

Saliha BEDDEK
Yann DELPERIE
Thibault JOLIVET

DOSSIER DE CONCEPTION

PROJET FIL ROUGE

NESTI

Application en Java – Gestion des stocks

Sommaire

1/ Projet	3
a) Sommaire.....	3
b) Utilisateurs.....	3
c) Référentiel des contraintes	3
d) Type d'application	3
2/ Planning.....	4
3/ Analyse et Conception	5
a) Cas d'utilisation.....	5
b) Diagramme de classe.....	7
c) Dictionnaire des données	8
d) Matrice de Dépendances Fonctionnelles	9
e) Modèle Conceptuel de Données et Modèle Logique de Données.....	10
f) Convention de nommage	11

1/ Projet

a) Sommaire

L'entreprise Nesti souhaite gérer ses stocks via une application.

Le but de cette application est de gérer les articles de type ustensile de cuisine, ainsi que les ingrédients nécessaires aux recettes.

b) Utilisateurs

2 utilisateurs différents auront accès à cette application :

- Les administrateurs,
- Un super administrateur qui, en plus des fonctions des administrateurs, pourra également créer des administrateurs.

c) Référentiel des contraintes

L'application sera accessible après une étape de connexion par un administrateur.

L'application doit permettre de créer, supprimer et mettre à jour les informations des ingrédients.

A partir d'un ingrédient, l'administrateur peut choisir de créer un article vendable.

L'administrateur peut aussi ajouter des articles de types ustensiles de cuisine.

Les frais de livraison varient en fonction du poids de l'article.

Les articles seront achetés à différents fournisseurs. Il se peut qu'un nouveau fournisseur puisse nous vendre des articles.

Les prix varient aussi en fonction du temps.

Prévoir un affichage qui permet de saisir une entrée dans le stock d'un article.

Le prix d'achat peut varier d'un fournisseur à l'autre.

NB : pour la suite du dossier, on nommera produit un ustensile ou un ingrédient.

d) Type d'application

L'application sera développée en JAVA, accompagnée d'une base de données locale qui sera accessible uniquement par les administrateurs.

Le code sera effectué via l'application ECLIPSE, du fait de son environnement de développement gratuit, sa facilité d'utilisation, ainsi que sa puissance de développement.

La base de données locale sera supportée par MySQL, SGBD solide et utilisé par de grands groupes spécialisés dans l'Internet.

2/ Planning

Taches/Délais	S1	S2	S3	...	S21
Analyse et Conception					
Analyse des besoins du client					
Capture des besoins fonctionnels					
Capture des besoins techniques					
Conception					
Définitions des cas utilisateurs					
Définition des classes					
Définition du Modèle conceptuel de données					
Réalisation de la Maquette					
Réalisation du StoryBoard					
Realisation					
Conception/ Programmation					
Guide d'utilisation					
Livraison					★

Planning détaillé des actions à mener

3/ Analyse et Conception

a) Cas d'utilisation

2 cas principaux d'utilisations sont présents (Admin et Super-Admin).

Super-Admin :



Diagramme de cas généralisé pour un Super Administrateur

Admin :

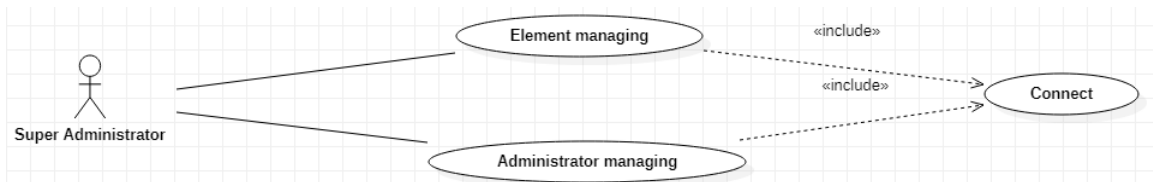


Diagramme de cas généralisé pour un Administrateur

La différence entre Super administrateur et administrateur « simple » réside seulement dans la possibilité de la gestion des administrateurs. Pour la suite des « cas utilisateurs » on étudiera seulement les administrateurs.

