

AP2 : Gestion de stock

Titre de la séquence :	AP 2 : Gestion de stock	Auteur :	RINALDI Axel
Objectifs pédagogiques			
B2 compétences : <ul style="list-style-type: none"> - Participer à la conception de l'architecture d'une solution applicative - Exploiter les ressources du cadre applicatif (framework) - Intégrer en continu les versions d'une solution applicative - Réaliser les tests nécessaires à la validation ou à la mise en production d'éléments adaptés ou développés Savoirs associés: <ul style="list-style-type: none"> - Méthodes, normes et standards associés au processus de conception et de développement d'une solution applicative - Architectures applicatives : concepts de base et typologies - Interfaces homme-machine : principes ergonomiques, techniques de conception, d'évaluation et de validation. - Concepts de la programmation événementielle : techniques de gestion des événements et exploitation de bibliothèques de composants graphiques - Caractéristiques des formats de données : structurées ou non - Fonctionnalités d'un outil de gestion de projets. - Fonctionnalités avancées d'un environnement de développement. - Techniques de gestion de versions 			
Pré requis :	- Algorithmique, programmation orientée objet, langage JAVA ou C#		
Activités :	- Programmation événementielle et objet avec Java ou C# - Concevoir tout ou partie d'un logiciel à partir d'un cahier des charges - Connaissance des outils logiciel : IDE, SGBD, ... - Mise en œuvre de l'outil de gestion de version GIT		
Public :	BTS	Durée prévue :	Semestre n°4

Table des matières

1. Présentation du projet.....	3
1.1. Contexte.....	3
1.2. Choix du langage de programmation.....	3
1.3. Choix du SGBD.....	3
1.4. Modalités et consignes.....	3
2. Cahier des charges.....	4
2.1. Noyaux de l'application.....	4
2.1.1. Modèle de données.....	4
2.1.1.1. Articles.....	4
2.1.1.2. Catégories d'article.....	5
2.1.1.3. Les fabricants.....	5
2.1.1.4. Les unités.....	5
2.1.1.5. Les dépôts.....	5
2.1.1.6. Les mouvements de stock.....	5
2.1.1.6.1. Exemple de scénario :.....	6
2.1.2. IHM.....	6
2.1.2.1. Menu de l'application.....	7
2.1.2.1.1. Menu paramètres :.....	7
2.1.2.1.2. Menu données :.....	7
2.1.2.1.3. Menu stock :.....	7
2.1.2.2. Liste multicritères des articles.....	7
2.1.2.3. Formulaire Article.....	8
2.1.2.4. Liste des catégories, unités et des fabricants.....	9
2.1.2.5. Liste des dépôts.....	10
2.1.2.6. Formulaire pour un dépôt.....	10
2.1.2.7. Liste des mouvements pour un article dans un dépôt.....	11
2.1.2.8. Inventaire.....	11
3. Aspects techniques.....	11
3.1. Utilisation d'un modèle de type MVC.....	11
3.2. Aspects sécuritaire.....	12
3.3. Gestion des versions.....	12
4. Documents attendus.....	12

1. Présentation du projet

1.1. Contexte

L'objet de cet atelier professionnel est la réalisation d'une application de gestion de stock destinée à devenir un module d'un ERP. (Enterprise Resource Planning). Avec l'avènement du commerce en ligne ou bien la gestion des pièces et composants pour la fabrication de produits finis la gestion de stock est devenue un enjeu majeur dans le process de beaucoup d'entreprise.

Les entreprises fabriquant des objets complexes, voitures, avions, machines outils ou produits électroménager et celles gérant la distribution de produits de vente en ligne doivent pouvoir gérer de manière efficace leur stock afin d'être en mesure de livrer des pièces ou de fabriquer des produits finis en temps et en heure malgré les aléas de livraison et de fabrication. Les lieux de stockage dont certains peuvent être très grand, sont en nombre et de taille limités. Il doivent donc être l'objet d'une optimisation qui, compte tenu du très grand nombre d'articles stockés, ne peut se faire qu'avec l'aide d'un logiciel spécialisé.

