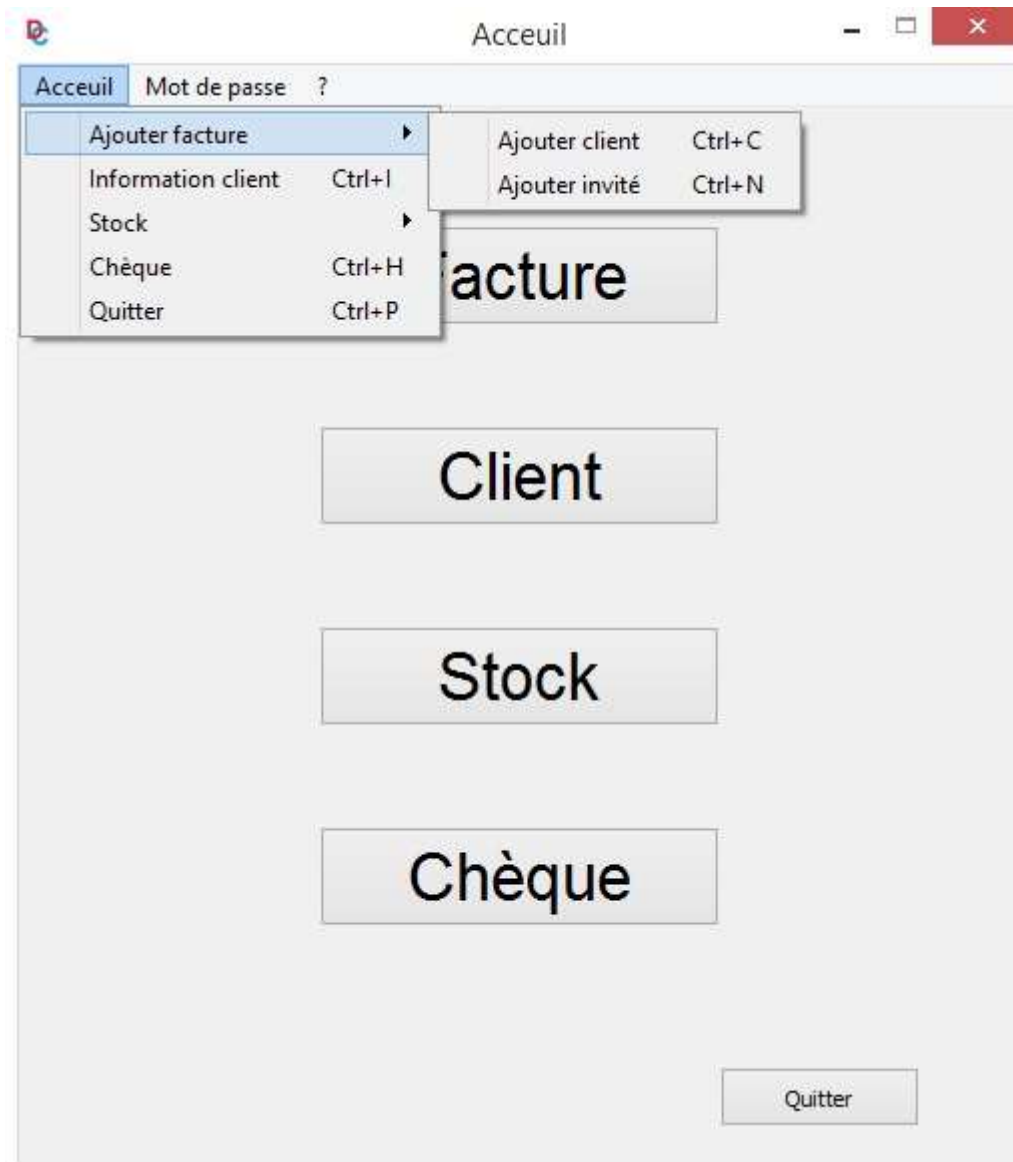


Figure 28 : Menu accueil ajouter facture

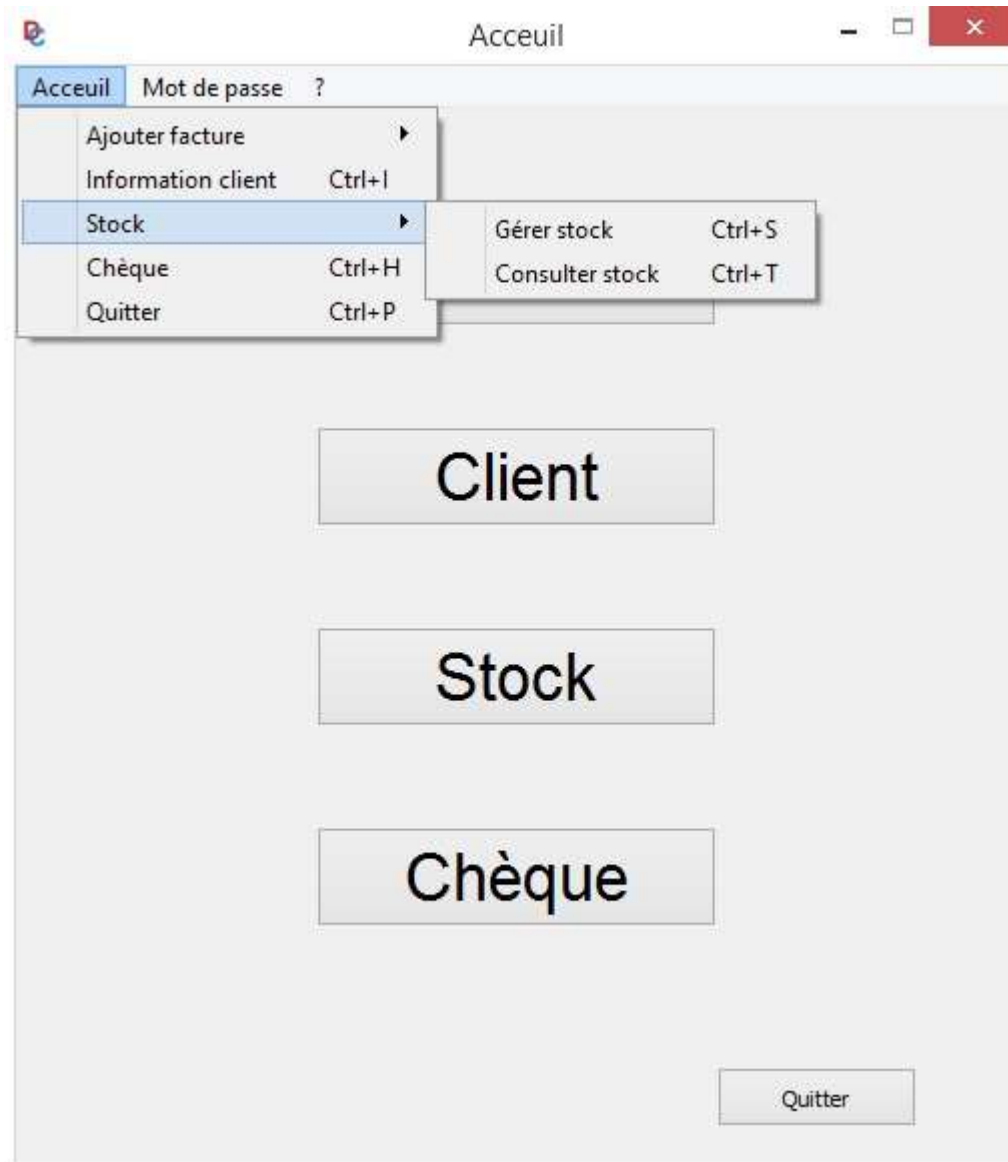


