

---

# CAHIER DES CHARGES

---

Projet : Application Web d’Evaluation Commerciale

Réalisé par :

- Esrae Ben Selma
- El Hassan Gouiga
- Ahmed Hammadi

Encadrant :

- MMe. Nasrine Rym Guibadg

Établissement :

- EILCO



## Table des matières

1. Introduction .....	3
1.1 Contexte du projet .....	3
1.2 Objectifs du projet .....	3
1.3 Périmètre du système .....	3
2. Acteurs et description des utilisateurs .....	4
2.1 Liste des acteurs .....	4
2.2 Rôles et responsabilités des acteurs .....	4
Administrateur .....	4
Gestionnaire .....	4
Commercial .....	4
3. Besoins fonctionnels (ce que le système fait) .....	5
3.1 Cas d'utilisation – Administrateur .....	5
3.2 Cas d'utilisation – Gestionnaire .....	5
3.3 Cas d'utilisation – Commercial .....	5
4. Besoins non fonctionnels .....	6
4.1 Sécurité .....	6
4.2 Performance .....	6
4.3 Ergonomie et expérience utilisateur (UX) .....	6
4.4 Compatibilité .....	7
4.5 Fiabilité et intégrité des données .....	7
5. Interfaces utilisateur (maquettes) .....	8
5.1 Liste des écrans prévus .....	8
6. Modélisation des données et du système .....	8
6.1 Modélisation UML — Diagramme de classes .....	8
6.2 Modèle Conceptuel de Données — MCD .....	9
6.3 Modèle Logique de Données — MLD .....	10
Table : UTILISATEUR .....	10
Table : SQUAD .....	11
Table : VENTE .....	11
Table : MESSAGE .....	11
7. Technologies prévues .....	12
Frontend (Interface utilisateur) .....	12
Backend (Logique applicative) .....	12
Base de données .....	12
Sécurité .....	12
Visualisation des données .....	12
8. Planning prévisionnel .....	13
9. Conclusion .....	13

# 1. Introduction

## 1.1 Contexte du projet

Dans le cadre du Projet de Bureau d'Études en Informatique (BE INFO), il est demandé aux étudiants de concevoir une application web en respectant un processus d'ingénierie logicielle complet : analyse du besoin, modélisation, conception de base de données, réalisation et présentation finale.

Le thème retenu pour ce projet consiste à développer une plateforme de gestion et d'évaluation des performances commerciales. Ce sujet permet d'aborder de nombreux aspects essentiels du développement web : gestion des utilisateurs, authentification, stockage de données, affichage de statistiques, permissions selon les rôles, etc.

Ainsi, ce projet constitue une mise en situation pratique, permettant d'appliquer les connaissances acquises en développement et en conception de systèmes d'information.

## 1.2 Objectifs du projet

Le projet a pour objectif de développer une plateforme web permettant :

- la **saisie et la consultation de ventes** réalisées par les commerciaux ;
- le **suivi des performances individuelles** (classements, statistiques) ;
- le **suivi d'une équipe commerciale** par un gestionnaire ;
- la **communication interne** grâce à une messagerie simple ;
- la **gestion des utilisateurs et des squads**(équipes) par l'administrateur.

Ce projet vise à fournir une solution fonctionnelle répondant aux besoins pédagogiques du BE INFO : modélisation, implémentation, sécurisation, test et évaluation.

## 1.3 Périmètre du système

Le système permettra :

- l'authentification et la gestion des rôles (Admin / Gestionnaire / Commercial) ;
- l'affectation des commerciaux dans des squads ;
- la gestion des ventes saisies par les commerciaux ;
- l'affichage de statistiques et classements ;
- la consultation et l'envoi de messages internes.

### Hors périmètre de la version actuelle

Afin d'adapter le projet à la durée du BE, certains éléments ne seront pas pris en charge :

- pas de connexion à un système métier existant (ERP/CRM) ;
- pas de gestion de catalogue produits ;
- pas d'avis clients ;
- pas de données externes de vente.

Ces fonctionnalités représentent des pistes d'amélioration pour une version future.

## 2. Acteurs et description des utilisateurs

### 2.1 Liste des acteurs

Le système concerne trois types d'utilisateurs ayant des rôles et droits distincts :

- **Administrateur**
- **Gestionnaire** (aussi appelé *Manager* ou *Tuteur*)
- **Commercial**

Ces acteurs interagissent avec l'application selon des responsabilités spécifiques.