1.2. Choix du langage de programmation

Le présent atelier vous propose de réaliser par étapes un logiciel de gestion de stock dans le cadre d'une application lourde (code compilé et installé sur un ordinateur) en langage Java ou C# selon votre choix.

1.3. Choix du SGBD

Cette application sera interfacée avec un SGBD de type SQLSERVER ou SQLSERVER EXPRESS pour la gestion des données et proposera une IHM (Interface Homme Machine) adaptée à la gestion de stock.

1.4. Modalités et consignes

Vous travaillerez seul sur votre projet et devrez rendre un produit fini fonctionnel répondant au cahier des charges ci-dessous.

Ce dernier est organisé en plusieurs parties correspondants chacune à une évolution du projet.

Cette version du document ne présente que la première version du cahier des charges du logiciel, c'est à dire, celle consacrée au **noyaux de l'application**. Cette version est la seule qui soit obligatoire.

Des évolutions suivront pour ceux qui avancent plus vite et souhaitent améliorer leurs compétences. Elles seront comptabilisées sous la forme de bonus.

2. Cahier des charges

Le cahier des charges est organisé en versions ou évolutions du logiciel. Il faudra réaliser ces versions dans l'ordre en commençant par la première qui décrit le noyau de l'application.

2.1. *Noyaux de l'application*

L'application permet de gérer le stock d'articles dans un ou plusieurs dépôts.

2.1.1. Modèle de données

L'ensemble du modèle de données est décrit ci-dessous. Il s'articule autour de la notion d'article, de dépôt de stock et de mouvement de stock. Des entités annexes viennent compléter la gestion des articles.

L'ensemble des données devra être géré avec le SGBD SQLSERVER de Microsoft dans le cadre d'une base de données nommée **BD_STOCK**. Vous produirez un fichier **bd_stock.sql** contenant les requêtes SQL de création des tables, contraintes d'intégrité et vues utilisées dans cette version du logiciel.

2.1.1.1. **Articles**

Un article correspond à un type d'objet ou produit dont on veut gérer le stock. Par exemple, l'ensemble des Iphones 15 est représenté par un article qui regroupe l'ensemble des propriétés utiles à la gestion du stock d'un téléphone du type Iphone 15 de la marque Apple.

Chaque article est identifié par une **référence unique** sous la forme d'une chaîne de caractères d'une taille maximum de 20 caractères alphanumériques, possède un **libellé** sous la forme d'une chaîne de caractères d'une taille maximale de 120 caractères alphanumériques, un **identifiant de fabricant** sous la forme d'une chaîne de caractères d'une taille maximale de 10 caractères alphanumériques référant un **fabriquant** et un **identifiant de catégorie** sous la forme d'une chaîne de caractères d'une taille maximale de 5 caractères alphanumériques référant une **catégorie d'article** à laquelle l'article appartient.

Par exemple, le téléphone d'Apple de type « *Iphone 15 pro* » possède le libellé « *Téléphone Iphone 15 Pro* » est associé à la marque « *Apple* » et à la catégorie « *Smartphone* ».

Dans le cas où l'on gère pour un téléphone Iphone 15 Pro la couleur et la capacité mémoire alors il faudra créer autant d'articles différents que de combinaisons de valeur. Si les téléphones « Iphone 15 pro » peuvent se décliner selon les 4 couleurs : or, acier, blanc ou noir et les 3 capacités de mémoire 126 Go, 256 Go et 512 Go il faudra créer $3 \times 4 = 12$ articles différents, un pour chaque combinaison de couleur et de capacité mémoire avec une référence différente et un libellé qui reprendra sous forme abrégé la couleur et la capacité mémoire.

Selon le type d'article la quantité de ce dernier peut être gérée en « unité » dans le cas d'objets distincts dénombrables. S'il s'agit d'un liquide, on utilisera plutôt une unité de volume (litre ou ml), pour une poudre ou des cacahuètes on préférera une unité de masse (kg ou g), etc. Aussi, dans le cadre de la gestion de stock il semble judicieux d'associer à chaque article une unité de mesure pour la quantité. Cette dernière sera encodée sous la forme de 3 caractères alphabétiques et référencera une unité.

