

Table des matières

1	Introduction.....	3
1.1	Cahier des charges.....	3
1.2	Outils utilisé :.....	3
1.3	Convention de nommage.....	4
1.4	Rôle du groupe	4
1.5	Point important	4
2	Notre Méthode agile	4
2.1	Schémas de cycle agile.....	5
2.2	Schéma	5
3	MCD – GestionStock.....	6
4	MLD – GestionStock	7
5	Thésaurus	8
6	Installation et mise en place des outils	11
6.1	Installation JavaFX	11
6.2	Lié le SceneBuilder avec Eclipse	12
6.3	Installation JFoenix dans SceneBuilder	14
7	Création d'un JAR exécutable et d'un installateur	15
8	Conclusion	20

1 Introduction

Ce projet a pour objectif d'utiliser les outils et méthodes vus depuis le début de ce bachelier dans le but de produire une application la plus complète et professionnelle possible. Le logiciel en question devra disposer des fonctionnalités suivantes :

- Gérer un stock d'article et les informations associés à chacun d'entre eux.
- Générer des codes-barres pour chaque article et être capable de les lire.
- Gérer la réception des commandes.
- Garder une trace des déplacements des articles en interne (qui a pris quoi).

Pour y arriver, le groupe devra mettre en place sa propre méthode agile et respecter un échéancier mis à jour régulièrement.

Pendant les différents cycles du développement, une série de tests permettra de valider les étapes au fur et à mesure ainsi que de consolider le code afin d'éviter des bugs ou autres problèmes indésirables.

Le groupe devra également réaliser une documentation utilisateur pour permettre à ce dernier de manier correctement le logiciel et une documentation technique qui expliquera les différentes parties du code au cas où il serait nécessaire de le retravailler.

1.1 Cahier des charges

1. Réflexion autour du projet
2. Création d'une méthode agile
3. Réflexion et mise en place d'un MCD/MLD de la base de données.
4. Réflexion sur l'architecture du programme.
5. Réflexion sur les outils utilisés pour le programme.
6. Développement du programme.
7. Création de documentation utilisateur et technique.
8. Test, mise en pratique et avis des clients.

1.2 Outils utilisé :

- Eclipse : IDE de développement java
- Discord : outil de communication vocal et écrit
- Google Drive : partage et mise en commun du code
- JavaFX : outils graphique pour java

1.3 Convention de nommage.

- Classe : Première lettre majuscule et début des prochains mots (MaClasse)
- Variable : Première lettre en minuscule et majuscule début des prochains mots (maVariable)
- Fonction : Première lettre en minuscule et majuscule début des prochains mots (maFonction)
- Nom de fichier : Première lettre majuscule et début des prochains mots (MonFichier)

1.4 Rôle du groupe

- Client : ECI (HILGERS Martine)
- Chef : J.NINFORGE
- Coach : J.NINFORGE
- Testeur : HERMASSI Mounir et SABIA Miguel
- Développeur : HERMASSI Mounir et SABIA Miguel
- Consultant : J.NINFORGE
- Tortionnaire : J.NINFORGE

1.5 Point important

- Travail de groupe
- Bonne entente
- Respect de l'échéancier
- Bonne humeur

2 Notre Méthode agile

Pour notre projet qui est une gestion de stock avec utilisation de code barre. Nous allons créer une méthode basée sur la méthode Crystal.

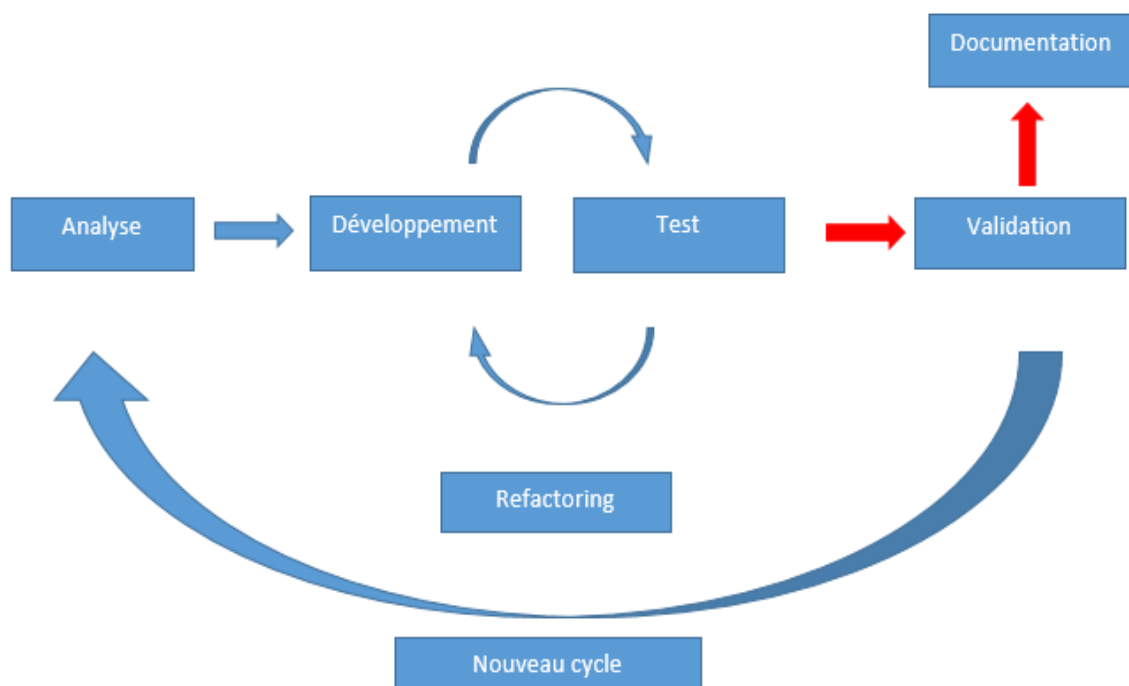
Nos cycle se compose d'une phase d'analyse, suivit du développement ou d'une mise en pratique. Nous avons ensuite une phase de test qui peut conduire a du refactoring ou une remise a 0 du cycle en cours les tests sont vraiment mauvais. Nous finissons par une validation du client et une documentation en anglais du cycle fini.

Nous adapterons bien entendu la méthode en fonction des retards, modifications dans la conception ou autres aléas que nous pourrions rencontrer.