Figure 29 : Menu accueil stock

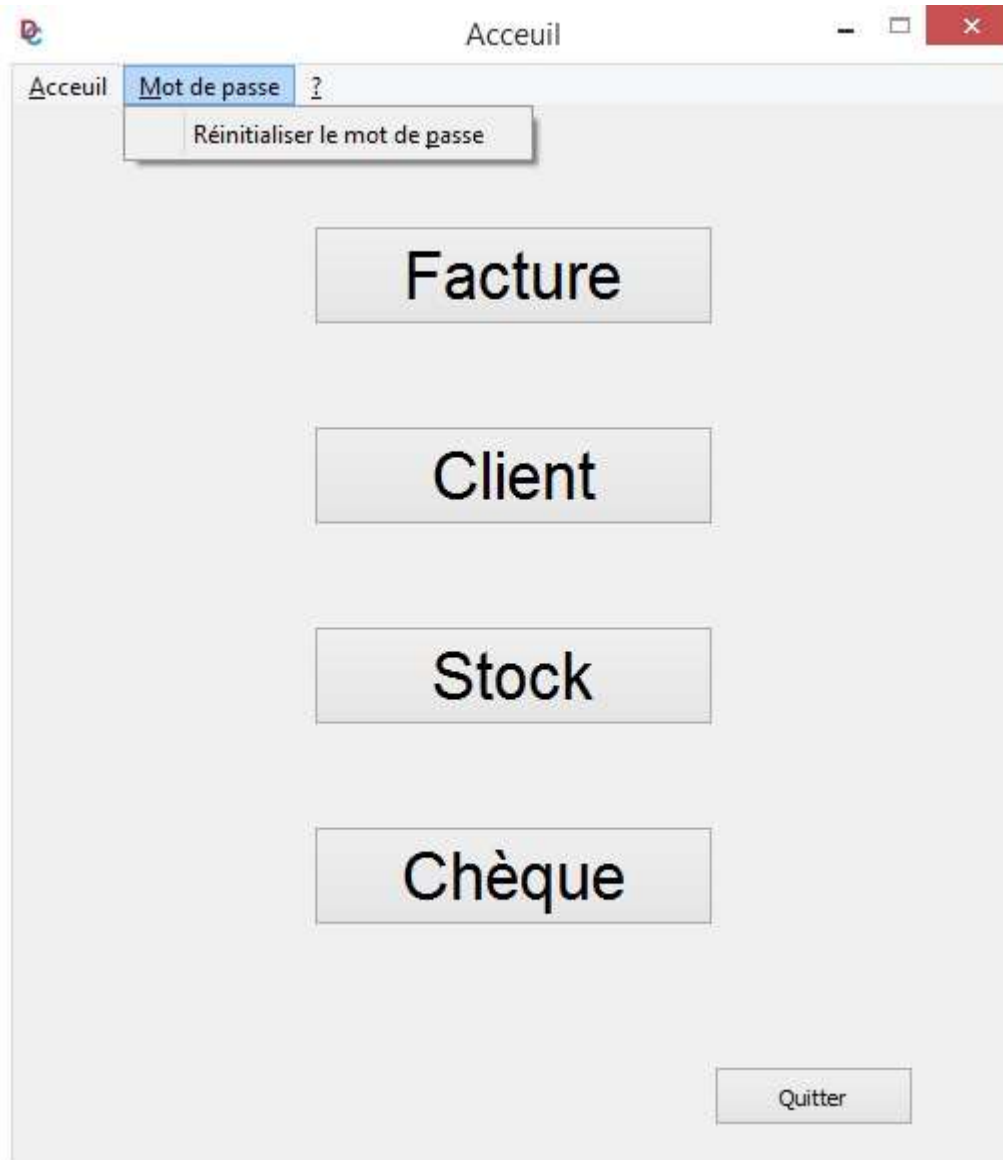


Figure 30 : Menu mot de passe