2.1.1.2. Catégories d'article

Chaque article appartient toujours à une seule catégorie qui permet de regrouper ces derniers comme « Smartphone » pour des téléphones ou « Légumes » pour des produits agroalimentaires de type légume. Chaque catégorie est identifiée par un **code de catégorie** unique sous la forme d'une chaîne de caractères d'une taille maximale de 5 caractères alphanumériques et possède un **libellé** sous la forme d'une chaîne de caractères d'une taille maximale de 50 caractères alphanumériques.

Il doit être impossible de supprimer une catégorie qui soit déjà référencée par au moins un article. En revanche il doit être possible de modifier le code d'une catégorie même si cette dernière est déjà référencée par des articles. Dans ce cas, les valeurs du code de catégorie au niveau des articles doivent automatiquement être modifiées en conséquence.

2.1.1.3. Les fabricants

Chaque article est toujours fabriqué par un seul fabricant. Par exemple, un téléphone « Iphone » est fabriqué uniquement par la société Apple. Deux articles semblables, par exemple deux télévisions de même taille et offrant les mêmes caractéristiques peuvent être néanmoins fabriquées par deux sociétés différentes, elles auront par conséquent chacune une référence unique.

Chaque fabricant possède une **référence unique** sous la forme d'une chaîne de caractères d'une taille maximale de 10 caractères alphanumériques. Il est aussi associé à un **nom d'entreprise** sur 50 caractères et à celui d'une **marque** sur 50 caractères.

Par exemple, l'entreprise « Unilever » possède plusieurs marques dont « Dove », « Rexona » ou « Cif ». Ces trois marques donneront lieu à trois fabricants différents ayant le même nom d'entreprise.

Comme pour les catégories, il doit être impossible de supprimer un fabricant qui soit déjà référencé par au moins un article. En revanche il doit être possible de modifier la valeur du référent fabricant même si ce dernier est déjà référencé par des articles. Dans ce cas, les valeurs du référent fabricant au niveau des articles doivent automatiquement être modifiées en conséquence.

2.1.1.4. Les unités

Chaque article est toujours associé à une seule unité pour la gestion des quantités. L'ensemble des unités utilisables doit être géré séparément dans une entité identifiée par un **code unité** unique sur 3 caractères alphanumériques et un **libellé** donnant le nom de l'unité sur maximum 30 caractères, par exemple « ml » pour « millilitre ».

Il doit être impossible de supprimer une unité qui soit déjà référencée par au moins un article.

2.1.1.5. Les dépôts

Un dépôt de stockage est un lieu dont le rôle est de permettre de stocker des articles différents. Ces derniers peuvent être de tailles diverses et situés dans des lieux différents.

Chaque dépôt est identifié par un **numéro** unique et possède un **nom** sur 50 caractères maximum. Pour connaître la position d'un dépôt on lui associe une **ville** sur 50 caractères et un **pays** sur 50 caractères. Il se peut que plusieurs dépôts soient dans la même ville aussi, chaque dépôt dispose de coordonnées GPS sous la forme d'une **latitude** et d'une **longitude**.

2.1.1.6. Les mouvements de stock

La gestion d'un stock se fait en général au travers de la notion de mouvements de stock, c'est à dire, par la prise en compte des entrées et des sorties de stock. Pour chaque article on gère deux types de mouvement de stock. Le premier de type « INV » pour inventaire ; il permet de définir le niveau de stock d'une référence d'article dans un dépôt particulier à une date et

heure précise. Le deuxième, de type « **MVT** », correspond à une entrée ou une sortie de stock selon le signe de la quantité associée d'article ajoutée ou retirée du stock. Une quantité **positive** correspond à un ajout au stock d'un nombre d'article donnée par cette dernière à une date et heure précise ; il s'agit d'une **entrée de stock**. En revanche une quantité **négative** correspond à un **retrait du stock** d'autant d'articles que mentionné par cette dernière à un instant donné ; il s'agit d'une **sortie de stock**.