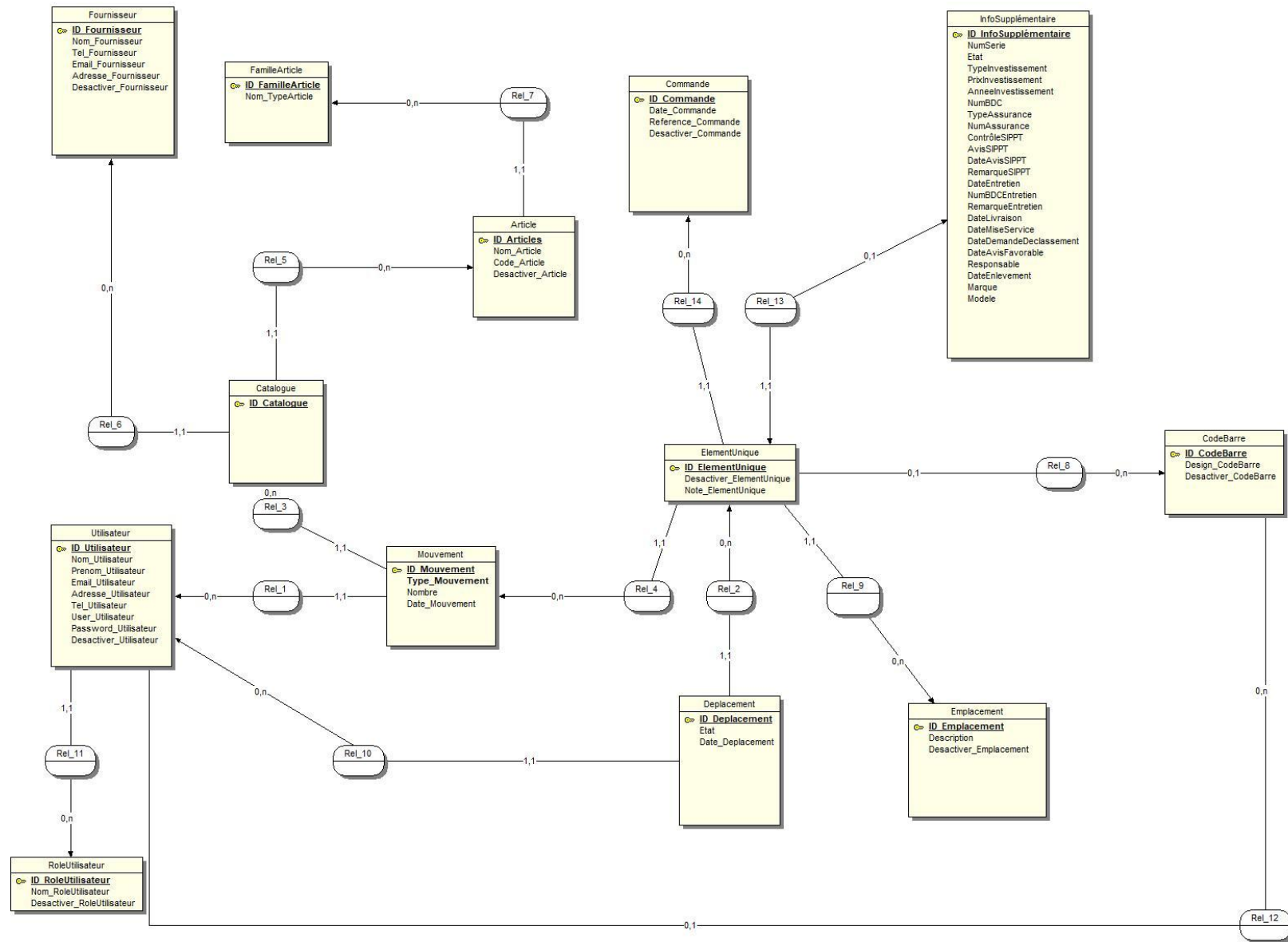
2.1 Schémas de cycle agile.

- Analyse et réflexion du travail a effectué
- Ebauche du travail, programmation ou analyse
- Premier test du travail
- Refactoring, si nécessaire
- Test final
- Validation du travail du cycle par le client
- Documentation