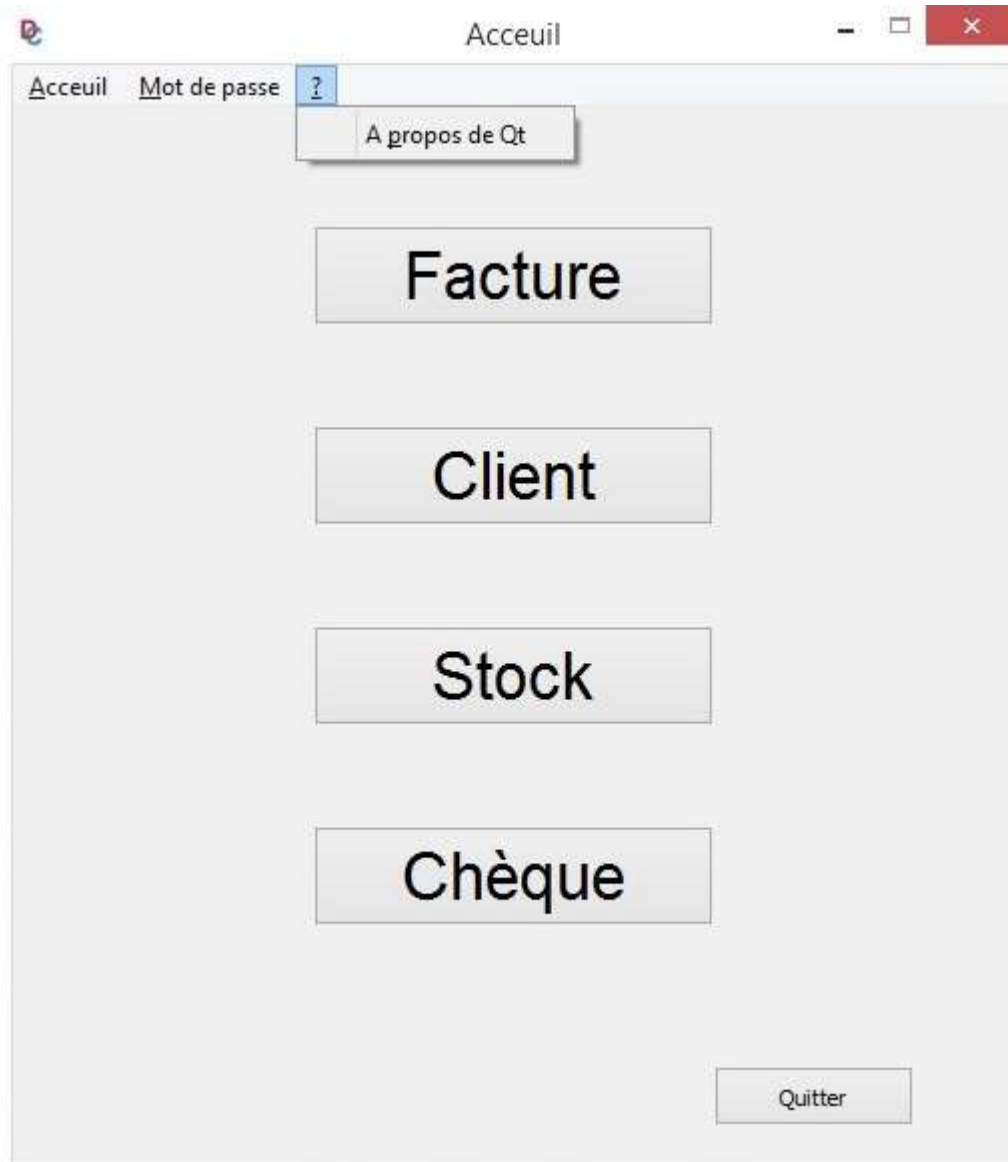


Figure 31 : Menu à propos de Qt

Pour le menu "A propos de Qt", il nous donne seulement des informations sur le logiciel utilisé comme nous pouvons voir dans la figure suivante :