Un mouvement de stock est identifié par la **référence de l'article** concerné, le **numéro du dépôt** dans lequel il s'opère et la **date et heure** du mouvement (utiliser un champ de type Date/Heure). Chaque mouvement possède en plus un champ **type** sur 3 caractères (type CHAR et non VARCHAR) dont les valeurs sont « **INV** » pour inventaire et « **MVT** » pour mouvement et un champ **qte** de type réel (float) pour la quantité exprimée dans l'unité associée à l'article.

2.1.1.6.1. Exemple de scénario :

On considère l'article « *Tringle à rideau* » ayant comme référence « *X079077* », le dépôt n° 818 et les 4 mouvements de stock suivants :

Ref article	Dépôt	Type	Date et heure	Qte
X079077	818	INV	22/12/2024 16:32:23	7
X079077	818	MVT	23/12/2024 09:10:27	-4
X079077	818	MVT	27/12/2024 10:45:53	5
X079077	818	MVT	29/12/2024 15:07:37	-2

Le niveau de stock d'un article dans un dépôt particulier est donnée à un instant précis par le cumul des quantités entre cet instant et le dernier mouvement de type « **INV** » inventaire.

Ainsi la quantité d'article ayant la référence « *X079077* » présent dans le dépôt n° 818 est de 7 le 23/12/2024 à 08:00:00, de 3 le 24/12/2024 à 23:00:00 et de 6 le 01/01/2025 à 00:00:01.

2.1.2. IHM

La partie IHM (Interface homme – machine) doit permettre d'accéder aux fonctionnalités du logiciel. Parmi ces dernières, elle doit permettre de mettre à jour un article, c'est à dire créer, modifier et supprimer ce dernier dans le respect des contraintes d'intégrité et des règles inhérentes à la gestion de stock.

Comme les articles sont associés à des fabricants, des catégories et utilisent des unités pour gérer les quantités ces données annexes doivent aussi être accessibles en modifications par l'utilisateur au moyen de formulaires ou listes ad hoc.

L'interface graphique du logiciel doit aussi permettre de gérer les dépôts de stockage et proposer une liste et un formulaire de mise à jour pour ces derniers.

Le coeur du logiciel est de pouvoir gérer des entrées ou sorties de stock et de faire des inventaires. Pour cela, le logiciel doit proposer des listes et/ou formulaire ad hoc.

Enfin, le plus important est de pouvoir visualiser les niveaux de stock à une date et heure donnée afin d'organiser les approvisionnements et la distribution. Le logiciel proposera une vue sous forme de liste multicritères qui se basera sur une ou plusieurs vues au niveau de la base de données.

2.1.2.1. Menu de l'application

L'ensemble de ces fonctionnalités doivent être accessibles au moyen d'une barre de menu proposant les menus déroulants suivants :

2.1.2.1.1. Menu paramètres :

- Liste des catégories d'article → visualisation, création, mise à jour, suppression de catégorie.
- Liste des unités → visualisation, création, mise à jour, suppression d'unité.

2.1.2.1.2. Menu données :

- Liste multicritère de recherche d'un fabricant → accès au formulaire de mise à jour d'un fabricant.
- Liste multicritère de recherche d'un ensemble d'articles → accès au formulaire de mise à jour d'un article.
- Liste multicritère de recherche d'un ensemble de dépôts → accès au formulaire de mise à jour d'un dépôt.

2.1.2.1.3. Menu stock :

- Liste multicritères des niveaux de stock → affichage des niveaux de stock à une date et heure donnée → possibilité de réaliser des entrées et sortie de stock.
- Liste multicritères pour inventaire → → affichage des niveaux de stock à une date et heure donnée → possibilité de modifier directement le niveau de stock par l'ajout de mouvement de type « INV »
- Liste multicritères des mouvements de stock → visualisation des mouvements de stock pour un ensemble d'articles → possibilité de créer, modifier ou supprimer un mouvement.

