

Université Abderrahmane Mira – Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Mini Projet de Génie Logiciel

Thème :
Application Desktop de Gestion de Livraison

Niveau : 3ème année Licence – Informatique
Année universitaire : 2025 / 2026

Réalisé par :
Makhlouf Yasmine

Encadré par :
Dr N. Yessad

Groupe : b1

decembre2025

1. Introduction

Le présent mini projet s'inscrit dans le cadre du module de Génie Logiciel.

L'objectif principal est de concevoir et développer une **application desktop Java** utilisant l'environnement NetBeans, en respectant les notions fondamentales du génie logiciel, notamment :

- l'analyse des besoins,
- la modélisation UML,
- l'architecture en trois tiers,
- la conception d'une base de données relationnelle,
- la mise en œuvre d'interfaces graphiques avec Java Swing,
- ainsi que la génération d'un exécutable fonctionnel.

Cette application interne de gestion des livraisons est destinée à être utilisée uniquement par le réceptionniste du service.

Ce rapport présente les différentes étapes de conception, depuis l'analyse des besoins jusqu'à la base de données et la modélisation du système.

2. Analyse des Besoins

2.1 Objectif général

L'objectif du projet est de développer une application permettant :

- la gestion des clients ;
- la gestion des commandes ;
- l'affectation des livreurs aux livraisons ;
- le suivi des livraisons ;
- la gestion des véhicules utilisés ;
- la gestion des partenaires commerciaux.

L'application doit permettre de centraliser l'information liée à la livraison et d'offrir une interface simple, professionnelle et intuitive.

• 2.2 Acteur du système:

- L'application est utilisée exclusivement par **le réceptionniste**, qui constitue **l'unique acteur** du système.

Rôle du Réceptionniste :

Le réceptionniste est l'unique utilisateur de l'application et occupe un rôle central dans le fonctionnement du système.

Il est responsable de l'ensemble des opérations administratives liées à la gestion des livraisons.

Ses tâches couvrent la création, le suivi et la clôture du processus de livraison, ainsi que la gestion des informations relatives aux clients, livreurs, véhicules et partenaires.

1. Gestion des Clients

Le réceptionniste peut enregistrer un nouveau client lorsque celui-ci dépose un colis ou demande une livraison.
Il peut également mettre à jour les informations du client ou supprimer un client dans le cas d'erreurs ou de données obsolètes.

2. Gestion des Commandes

Le réceptionniste est chargé de créer les commandes de livraison en saisissant toutes les informations nécessaires : poids, prix, mode de paiement, état de paiement, date de création, et informations du client.
Il peut également modifier une commande en cas d'erreur ou l'annuler à la demande d'un client.

3. Attribution des Ressources

Le réceptionniste sélectionne un livreur disponible pour effectuer la livraison et choisit le véhicule adéquat selon le type de colis ou la capacité requise.
Il vérifie également l'état de disponibilité des livreurs et des véhicules.

4. Gestion des Livraisons

Après la création d'une commande, le réceptionniste génère automatiquement une livraison.

Il peut :

démarrer la livraison,

mettre à jour son statut (en cours, livrée, retournée),

clôturer la livraison une fois terminée.

Il peut aussi consulter tous les détails d'une livraison passée ou en cours.

5. Gestion des Paiements

Le réceptionniste enregistre le mode de paiement, valide l'état du paiement (payé / non payé), et renseigne la date de paiement.

6. Gestion des Retours

En cas de livraison échouée ou refusée par le client, le réceptionniste met à jour l'état de retour de la commande et enregistre la raison du retour si nécessaire.

7. Gestion des Partenaires

Le réceptionniste peut gérer les partenaires commerciaux en :

ajoutant un partenaire,

modifiant ses informations,

le supprimant,

et en associant un partenaire à certaines commandes.

8. Consultation et Historique

Le réceptionniste peut consulter :

la liste complète des commandes,

l'historique des livraisons,

les livreurs disponibles,

les véhicules disponibles,

les partenaires,

les commandes par client ou par date.

2.3 Fonctionnalités principales

Gestion des Clients

- Ajouter un client
- Modifier un client
- Supprimer un client

Gestion des Commandes

- Création d'une commande
- Mise à jour des informations
- Gestion du paiement
- Annulation d'une commande

Gestion des Livraisons

- Génération d'une livraison à partir d'une commande
- Affectation d'un livreur
- Utilisation d'un véhicule
- Changement du statut de la livraison

Gestion des livreurs

- Consulter la liste des livreurs disponibles
- Modifier la disponibilité d'un livreur si nécessaire