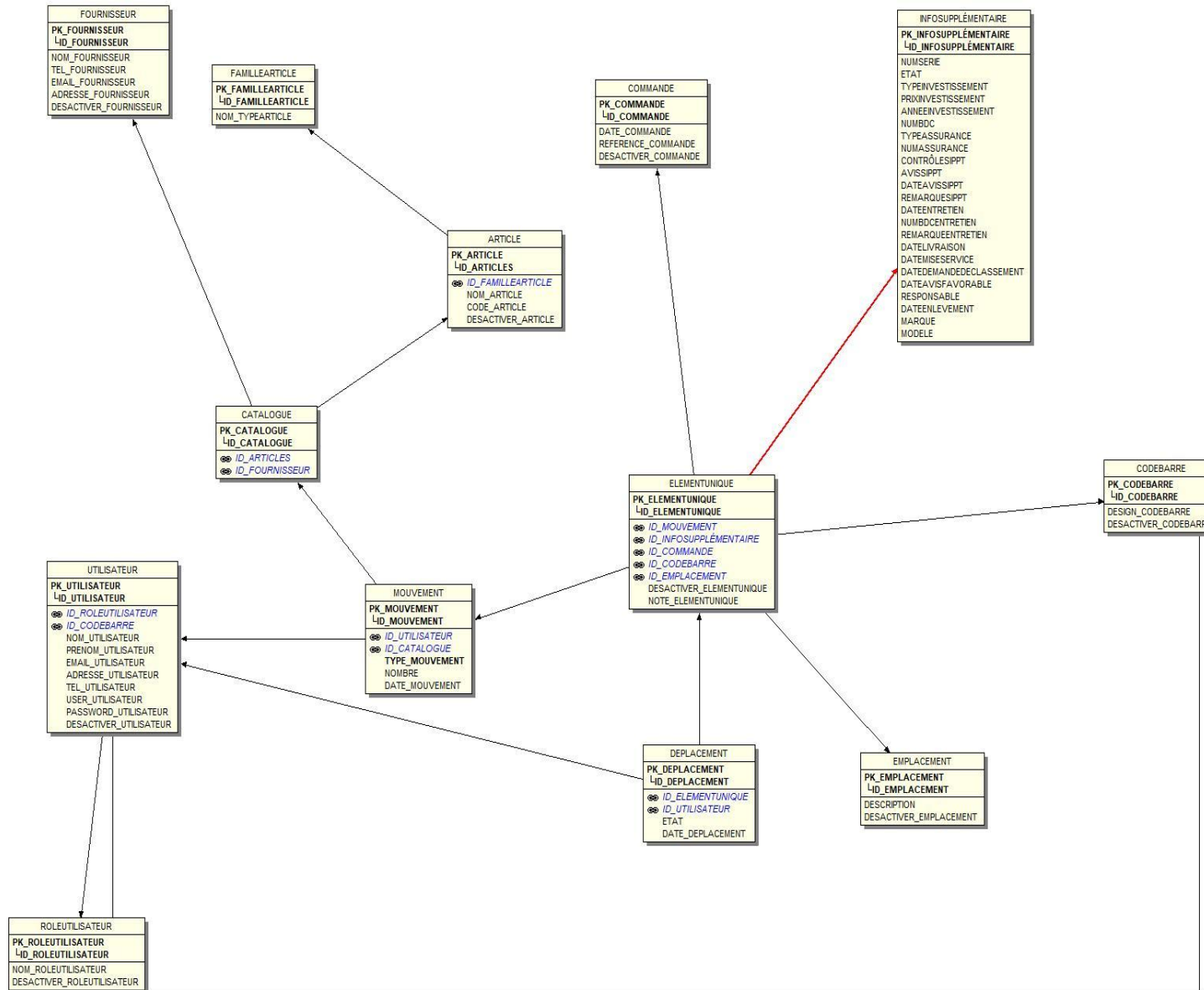
2.2 Schéma



3 MCD – GestionStock



4 MLD – GestionStock



5 Thésaurus

Entité « Article » :

Cette entité représente les types d'articles faisant partie de l'inventaire.

Exemple : Pc Dell, Télévision Samsung, etc.

- **ID_ARTICLES** : permet d'identifier l'élément de façon unique (clé primaire).
- **ID_FAMILLEARTICLE** : clé secondaire vers la famille d'article correspondante.
- **NOM_ARTICLE** : correspond au nom de l'article.
- **CODE_ARTICLE** : correspond au code unique associé à l'article.
- **DESACTIVER_ARTICLE** : correspond à l'état de l'article (supprimer ou non).

Entité « Catalogue » :

Cette entité représente les catalogues d'articles des différents fournisseurs.

- **ID_CATALOGUE** : permet d'identifier l'élément de façon unique (clé primaire).
- **ID_ARTICLES** : clé secondaire vers l'article correspondant.
- **ID_FOURNISSEUR** : clé secondaire vers le fournisseur correspondant.

Entité « Codebarre » :

Cette entité représente les codes-barres associés aux articles et aux utilisateurs.

- **ID_CODEBARRE** : permet d'identifier l'élément de façon unique (clé primaire).
- **DESIGN_CODEBARRE** : correspond aux données du code-barres.
- **DESACTIVER_CODEBARRE** : correspond à l'état du code-barres (supprimer ou non).

Entité « Commande » :

Cette entité représente les commandes encodées dans le programme.

- **ID_COMMANDE** : permet d'identifier l'élément de façon unique (clé primaire).
- **DATE_COMMANDE** : correspond à la date à laquelle la commande a été encodée.
- **REFERENCE_COMMANDE** : correspond à la référence de la commande.

Entité « Déplacement » :

Cette entité représente les déplacements des articles en interne.

- **ID_DEPLACEMENT** : permet d'identifier l'élément de façon unique (clé primaire).
- **ID_ELEMENTUNIQUE** : clé secondaire vers l'article unique correspondant.
- **ID_UTILISATEUR** : clé secondaire vers l'utilisateur correspondant.
- **ETAT** : correspond à l'état de l'objet (départ ou retour).
- **DATE_DEPLACEMENT** : correspond à la date à laquelle le déplacement a eu lieu.

Entité « Elementunique » :

Cette entité représente les articles uniques faisant partie de l'inventaire.

Exemple : s'il y a dans l'inventaire trois Pc Dell, chacun d'entre eux est unique, est possède, ses propres caractéristiques.

- **ID_ELEMENTUNIQUE** : permet d'identifier l'élément de façon unique (clé primaire).
- **ID_MOUVEMENT** : clé secondaire vers le mouvement correspondant.
- **ID_INFOSUPPLÉMENTAIRE** : clé secondaire vers les informations supplémentaires correspondante.
- **ID_COMMANDE** : clé secondaire vers la commande correspondante.
- **ID_CODEBARRE** : clé secondaire vers le code-barres correspondant.
- **ID_EMLACEMENT** : clé secondaire vers l'emplacement correspondant.
- **NOTE** : correspond aux notes associées à l'article.
- **DESACTIVER_ELEMENTUNIQUE** : correspond à l'état de l'article (supprimer ou non).

Entité « Emplacement » :

Cette entité représente les différents lieux dans lequel les articles peuvent être placés.

- **ID_EMLACEMENT** : permet d'identifier l'élément de façon unique (clé primaire).
- **DESCRIPTION** : correspond au nom de l'emplacement.

Entité « Famillearticle » :

Cette entité représente la famille à laquelle un objet appartient.

Exemple : Ordinateur, Meuble, Plante, Voiture, etc.

- **ID_FAMILLEARTICLE** : permet d'identifier l'élément de façon unique (clé primaire).
- **NOM_TYPEARTICLE**: correspond au nom de la famille.

Entité « Fournisseur » :

Cette entité représente les différents fournisseurs.

- **ID_FOURNISSEUR** : permet d'identifier l'élément de façon unique (clé primaire).
- **NOM_FOURNISSEUR** : correspond au nom du fournisseur.
- **TEL_FOURNISSEUR** : correspond au numéro de téléphone du fournisseur.
- **EMAIL_FOURNISSEUR** : correspond à l'adresse mail du fournisseur.
- **ADRESSE_FOURNISSEUR** : correspond à l'adresse du fournisseur.

Entité « Infosupplémentaire » :

Cette entité représente les informations supplémentaires liées aux articles uniques.

- **ID_INFOSUPPLÉMENTAIRE** : permet d'identifier l'élément de façon unique (clé primaire).