Éléments :

Ici, les éléments sont organisés en 4 parties :

- Produit,
- Article,
- Commande,
- Fournisseur.

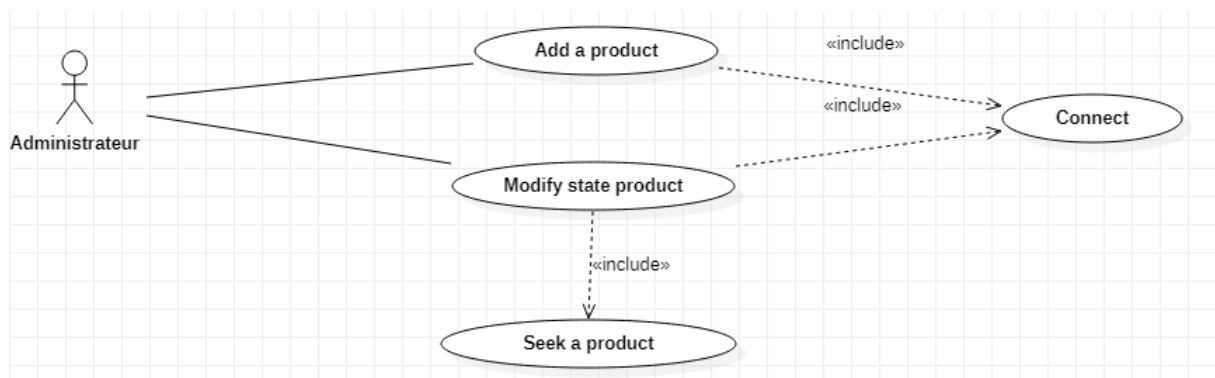


Diagramme de cas utilisateur pour la gestion d'un produit

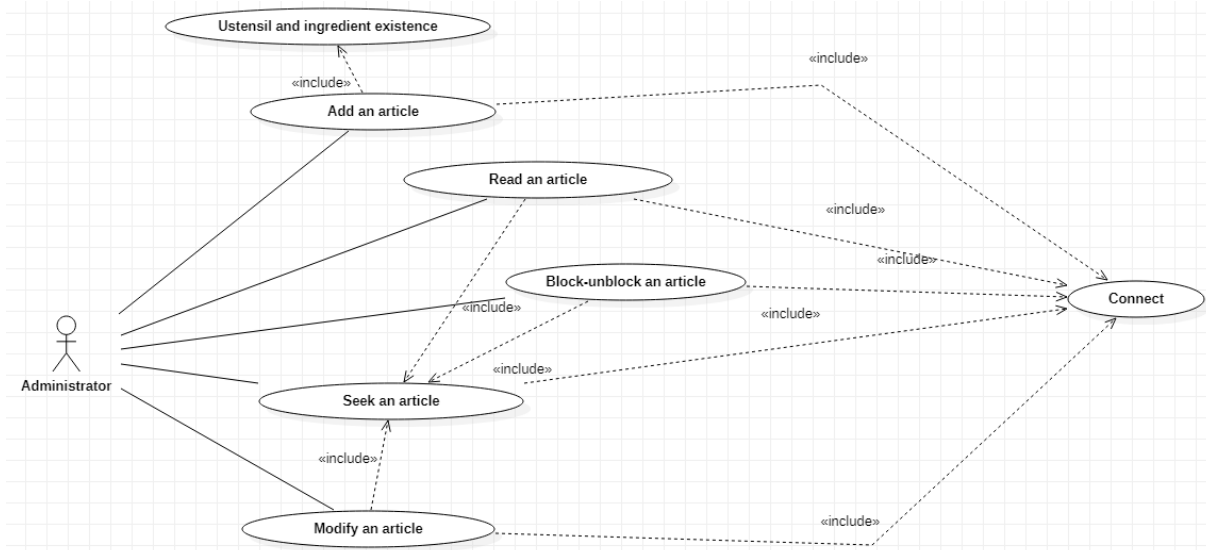


Diagramme de cas utilisateur pour la gestion d'un article

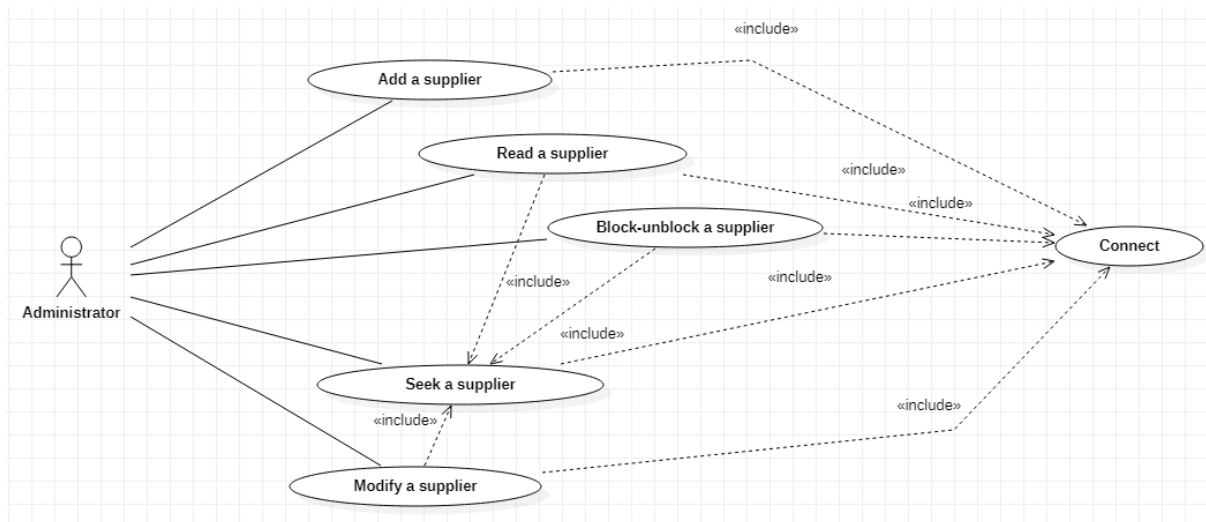