Gestion des Véhicules

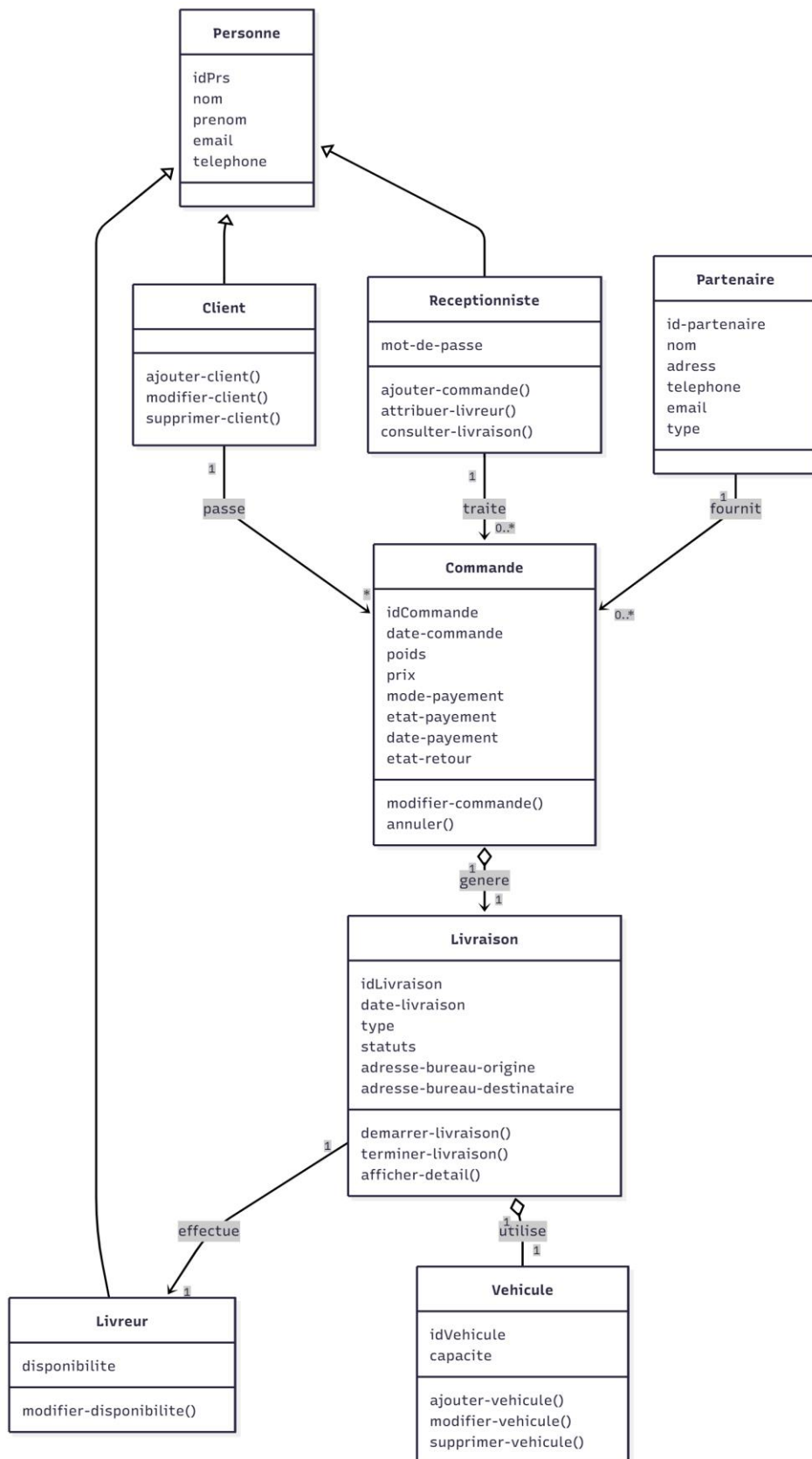
- Ajout d'un véhicule
- Modification d'un véhicule
- Suppression d'un véhicule

Gestion des Partenaires

- Enregistrement d'un partenaire
- Attribution de commandes

3. Diagramme de Classes

Voici le diagramme de classes utilisé pour la conception du projet



4. Modèle Relationnel :

- personne (id_pers, nom, prénom, email, téléphone)
- client (# id-pers)
- réceptionniste (# id-pers, mot de passe)
- Livreur (id-pers, disponibilité, # id-livraison)
- Partenaire (id-partenaire, nom, adresse, téléphone, email, type)
- Commande (id-commande, date commande, poids, prix, mode paiement, état paiement, date paiement, état retour, # id partenaire, # id réception, # id client)
- Livraison (id livraison, # id commande, date livraison, type, statut, adresse bureau origine, adresse bureau destination)
- Véhicule (id véhicule, # id livraison, capacité)