- **NUMSERIE** : corresponds au numéro de série de l'article.
- **ETAT** : corresponds à l'état de l'article (excellent, bon, mauvais).
- **TYPEINVESTISSEMENT** : corresponds au type d'investissement.
- **PRIXINVESTISSEMENT** : corresponds au prix de l'investissement.
- **ANNEEINVESTISSEMENT** : corresponds à l'année à laquelle l'investissement à eu lieu.
- **NUMBDC** : corresponds au numéro du bon de commande lié à l'investissement.
- **TYPEASSURANCE** : corresponds au type d'assurance de l'article.
- **NUMASSURANCE** : corresponds au numéro de l'assurance de l'article.
- **CONTRÔLESIPPT *** : corresponds au type de contrôle de la SIPPT.
- **AVISSIPPT *** : corresponds à l'avis donné par la SIPPT.
- **DATEAVISSIPPT *** : corresponds à la date à laquelle la SIPPT à donné sont avis.
- **REMARQUESIPPT *** : corresponds aux remarques faites par la SIPPT.
- **DATEENTRETIEN** : corresponds à la date du dernière entretien de l'article.
- **NUMBDCENTRETIEN** : corresponds au numéro du bon de commande lié à l'entretien de l'article.
- **REMARQUEENTRETIEN** : corresponds aux remarques concernant l'entretien de l'article.
- **DATELIVRAISON** : corresponds à la date de livraison de l'article.
- **DATEMISESERVICE** : corresponds à la date de mise en service de l'article.
- **DATEDEMANDEDECLASSEMENT** : corresponds à la date de demande de déclassement de l'article.
- **DATEAVISFAVORABLE** : corresponds à la date d'avis favorable de déclassement de l'article.
- **DATEENLEVEMENT** : corresponds à la date d'enlèvement de l'article.
- **RESPONSABLE** : corresponds à la personne responsable de l'article.
- **PRIX** : corresponds au prix de l'article.
- **MARQUE** : corresponds à la marque de l'article.
- **MODELE** : corresponds au modèle de l'article.

* **S**ervice **I**nterne pour la **P**rotection et la **P**révention au **T**ravail.

Entité « Mouvement » :

Cette entité représente les mouvements d'entrer et de sortie des articles dans le programme (lié aux commandes).

- **ID_MOUVEMENT** : permet d'identifier l'élément de façon unique (clé primaire).
- **ID_UTILISATEUR** : clé secondaire vers l'utilisateur correspondant (celui ayant généré le mouvement).
- **ID_CATALOGUE** : clé secondaire vers le catalogue correspondant.
- **TYPE_MOUVEMENT** : corresponds au type de mouvement (entrer ou sortie).
- **NOMBRE** : corresponds au nombre d'articles liés au mouvement.
- **DATE_MOUVEMENT** : corresponds à la date à laquelle le mouvement à eu lieu.

Entité « Roleutilisateur » :

Cette entité représente le rôle de l'utilisateur dans le programme, est donc, les droits auxquels il a accès.

- **ID_ROLEUTILISATEUR** : permet d'identifier l'élément de façon unique (clé primaire).
- **NOM_ROLEUTILISATEUR** : correspond au nom du rôle.
- **DESACTIVER_ROLEUTILISATEUR** : correspond à l'état du rôle (supprimer ou non).

Entité « Utilisateur » :

Cette entité représente les utilisateurs du programme.

- **ID_UTILISATEUR** : permet d'identifier l'élément de façon unique (clé primaire).
- **ID_ROLEUTILISATEUR** : clé secondaire vers le rôle correspondant
- **ID_CODEBARRE** : clé secondaire vers le code-barres correspondant
- **NOM_UTILISATEUR** : correspond au nom de l'utilisateur.
- **PRENOM_UTILISATEUR** : correspond au prénom de l'utilisateur.
- **EMAIL_UTILISATEUR** : correspond à l'adresse mail de l'utilisateur.
- **ADRESSE_UTILISATEUR** : correspond à l'adresse de l'utilisateur.
- **TEL_UTILISATEUR** : correspond au numéro de téléphone de l'utilisateur.
- **USER_UTILISATEUR** : correspond au pseudo de l'utilisateur.
- **PASSWORD_UTILISATEUR** : correspond au mot de passe de l'utilisateur.
- **DESACTIVER_UTILISATEUR** : correspond à l'état de l'utilisateur (supprimer ou non).

6 Installation et mise en place des outils

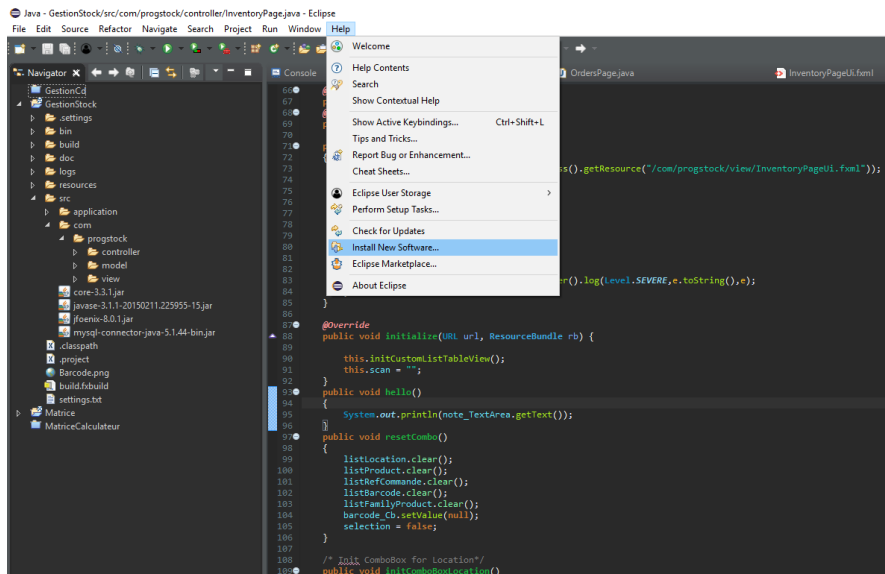
Pour commencer, voici la liste des liens permettant de télécharger tout les outils nécessaire :

- Le dernier JDK : <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Eclipse : <http://www.eclipse.org/downloads/>
- SceneBuilder : <https://gluonhq.com/products/scene-builder/>
- JFoenix : <http://www.jfoenix.com/>
- MySQLconnectorJ : <https://dev.mysql.com/downloads/connector/j/>
- ZXingCore (code-barres) : <https://mvnrepository.com/artifact/com.google.zxing/core>
- ZXing (code-barres) : <https://mvnrepository.com/artifact/com.google.zxing/javase>