Diagramme de cas utilisateur pour la gestion d'un fournisseur

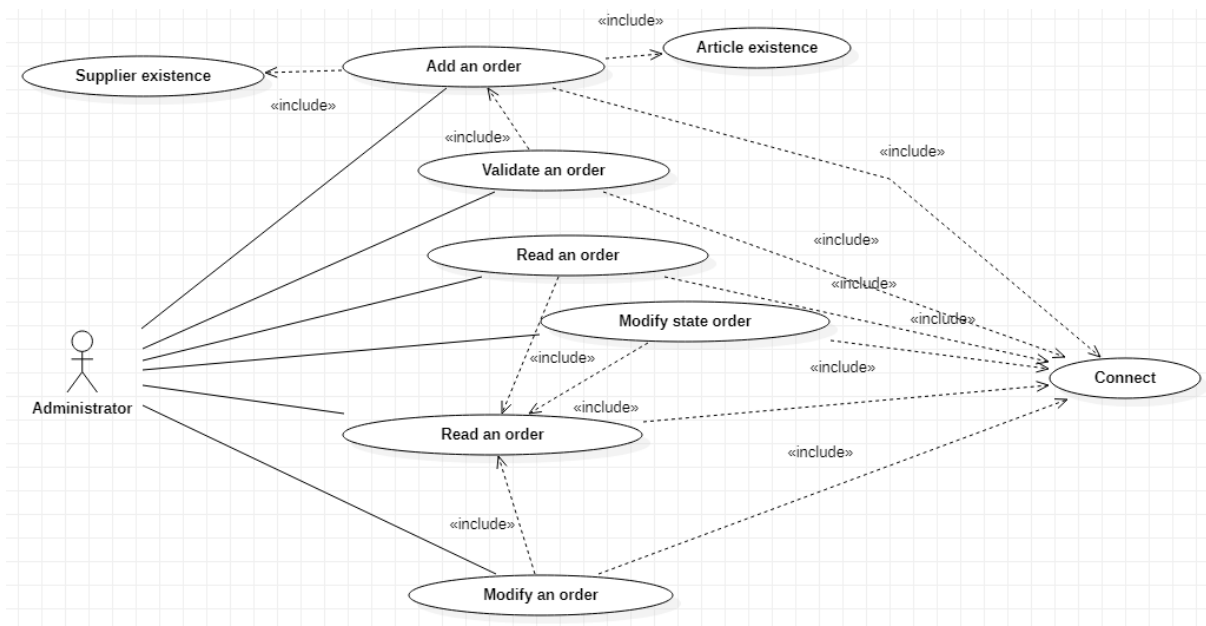


Diagramme de cas utilisateur pour la gestion d'une commande

b) Diagramme de classe

Les schémas ci-dessous représentent les différentes classes. Ils décrivent leur contenu ainsi que les éventuelles relations pouvant exister entre elles :