La suite de cette partie propose des modèles de formulaire ou de liste tabulaire pour certaines fonctionnalités importantes.

2.1.2.2. Liste multicritères des articles

A partir du menu des données il est possible d'accéder à la liste multicritère des articles dont l'aspect est présenté ci-dessous.

Liste des articles

Recherche par catégorie: *
 Recherche par fabricant: *
 Recherche par dépôt: *
 Recherche par libellé: *

☐ Classement décroissant

Référence	Libellé	Catégorie	Fabricant	Unité
X00	Iphone 15 option 0	Smartphone	Apple	Unité
X01	Iphone 15 option 1	Smartphone	Apple	Unité
X02	Iphone 15 option 2	Smartphone	Apple	Unité
X03	Iphone 15 option 3	Smartphone	Apple	Unité
X04	Iphone 15 option 4	Smartphone	Apple	Unité
X05	Iphone 15 option 5	Smartphone	Apple	Unité
X06	Iphone 15 option 6	Smartphone	Apple	Unité
X07	Iphone 15 option 7	Smartphone	Apple	Unité
X08	Iphone 15 option 8	Smartphone	Apple	Unité
X09	Iphone 15 option 9	Smartphone	Apple	Unité
X010	Iphone 15 option 10	Smartphone	Apple	Unité
X011	Iphone 15 option 11	Smartphone	Apple	Unité
X012	Iphone 15 option 12	Smartphone	Apple	Unité
X013	Iphone 15 option 13	Smartphone	Apple	Unité
X014	Iphone 15 option 14	Smartphone	Apple	Unité
X015	Iphone 15 option 15	Smartphone	Apple	Unité
X016	Iphone 15 option 16	Smartphone	Apple	Unité
X017	Iphone 15 option 17	Smartphone	Apple	Unité
X018	Iphone 15 option 18	Smartphone	Apple	Unité
X019	Iphone 15 option 19	Smartphone	Apple	Unité

Par défaut le formulaire affiche la liste des 20 premiers articles.

La partie entête contient des champs de recherche ou filtrage concernant la liste d'articles que l'on désire obtenir. La recherche peut se faire à partir d'une **catégorie** et/ou d'un **fabricant** (ici une **entreprise**) et/ou d'un **dépôt** et/ou d'une partie du **libellé** en utilisant les caractères '*' ou '%' remplaçant une partie de chaîne de caractère.

La case à cocher « **Classement décroissant** » permet d'inverser l'ordre de tri.

En double cliquant sur un ligne on accède au formulaire de mise à jour de l'article sélectionné. Au sortir du formulaire la liste est mise à jour tout en restant à une position de scrolling identique à celle avant modification.

Le bouton « **Nouvel article** » crée un nouvel article et ouvre le formulaire de mise à jour de l'article.

2.1.2.3. Formulaire Article

L'application permet la consultation, mise à jour et suppression d'un article au travers du formulaire FrArticle dont l'aspect est présenté ci-dessous.

The screenshot shows a window titled 'Article' with a yellow header. Below the header are two search filters: 'Recherche par référence:' and 'Recherche par libellé:', both with dropdown menus. The main form contains the following fields:

- Référence:** A text input containing 'IP034506' and a button with three dots '...'.
- Libellé:** A text input containing 'Iphone 15 Pro OR 256'.
- Catégorie:** A dropdown menu showing 'Smartphone'.
- Fabricant:** A dropdown menu showing 'Apple'.
- Unité:** A dropdown menu showing 'unité'.

At the bottom of the form are three buttons: 'Annuler', 'Enregistrer', and 'Supprimer'. To the right of the form is a table with two columns: 'Dépôt' and 'Stock'. The table is currently empty.

Le bouton **Annuler** permet de quitter le formulaire sans enregistrer les modifications.

le bouton **Enregistrer** vérifie la justesse des données saisies et enregistre en base de données les modifications saisies. En cas d'erreur au moment de la vérification, un message d'erreur est affiché et les champs concernés affichés en rouge.