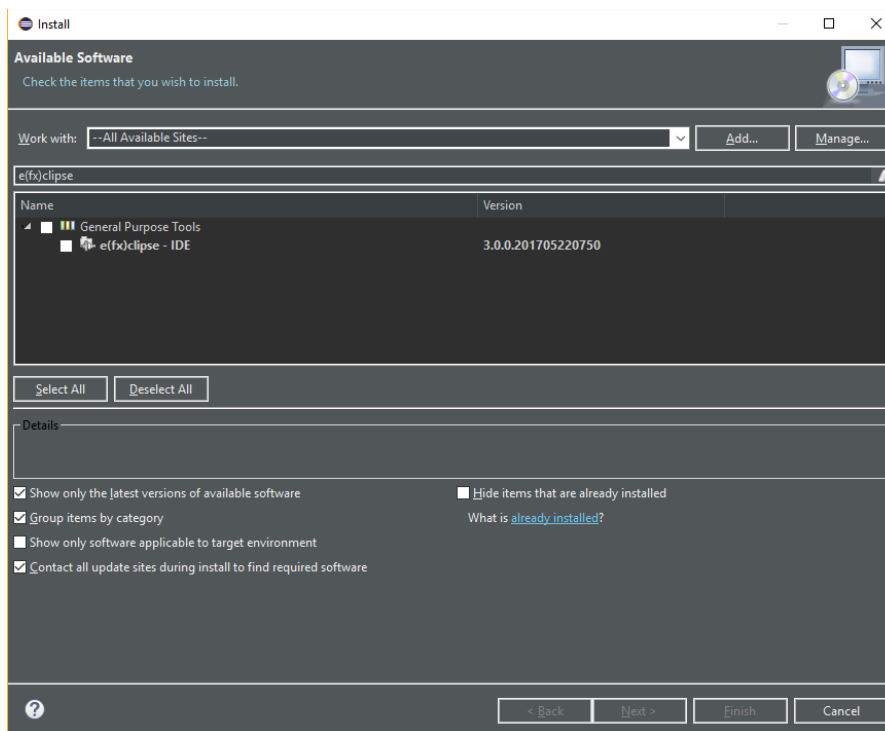
Pour ce qui est du JDK, Eclipse et SceneBuilder, il n'y a rien de particulier à faire. Téléchargé simplement l'exécutable et suivez les instructions jusqu'à ce que l'installation soit terminée.

6.1 Installation JavaFX

Dans Eclipse aller dans l'onglet "help" et ensuite Install New Software...

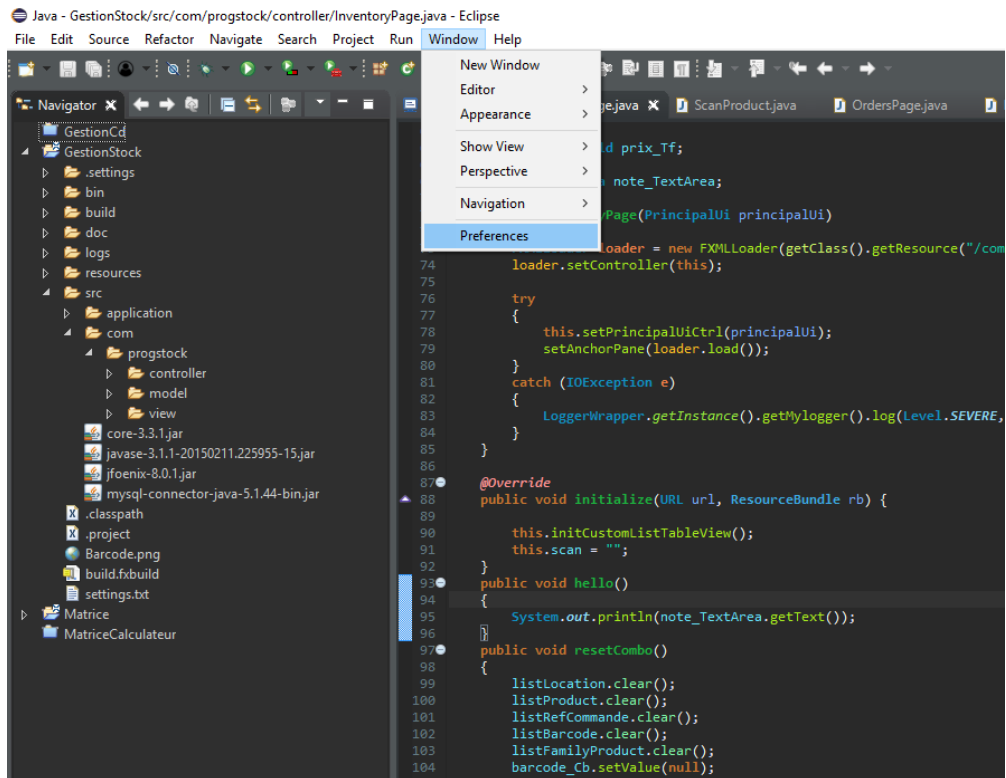


Dans la recherche de software écrire "e(fx)clipse" et suivre l'installation.