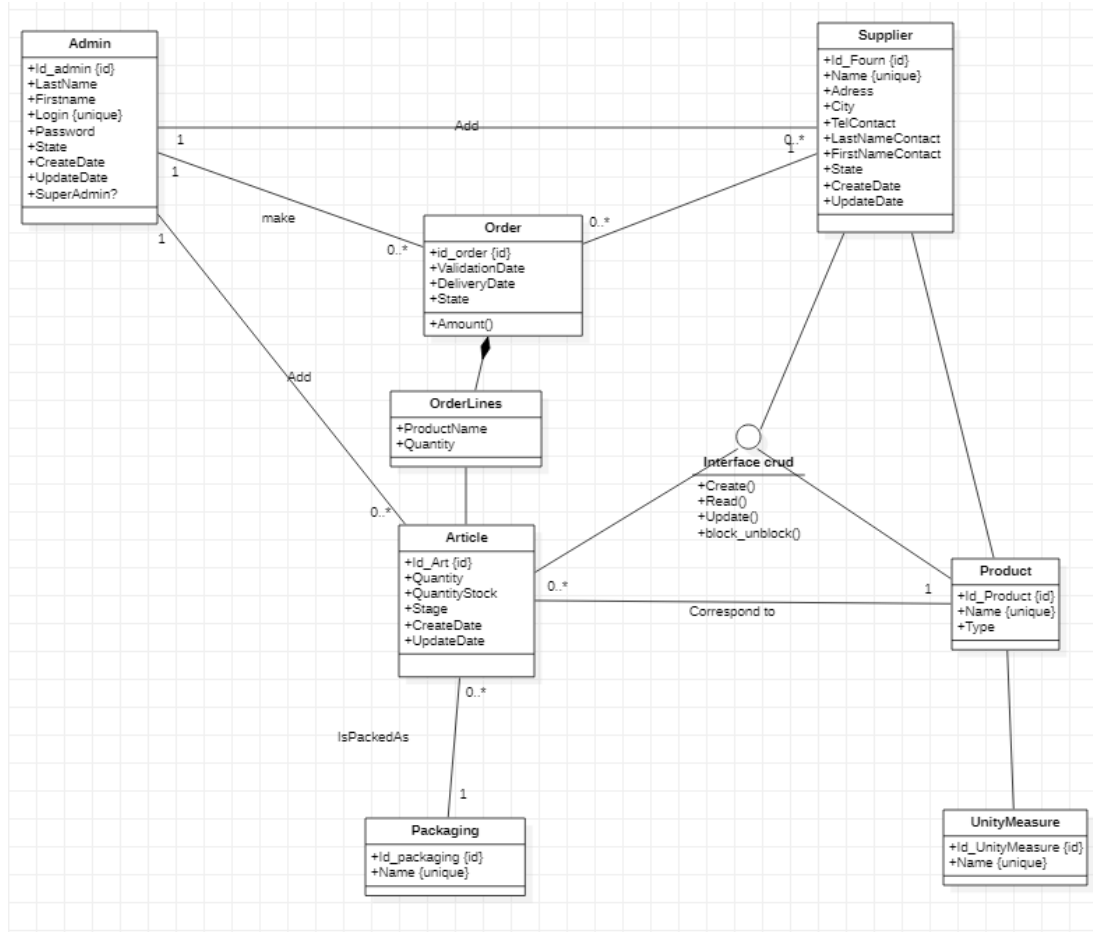


Diagramme de classe Métier

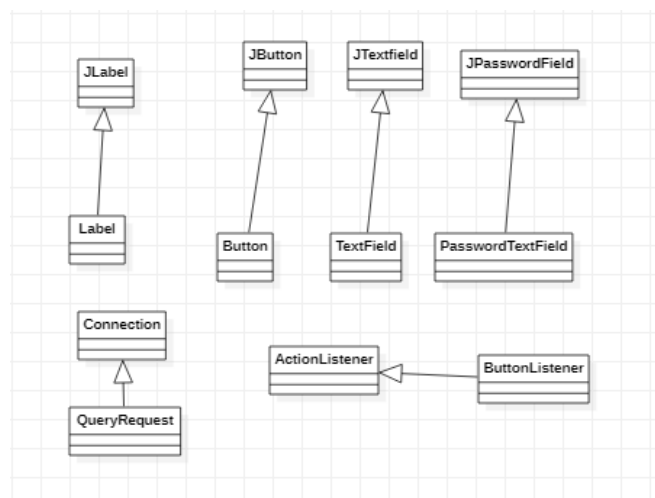


Diagramme de classe conception

c) Dictionnaire des données

Il représente le catalogue de données de référence nécessaire à la conception de la base de données relationnelle.

Donnée	Type	Longueur	Code BDD	Exemple		Donnée	Type	Longueur	Code BDD	Exemple
Admin Identifiant	COUNTER		Id_Admin	12						
Admin Last Name	VARCHAR	50	LastName	Smith						
Admin FirstName	VARCHAR	50	FirstName	Joe						
Admin Login	VARCHAR	50	Login	JSmith1234						
Admin Password	VARCHAR	50	Password	*****						
Admin State	VARCHAR	50	State	Unblocked						
Admin Creation Date	DATETIME		CreatingDate	02/11/2020						
Admin Update Date	DATETIME		UpdateDate							
Is Admin SuperAdmin	BOOLEAN		IsSuperAdmin	Yes/No						
Supplier Identifiant	COUNTER		Id_Supplier	20	Packaging Identifiant	COUNTER			Id_Packaging	
Supplier Name	VARCHAR	50	Name	Fruity	Packaging Name	VARCHAR	50		Name	
Supplier Address	VARCHAR	50	Address	123 John St						
City	VARCHAR	50	City	London	Product Identifiant	COUNTER			Id_Product	