5. Script SQL de Création de la Base de Données

```
CREATE TABLE Personne ( idPrs INT PRIMARY KEY AUTO_INCREMENT,  
nom VARCHAR(100) NOT NULL, prenom VARCHAR(100) NOT NULL, email  
VARCHAR(150) UNIQUE NOT NULL, telephone VARCHAR(20),  
type_personne ENUM('Client', 'Receptionniste', 'Livreur') NOT NULL, INDEX  
idx_email (email), INDEX idx_type (type_personne) );
```

```
CREATE TABLE Client ( idPrs INT PRIMARY KEY, date_inscription DATE
DEFAULT (CURRENT_DATE), FOREIGN KEY (idPrs) REFERENCES
Personne(idPrs) ON DELETE CASCADE );
```

```
CREATE TABLE Receptionniste ( idPrs INT PRIMARY KEY, mot_de_passe
VARCHAR(255) NOT NULL, date_embauche DATE DEFAULT
(CURRENT_DATE), FOREIGN KEY (idPrs) REFERENCES Personne(idPrs) ON
DELETE CASCADE );
```

```
CREATE TABLE Livreur ( idPrs INT PRIMARY KEY, disponibilite
ENUM('disponible', 'en_livraison', 'indisponible') DEFAULT 'disponible',
date_embauche DATE DEFAULT (CURRENT_DATE), FOREIGN KEY (idPrs)
REFERENCES Personne(idPrs) ON DELETE CASCADE, INDEX
idx_disponibilite (disponibilite) );
```

```
CREATE TABLE Partenaire ( id_partenaire INT PRIMARY KEY
AUTO_INCREMENT, nom VARCHAR(150) NOT NULL, adresse
VARCHAR(255), telephone VARCHAR(20), email VARCHAR(150), type
VARCHAR(50), INDEX idx_nom (nom) );
```

```
CREATE TABLE Vehicule ( idVehicule INT PRIMARY KEY AUTO_INCREMENT,
capacite VARCHAR(50) NOT NULL, marque VARCHAR(100), modele
VARCHAR(100), immatriculation VARCHAR(50) UNIQUE, statut
ENUM('disponible', 'en_service', 'maintenance') DEFAULT 'disponible',
INDEX idx_statut (statut) );
```

```
CREATE TABLE Commande ( idCommande INT PRIMARY KEY
AUTO_INCREMENT, date_commande DATETIME DEFAULT
CURRENT_TIMESTAMP, poids DECIMAL(10,2) NOT NULL, prix
DECIMAL(10,2) NOT NULL, mode_payement ENUM('espece', 'carte',
'virement', 'cheque') NOT NULL, etat_payement ENUM('en_attente', 'paye',
'rembourse') DEFAULT 'en_attente', date_payement DATETIME, etat_retour
ENUM('aucun', 'demande', 'en_cours', 'termine') DEFAULT 'aucun', idClient
INT NOT NULL, idReceptionniste INT, id_partenaire INT, FOREIGN KEY
(idClient) REFERENCES Client(idPrs) ON DELETE RESTRICT, FOREIGN KEY
(idReceptionniste) REFERENCES Receptionniste(idPrs) ON DELETE SET
NULL, FOREIGN KEY (id_partenaire) REFERENCES
```

```
Partenaire(id_partenaire) ON DELETE SET NULL, INDEX  
idx_date_commande (date_commande), INDEX idx_etat_paiement  
(etat_paiement), INDEX idx_client (idClient) );
```

```
CREATE TABLE Livraison ( idLivraison INT PRIMARY KEY AUTO_INCREMENT,  
date_livraison DATETIME, type ENUM('standard', 'express', 'economique') NOT NULL,  
statuts ENUM('en_attente', 'en_cours', 'livree', 'annulee') DEFAULT 'en_attente',  
adresse_bureau_origine VARCHAR(255) NOT NULL, adresse_bureau_destinataire  
VARCHAR(255) NOT NULL, idCommande INT UNIQUE NOT NULL, idLivreur INT,  
idVehicule INT, FOREIGN KEY (idCommande) REFERENCES Commande(idCommande)  
ON DELETE CASCADE, FOREIGN KEY (idLivreur) REFERENCES Livreur(idPrs) ON  
DELETE SET NULL, FOREIGN KEY (idVehicule) REFERENCES Vehicule(idVehicule) ON  
DELETE SET NULL, INDEX idx_statuts (statuts), INDEX idx_date_livraison  
(date_livraison), INDEX idx_livreur (idLivreur) );
```

6. Conclusion

Ce mini projet a permis de mettre en pratique les notions fondamentales du génie logiciel, telles que :

- l'analyse des besoins,
- la modélisation UML,
- l'architecture trois tiers,
- la création de bases de données relationnelles,
- et le développement d'interfaces graphiques Java Swing.

L'application de gestion de livraison répond aux besoins fonctionnels identifiés et propose une architecture claire, propre et évolutive.

La modélisation UML et le modèle relationnel assurent une bonne

structuration de l'information, tandis que la conception en couches facilite la maintenabilité.

Ce travail constitue une base solide pour un éventuel développement futur ou pour la transformation en application web ou mobile.