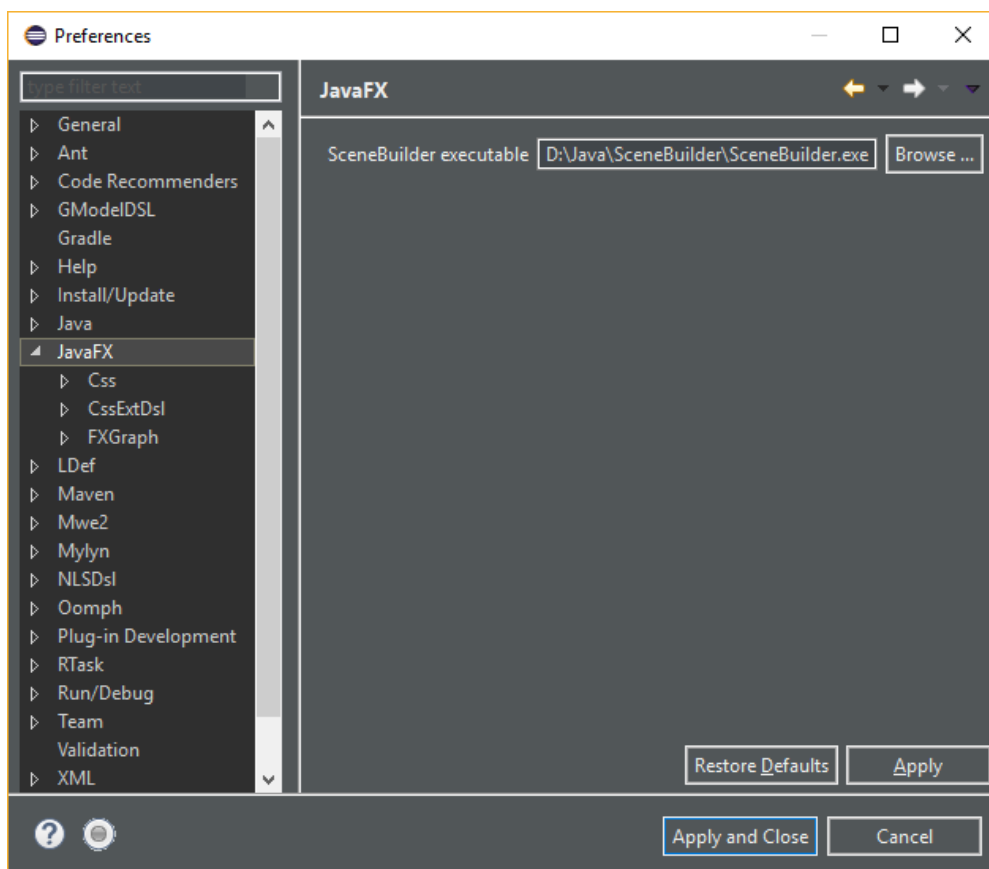


6.2 Lié le SceneBuilder avec Eclipse

Dans Eclipse aller dans l'onglet Window et ensuite Préférence.

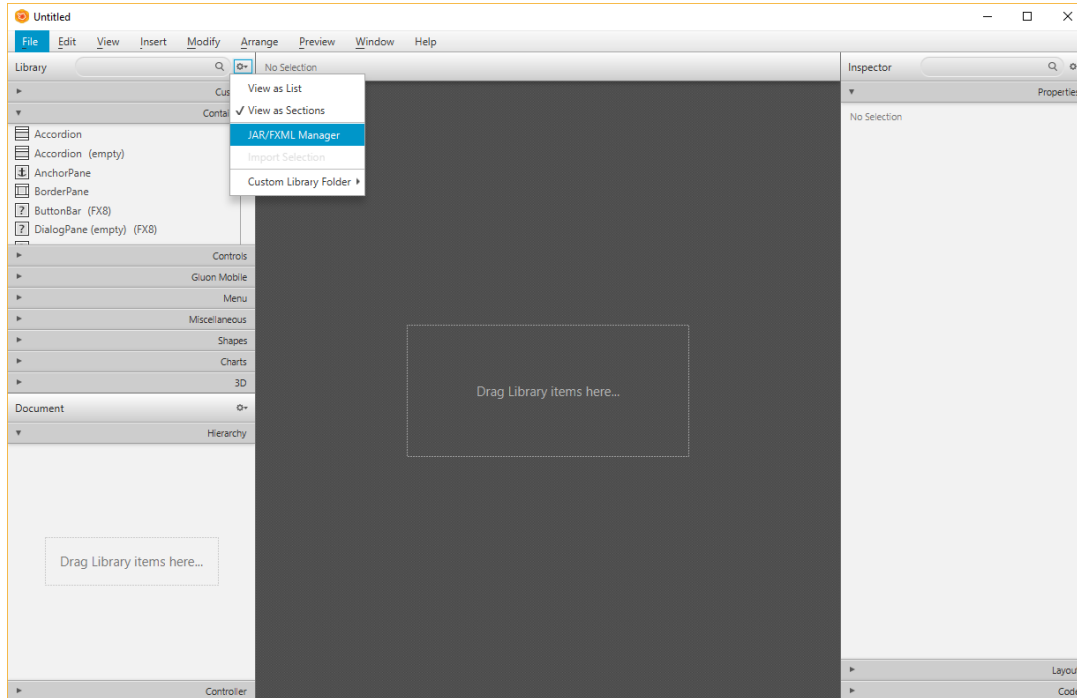


Liez le SceneBuilder avec Eclipse en lui donnant le chemin vers l'exécutable.

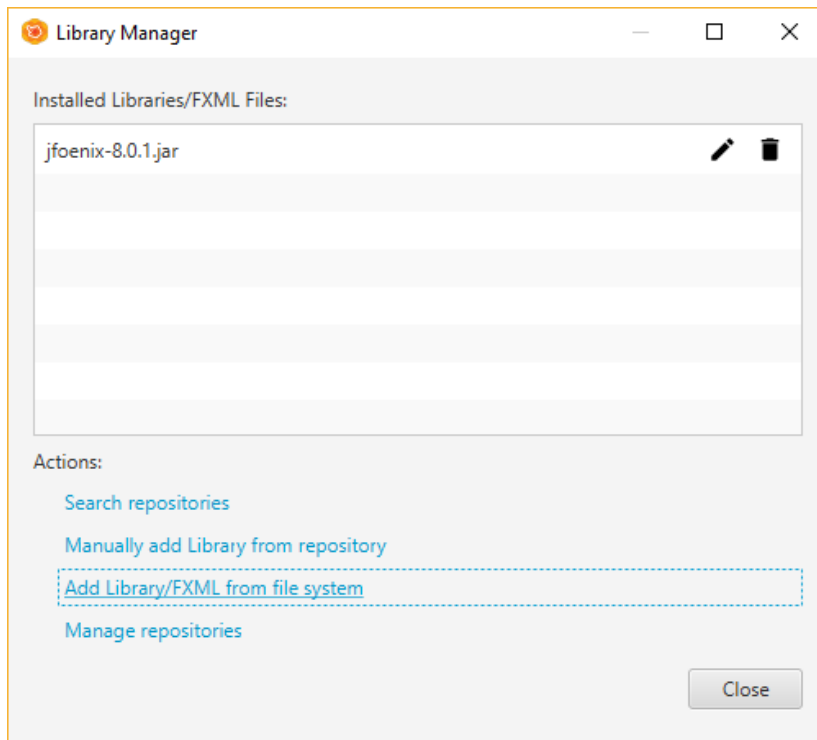


6.3 Installation JFoenix dans SceneBuilder

Dans SceneBuilder aller sur la roue de paramètre sur le côté de la recherche Library, et choisir JAR/FXML Manager.



Une fois dedans, cliquez sur "Add Library/FXML from file system" et sélectionner le fichier jar "jfoenix" télécharger précédemment.