Supplier Contact First name	VARCHAR	50	ContactFirstName	John	Product Name	VARCHAR	50		Name	
Supplier Contact Last name	VARCHAR	50	ContactLastName	Smith	Product Type	VARCHAR	20		Type	
Supplier Contact number	VARCHAR	50	ContactNumber	0102030405	Product State	VARCHAR	20		State	
Supplier State	VARCHAR	50	State	Unblocked	Article reference	VARCHAR	50		Id_Article	
Supplier Creating Date	DATETIME		CreatingDate	04/05/2020	Article Quantity	DECIMAL	7,2		Quantity	
Supplier Update Date	DATETIME		UpdateDate		Article Stock Quantity	INT	10		QuantityRealStock	
Order reference	VARCHAR	50	Id_Order	N°987654321	Article State	VARCHAR	50		State	
Order Validation Date	DATETIME		ValidationDate		Article 's Creation Date	DATETIME			CreatingDate	
Order Delivery Date	DATETIME		DeliveryDate		Article 's Update Date	DATETIME			UpdateDate	
Order State	VARCHAR	50	State	Valide	Buying Price	DECIMAL	7,2		Buying_price	
Order Line Quantity	INT	15	Quantity /		Price Update Date	DATE			PriceUpdateDate	
Unit Identifiant	COUNTER		Id_UnitMeasure							
Unit Name	VARCHAR	20	Name	Kg / L						