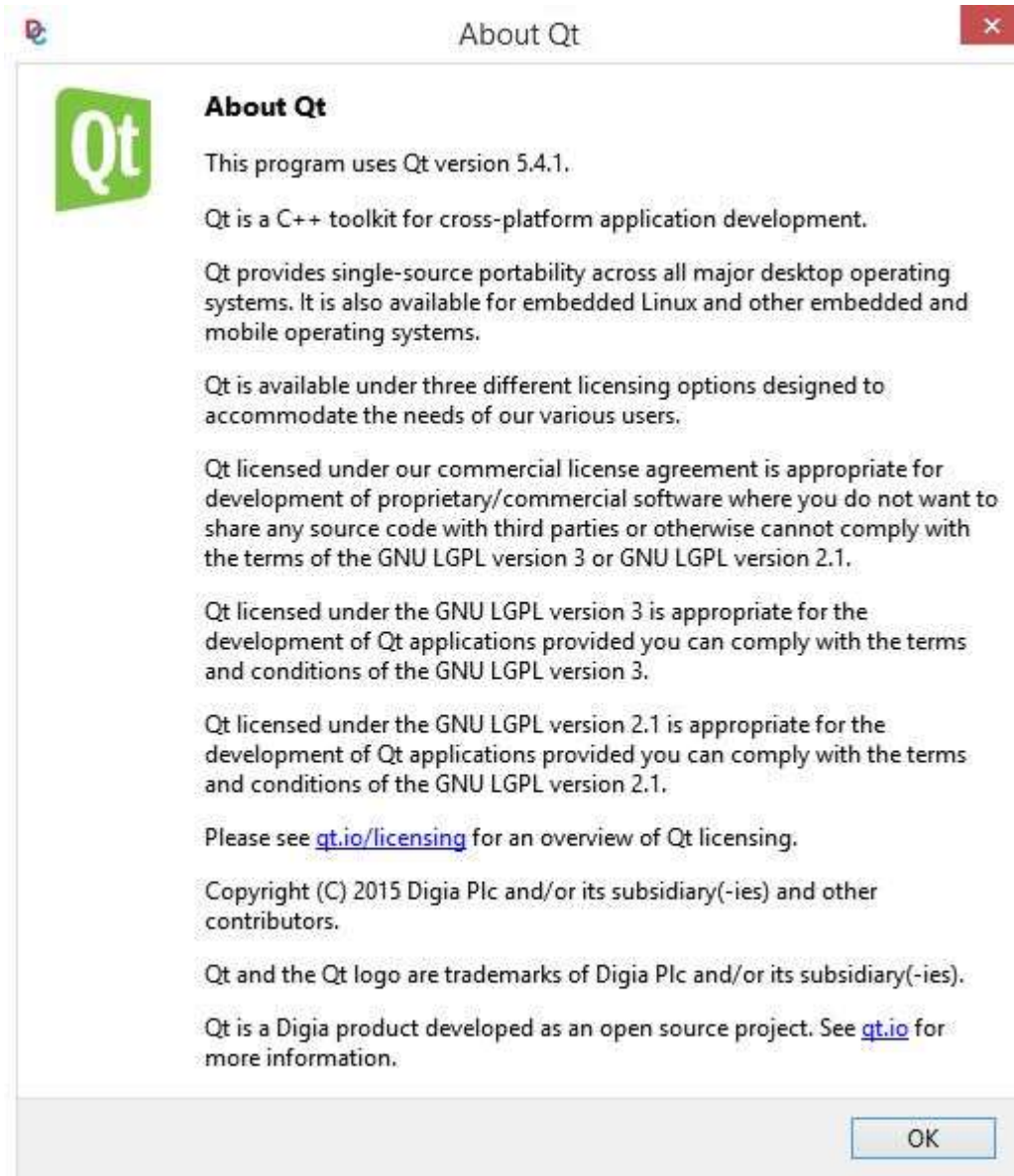


Figure 32 : Fenêtre à propos de Qt

Conclusion :

Nous avons vu dans ce chapitre les outils de développement et les étapes de la réalisation.



CONCLUSION GÉNÉRALE

Pendant la période de notre stage qui a duré deux mois, on a réalisé une application de gestion d'inventaire pour l'entreprise Dar du commerce moderne en utilisant de nouveaux outils et langages non vus lors du cursus universitaire telles que Qt et SQLite.

On a trouvé l'expérience très intéressante et enrichissante, où on a approfondie nos compétences, ce qui nous a permis de découvrir la partie opérationnelle du monde professionnel de développement.

Cette expérience nous a permis de mettre en pratique ce qu'on a appris en modélisation, et plus particulièrement, la modélisation avec le langage UML.

Ce stage a vraiment confirmé nos ambitions dans le domaine du développement informatique, même s'il nous reste encore beaucoup à apprendre.



WEBGRAPHIE

- 1) [http://fr.wikipedia.org/wiki/UML_\(informatique\)](http://fr.wikipedia.org/wiki/UML_(informatique))
- 2) <http://fr.wikipedia.org/wiki/Modèle-vue-contrôleur>
- 3) <http://openclassrooms.com/courses/programmez-avec-le-langage-c/introduction-a-gt>
- 4) <http://www.finalclap.com/faq/180-sqlite-definition>
- 5) <http://fr.wikipedia.org/wiki/SQLite>



BIBLIOGRAPHIE

Cours Ilham Chakir Génie logiciel chapitre2 le processus logiciel page 78.

Cours UML de Mr. BENNABOU Abderrahim