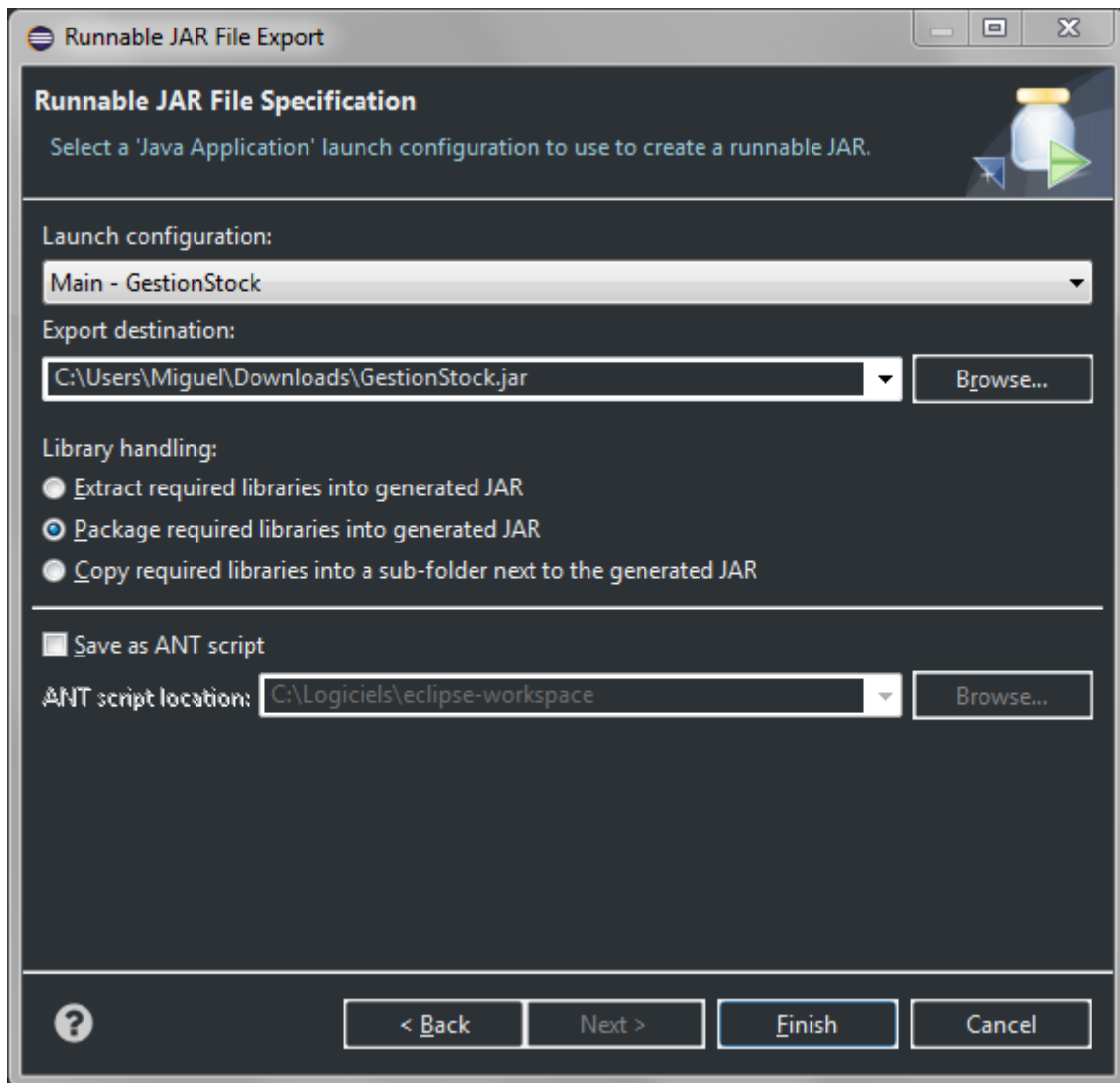


7 Création d'un JAR exécutable et d'un installateur

Avant toute chose, Il faut savoir qu'Eclipse ne génère pas tout seul un fichier exécutable. Il faut donc le générer nous-mêmes.

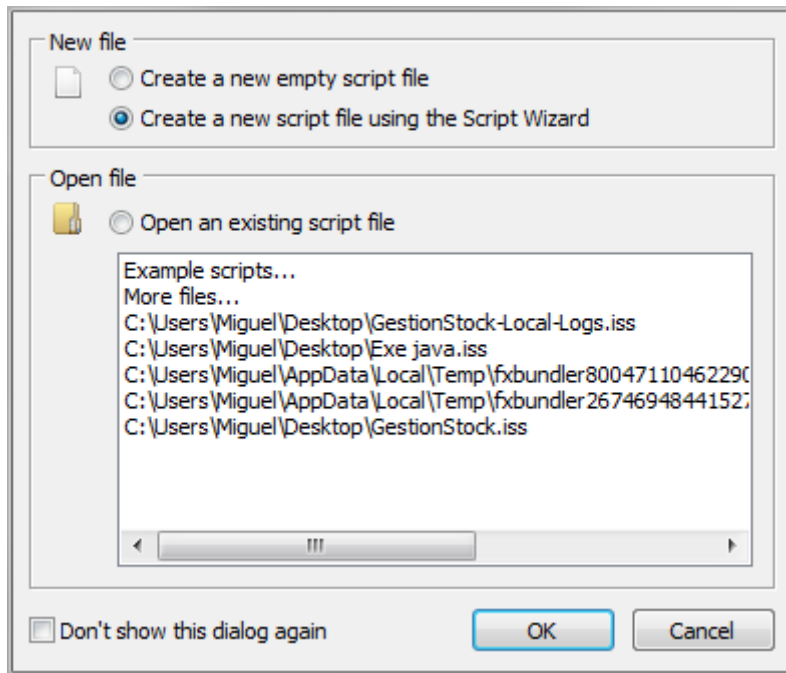
Pour le faire : Dans Eclipse, allez dans le menu File-->Export-->Java-->Runnable Jar File. Un assistant s'ouvre : renseignez, le répertoire dans lequel vous voulez déposer l'exécutable de votre programme.

Une fois, le fichier JAR crée, on peut passer à la suite.



Nous allons maintenant avoir besoin d'un programme qui se nomme "Inno Setup". C'est grâce à lui que nous pourrions créer un installateur qui nous permettra de déployer notre programme sur d'autres machines.

Étape 1 : lancez Inno setup puis choisissez (Create a new script file using script wizard). Un assistant vous aidera à construire votre installateur. Puis, cliquez sur le bouton OK.