Le bouton **Supprimer** permet de supprimer l'article affiché. En cas d'erreur, article référencé dans un stock, une confirmation est explicitement demandée pour forcer la suppression.

Les champs **Référence** et **Libellé** sont en saisie libre avec pour le premier la contrainte de devoir mettre une valeur unique. Ceci est vérifié au moment de l'enregistrement en base de données. Concernant le champ **Référence**, le bouton « ... » complète la référence par un numéro unique permettant de rendre cette dernière unique.

Les champs **Recherche par référence** et **Recherche par libellé** permettent de rechercher rapidement un article. Quand l'utilisateur tape le début d'une référence ou d'un libellé la liste de choix propose alors le choix de toutes les valeurs commençant par ce qui est tapé dans l'ordre lexicographique. Cette facilité de recherche est à développer après que le reste du formulaire fonctionne correctement.

La liste de droite affiche les dépôts dans lesquels l'article est stocké avec le niveau de stock à la date et heure courante.

2.1.2.4. Liste des catégories, unités et des fabricants

Les listes des catégories, unités et fabricants obéissent aux principes que la liste d'articles et le formulaire de mise à jour des articles.

Concernant les catégories et les unités il peut être possible de mettre à jour ces dernières directement à partir de la liste sans passer par un formulaire.

2.1.2.5. Liste des dépôts

La liste des dépôts est une liste multicritères permettant de visualiser l'ensemble des dépôts et aussi de filtrer ces derniers à partir d'un pays, d'une ville, d'un article, d'un fabricant et d'une catégorie.

La liste possède un bouton permettant de créer un nouveau dépôt.

En cliquant sur un dépôt on peut accéder au formulaire **FrDepot** permettant de modifier ou supprimer un dépôt s'il n'est pas associé à des articles.

2.1.2.6. Formulaire pour un dépôt

Le formulaire **FrDepot** permet de mettre à jour et/ou de supprimer un dépôt. Concernant les champs affichant le pays ou la ville, ces derniers sont des « combobox » affichant respectivement la liste des pays déjà utilisés et des villes déjà utilisées par ordre alphabétique. Ces combobox autorisent la saisie directe afin de permettre de définir un nouveau pays ou une nouvelle ville.

Concernant le numéro identifiant d'un dépôt, ce dernier peut être généré via la base de données à l'aide d'une séquence SQL.

L'entête du formulaire propose des champs de recherche à partir d'une partie du numéro identifiant ou bien d'une partie du nom au moyen de combobox.

Liste des dépôts avec niveau de stock

Liste des dépôts avec niveau de stock affiche l'ensemble des dépôts et articles avec le niveau de stock à la date et heure donnée par le champ « Date heure » sous la forme suivante :

The screenshot shows a window titled "Niveaux de stock". It contains several filter fields: "Dépôt:" with a dropdown menu showing "Colmar", "Fabricant:" with a dropdown menu showing "*", and "Catégorie:" with a dropdown menu showing "*". To the right of these are two input fields for "Stock <= à:" and "Stock >= à:", and a date/time field "Date et heure:" showing "03/01/2025 12:45:25". Below these fields is a table with four columns: "Dépôt", "Article", "Quantité", and "Unité". The table is currently empty.

Il est possible de filtrer l'affichage par dépôt et/ou fabricant et/ou catégorie et/ou niveau de stock mini ou maxi. Il peut être judicieux de pouvoir filtrer par ville ou par pays.

Une autre fonctionnalité serait d'avoir la somme des niveaux de stock pour chaque article sur l'ensemble des dépôts d'un pays ou d'une ville.

En double cliquant sur une ligne on doit pouvoir accéder à la liste des mouvements associés à un article et un dépôt.

2.1.2.7. Liste des mouvements pour un article dans un dépôt

Ce formulaire affiche la liste des mouvements pour un article et un dépôt. Il doit permettre de modifier en direct la valeur d'un mouvement, sa date et son heure. Il permet aussi de supprimer et de créer un mouvement de stock.