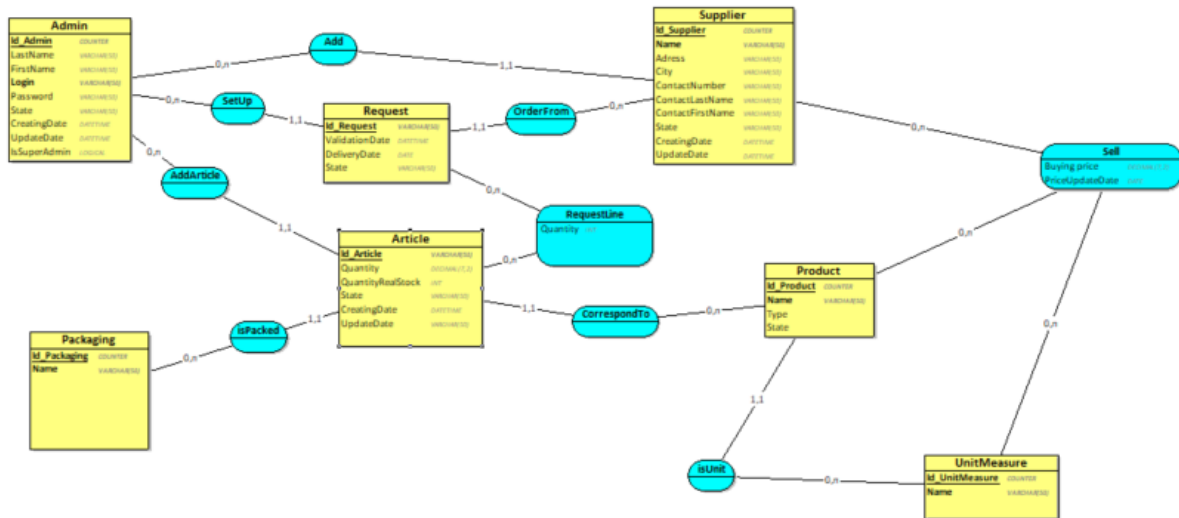
Dictionnaire des données

d) Matrice de dépendances fonctionnelles

	Admin	Supplier	Article	Product	Packaging	Order	Unit
	Id_Admin	Id_Supplier	Id_Article	Id_Product	Id_Packaging	Id_Order	Id_unit
Id_Admin	*						
Admin Last Name	1						
Admin FirstName	1						
Admin Login	1						
Admin Password	1						
Admin State	1						
Admin Creation Date	1						
Admin Update Date	1						
Is Admin SuperAdmin	1						
Id_Supplier		*					
Supplier Name		1					
Supplier Address		1					
Supplier Contact First name		1					
Supplier Contact Last name		1					
Supplier Contact number		1					
Supplier State		1					
Supplier Creating Date		1					
Supplier Update Date		1					
Order reference (Id_Order)	1	1				*	
Order Validation Date						1	
Order Delivery Date						1	
Order State						1	
Order Line QQuantity						1	
Unit Identifiant							
Unit Name							1
Packaging Identifiant					*		
Packaging Name					1		
Product Identifiant				*			
Product Name				1			
Product Type				1			1
Product State				1			
Id_Article			*				
Article Quantity			1				
Article Stock Quantity			1				
Article State			1				
Article 's Creation Date			1				
Article 's Update Date			1				
Buying Price		1	1				
Price Update Date		1	1				