L'assistant s'ouvre : cliquez sur next.

Étape 2 : renseignez le nom de l'application (Application name), sa version (Application version), la compagnie ou personne propriétaire (Application publisher) et enfin un site web de l'application (Application website). Puis, cliquez sur next.

Application Information

Please specify some basic information about your application.

Application name:

GestionStock

Application version:

1.0

Application publisher:

Application website:

bold = required

< Back Next > Cancel

Étape 3 : choisissez le répertoire qui accueillera l'application (Application destination base folder), lorsque vous l'installerez sur un ordinateur (Program Files, par défaut). Puis, cliquez sur next.

Application Folder
Please specify folder information about your application.

Application destination base folder:
 Program Files folder

Application folder name:
 GestionStock

☒ Allow user to change the application folder

Other:
☐ The application doesn't need a folder

bold = required

< Back Next > Cancel

Étape 4 : indiquez où se trouve le fichier exécutable de l'application (le JAR généré avec Eclipse).
 Puis, cliquez sur next.

Application Files
Please specify the files that are part of your application.

Application main executable file:
 C:\Users\Miguel\Downloads\GestionStock.jar Browse...

☒ Allow user to start the application after Setup has finished

☐ The application doesn't have a main executable file

Other application files:

Add file(s)...
 Add folder...
 Edit...
 Remove

bold = required

< Back Next > Cancel

Étape 5 : renseignez un certain nombre de Paramètres (à votre guise). Puis, cliquez sur next.

Application Shortcuts
Please specify which shortcuts should be created for your application.

☒ Create a shortcut to the main executable in the common Start Menu Programs folder
Application Start Menu folder name:
GestionStock

☒ Allow user to change the Start Menu folder name
☐ Allow user to disable Start Menu folder creation
☐ Create an Internet shortcut in the Start Menu folder
☐ Create an Uninstall shortcut in the Start Menu folder

Other shortcuts to the main executable:
☒ Allow user to create a desktop shortcut
☐ Allow user to create a Quick Launch shortcut on older versions of Windows

bold = required

< Back Next > Cancel

Étape 6 (Optionnel) : renseignez la licence du logiciel (Licence file) de l'application et d'autres informations telles que le Read me (Information file shown before the installation) et le manuel d'utilisation (Information file shown after the installation), Puis cliquez sur next.

Application Documentation
Please specify which documentation files should be shown by Setup during installation.

License file:
 Browse...

Information file shown before installation:
 Browse...

Information file shown after installation:
 Browse...

< Back Next > Cancel

Étape 7 : choisissez les différentes langues que propose Inno Setup à l'assistant d'installation pour guider l'utilisateur à déployer l'application sur son ordinateur. Puis, cliquez sur next.

Setup Languages
Please specify which Setup languages should be included.

Languages:

<input checked="" type="checkbox"/>	English
<input type="checkbox"/>	Brazilian Portuguese
<input type="checkbox"/>	Catalan
<input type="checkbox"/>	Corsican
<input type="checkbox"/>	Czech
<input type="checkbox"/>	Danish
<input type="checkbox"/>	Dutch
<input type="checkbox"/>	Finnish
<input checked="" type="checkbox"/>	French
<input type="checkbox"/>	German
<input type="checkbox"/>	Greek
<input type="checkbox"/>	Hebrew
<input type="checkbox"/>	Hungarian

bold = required

Select all
Deselect all

< Back Next > Cancel

Étape 8 : renseignez des informations telles que : le répertoire dans lequel sera déposé l'installateur de l'application (Custom compiler output folder), le nom que vous donnez à cet installateur (Computer output base file name), si vous voulez donner une image icône à votre installateur, renseignez le champ Custom file icon file. L'installateur doit-il installer votre application avec un mot de passe ? (Setup Password). Puis, cliquez sur next.

Compiler Settings
Please specify some basic compiler settings.

Custom compiler output folder:

Browse...

Compiler output base file name:

setup

Custom Setup icon file:

Browse...

Setup password:

< Back Next > Cancel

Cliquez sur next à la prochaine fenêtre, puis finish. Inno Setup vous demande si vous voulez compiler le script maintenant, répondez non. Ajouter simplement cette ligne dans le script, elle servira à créer le fichier qui permettra de stocker certaine donnée de notre programme, dont les logs.

```
[Dirs]
Name: "{%userprofile}\.GestionStock\logs"
```

Allez maintenant dans le menu d’Inno Setup : cliquez sur Build-->Compile ou simplement (CTRL + F9). L'installateur de l'application est automatiquement créé dans un dossier du répertoire que vous avez choisi.

8 Conclusion

Après avoir terminé le programme et rédigé la documentation utilisateur et technique, nous pouvons conclure les éléments suivants :

La version actuelle du logiciel répond à la demande initiale :

- Nous pouvons aisément introduire des articles dans le programme via à un système de commande, ainsi que géré tout les donnée relative à ceux-ci.
- Il est également possible de générer des codes-barres, que se soit pour les articles ou les utilisateurs, avec bien sûr la possibilité de les lires.
- Finalement, il est également possible d’avoir une trace des déplacements des articles en interne.

Travailler sur ce projet nous a également appris plusieurs choses :

- il est primordial de communiquer régulièrement avec les membres de l’équipe. En effet, même si les tâches sont bien réparties, échanger avec ses partenaires permet de débloquent certaines situations problématiques plus rapidement
- tenir un échéancier à jour en indiquant clairement les tâches restantes à réaliser ainsi que les personnes responsables permet de visualiser objectivement l’état d’avancement du projet, de savoir à tout moment si celui-ci a pris du retard et d’agir en conséquence avant qu’il ne soit trop tard
- pour mener à bien un projet, il est nécessaire de faire preuve de flexibilité et de s’adapter aux imprévus rencontrés.

C'était la première fois que nous devions travailler en groupe sur un projet conséquent et avec un but bien défini. Grâce à cette expérience, nous avons eue l'occasion d'apprendre énormément de chose sur le développement, sur le travail en équipe et sur bien d'autre point.

Cela nous à également permis de consolidé nos connaissances, mais également d'apprendre à crée des applications plus attrayantes et orientées pour le monde du travail.

Au niveau de la gestion du projet en équipe, nous avons réussi à bien nous répartir les tâches afin de réaliser nos objectifs et l'ambiance générale du groupe était très bonne.

Bref, une bonne expérience à renouveler !