L'entête du formulaire doit offrir des mécanismes de filtrage utiles pour l'utilisateur à l'image de ceux proposés par les autres formulaires.

2.1.2.8. Inventaire

Le formulaire d'inventaire doit permettre de réaliser un inventaire de stock, c'est à dire offrir la possibilité de saisir pour un ensemble d'articles dans un dépôt des niveaux de stock à une date et heure définie dans l'entête du formulaire. Chaque saisie engendre alors un mouvement de type « **INV** ».

L'entête du formulaire permet, en plus de définir une date et une heure, de filtrer les données par dépôt, ville, pays, catégorie d'article ou fabricant.

Le plus simple est d'afficher la liste avec le niveau de stock à la date et heure définie en entête et de permettre la modification de la valeur dans la même colonne. Un bouton « **Valider les niveaux saisis** » permet alors de créer pour tous les niveaux de stock modifiés des mouvements de type « **INV** ».

3. Aspects techniques

Les aspects techniques énumérés ci-dessus sont donnés plus à titre de conseil que d'obligation mais ils devraient vous permettre de simplifier le code en général, de rendre votre application plus lisible et plus facilement maintenable tout en permettant des évolutions pour le futur.

3.1. Utilisation d'un modèle de type MVC

Le modèle de conception de type MVC (Modèle - Vue – Contrôleur) est un standard que vous devez maîtriser en matière de développement. Aussi il vous est fortement conseillé de mettre en place un tel modèle.

L'ensemble des formulaires et listes sont des vues des objets gérés par le logiciel. Les objets correspondant au modèle de données décrit plus haut sont implémentés sous la forme de classes métiers, par exemple la classe Article pour gérer un article, etc.

Pour ceux qui choisissent de développer leur projet en Java je vous renvoie aux liens suivants concernant l'utilisation de la librairie Swing avec un modèle de conception MVC :

<https://jo-nfp121.e-monsite.com/medias/files/nfp121-cours-04-2-swing-mvc.pdf>

<https://miashs-www.u-ga.fr/prevert/Prog/Java/swing/JTable.html>

3.2. Aspects sécuritaire

Votre application étant un logiciel compilé et installé directement sur un ordinateur par un administrateur responsable de la sécurité elle présente moins de risque qu'une WEB application. Néanmoins, l'accès à une base de données doit toujours être l'objet d'une attention particulière. Aussi, il vous est demandé d'autoriser l'accès à la base de données par le logiciel qu'au travers d'un compte utilisateur bien identifié et possédant les droits minimum.

En ouverture l'utilisateur devra s'authentifier pour utiliser l'application. Cette dernière doit permettre à un utilisateur de modifier son mot de passe et ce dernier ne doit à aucun moment être stocké dans la base de données. L'authentification doit se faire via une empreinte SHA associé à du salage généré aléatoirement pour chaque compte.

3.3. Gestion des versions

Il vous est imposé de gérer les versions et évolutions de votre développement à l'aide d'un outil de gestion de version de type GIT.

4. Documents attendus

- 1) Vous devrez livrer un code source et un exécutable fonctionnel.
- 2) Le code source doit absolument être documenté et commenté suffisamment pour permettre à un autre technicien d'opérer une modification ou une correction de bogue. Ce document sera exigé pour l'épreuve pratique E7 de l'examen du BTS si vous avez opté de présenter votre projet à cette épreuve. La rédaction de ce dernier durant le développement de votre projet vous fera gagné un temps précieux.
- 3) Vous utiliserez un outil de génération de documentation automatique style **Javadoc** pour générer la documentation technique du logiciel.
- 4) Vous rendrez un **diagramme UML de classe** présentant l'ensemble des classes composant votre développement ainsi qu'un **modèle relationnel de données** décrivant les tables et contraintes de la base de données.
- 5) Enfin, vous rédigerez un documentation à l'attention de l'utilisateur lui permettant d'utiliser au mieux votre application.