Matrice fonctionnelle

e) Modèle Conceptuel de Données et Modèle Logique de Données



MCD

MLD :

Il consiste à décrire la structure de données utilisée.

Il s'agit d'une dérivation du MCD dans un formalisme adapté à une implémentation sous forme de base de données relationnelle.

Les clés primaires sont en gras et soulignées. Les clés étrangères sont en bleu.

Admin = (**Id Admin** COUNTER, LastName VARCHAR(50), FirstName VARCHAR(50), **Login** VARCHAR(50), Password VARCHAR(50), State VARCHAR(50), CreatingDate DATETIME, UpdateDate DATETIME, IsSuperAdmin LOGICAL);

Supplier = (**Id Supplier** COUNTER, Name VARCHAR(50), Address VARCHAR(50), City VARCHAR(50), ContactNumber VARCHAR(50), ContactLastName VARCHAR(50), ContactFirstName VARCHAR(50), State VARCHAR(50), CreatingDate DATETIME, UpdateDate DATETIME, #Id_Admin);

Request = (**Id Request** VARCHAR(50), ValidationDate DATETIME, DeliveryDate DATE, State VARCHAR(50), #Id_Supplier, #Id_Admin);

UnitMeasure = (**Id UnitMeasure** COUNTER, Name VARCHAR(50));

Packaging = (**Id Packaging** COUNTER, Name VARCHAR(50));

Product = (**Id Product** COUNTER, Name VARCHAR(50), Type , State , #Id_UnitMeasure);

Article = (**Id Article** VARCHAR(50), Quantity DECIMAL(7,2), QuantityRealStock INT, State VARCHAR(50), CreatingDate DATETIME, UpdateDate VARCHAR(50), #Id_Packaging, #Id_Admin, #Id_Product);

RequestLine = (#Id_Article, #Id_Request, Quantity INT);

Sell = (#Id_Supplier, #Id_Product, #Id_UnitMeasure, Buying_price DECIMAL(7,2), PriceUpdateDate DATE);

f) Convention de nommage

Pour la partie code en JAVA, les conventions de nommage seront :

- Package : Le nom du package doit être en minuscule (ex : project, java ...).
- Classes : Les classes commencent par une lettre majuscule et doivent être de type « Nom » (ex : Button, Listener, Object ...).
- Méthodes : Les méthodes doivent commencer par une lettre minuscule et être de type « Verbe » (ex : create(), check() ...).
- Variables : Les variables doivent commencer par une lettre minuscule (ex : name, number ...).
- CamelCase : Si le nom d'une classe/méthode/variable est composé de deux mots ou plus, chaque mot suivant doit commencer par une lettre majuscule (ex : firstName, checkPassword() ...).
- Global : Code et commentaires à écrire en anglais.

Pour la partie Base de données SQL, les conventions de nommages seront :

- Ne pas utiliser les mots réservés (date, delete, add ...).
- Ne pas utiliser de caractères spéciaux.
- Eviter les majuscules, privilégier les underscores pour l'utilisation de deux mots (ex : « date_inscription » plutôt que « DateInscription »).
- Eviter l'utilisation d'abréviation.
- Noms des tables : Utiliser un nom représentatif du contenu, utiliser un seul mot si possible, préfixer les noms des tables (ex : user_data, company_data ...).
- Noms des colonnes : Préfixer toutes les colonnes de la même façon pour chaque table (plus pratique pour les jointures).
- Global : Ecrire en anglais.