### 2.2 Rôles et responsabilités des acteurs

#### Administrateur

##### Rôle :

L'administrateur est responsable de la gestion générale du système.

Il supervise l'ensemble des utilisateurs et des squads commerciales.

##### Responsabilités dans l'application :

- Gérer les comptes (ajout, modification, désactivation)
- Créer et configurer les squads
- Affecter un gestionnaire à une squad
- Affecter des commerciaux aux squads
- Accéder à l'ensemble des statistiques globales

##### Besoins principaux :

- Avoir une vision complète de l'activité
- Assurer l'organisation des équipes
- Garantir le bon fonctionnement et la sécurité de la plateforme

#### Gestionnaire

##### Rôle :

Le gestionnaire encadre une squad (équipe de commerciaux).

Il suit les performances de chaque membre de son équipe.

##### Responsabilités dans l'application :

- Consulter le tableau de bord de son équipe
- Accéder aux ventes saisies par ses commerciaux
- Visualiser les classements internes
- Communiquer avec son équipe via la messagerie

##### Besoins principaux :

- Accès rapide à la performance de l'équipe
- Outils d'analyse et de supervision
- Communication efficace avec les commerciaux

#### Commercial

##### Rôle :

Le commercial saisit ses ventes et suit ses propres performances.

##### Responsabilités dans l'application :

- Se connecter à son espace dédié
- Saisir et consulter son historique de ventes
- Consulter son classement au sein de la squad

- Lire les messages reçus du gestionnaire
- Modifier son mot de passe

**Besoins principaux :**

- Saisie simple et rapide des ventes
- Interaction avec son gestionnaire

### 3. Besoins fonctionnels (ce que le système fait)

#### 3.1 Cas d'utilisation – Administrateur

- **Se connecter**
- **Gérer les utilisateurs**
  - Ajouter un utilisateur
  - Modifier un utilisateur
  - Désactiver / supprimer un utilisateur
- **Gérer les squads (équipes)**
  - Créer une squad
  - Affecter un gestionnaire
  - Affecter les commerciaux
- **Consulter les statistiques globales**
- **Générer des rapports de performance**

#### 3.2 Cas d'utilisation – Gestionnaire

- Se connecter
- Consulter le tableau de bord de son équipe
- Consulter les ventes de son équipe
- Consulter le classement interne
- Envoyer des messages aux commerciaux
- Filtrer par période (dates)

#### 3.3 Cas d'utilisation – Commercial

- Se connecter
- Saisir une vente
- Consulter l'historique des ventes
- Consulter ses statistiques personnelles
- Consulter son classement dans l'équipe
- Lire les messages reçus du gestionnaire
- Modifier son mot de passe

## 4. Besoins non fonctionnels

Les besoins non fonctionnels décrivent les contraintes de qualité que doit respecter l'application, indépendamment des fonctionnalités offertes. Ils concernent notamment la **sécurité**, la **performance**, l'ergonomie, la compatibilité et la fiabilité des données.

### 4.1 Sécurité

La sécurité est un aspect essentiel du système, même dans le cadre d'un projet pédagogique.

L'application doit respecter les exigences suivantes :

- **Authentification par identifiant et mot de passe**  
L'accès à la plateforme est réservé aux utilisateurs enregistrés. Chaque utilisateur dispose d'un compte personnel protégé par un mot de passe.
- **Protection des mots de passe**  
Les mots de passe ne doivent pas être stockés en clair dans la base de données. Ils seront enregistrés sous forme **hachée** (par exemple avec une fonction de type bcrypt), afin de limiter les risques en cas de fuite de données.
- **Gestion des rôles**  
Trois rôles sont prévus : **ADMIN**, **GESTIONNAIRE** et **COMMERCIAL**.  
Le système doit s'assurer que :
  - l'administrateur peut gérer l'ensemble des utilisateurs et des squads ;
  - le gestionnaire n'a accès qu'aux informations de sa propre squad ;
  - le commercial n'a accès qu'à ses propres ventes et messages.
- **Contrôle d'accès aux données**  
Les pages et les fonctionnalités doivent être protégées de manière à empêcher :
  - qu'un utilisateur non authentifié accède au système ;
  - qu'un utilisateur accède à des données qui ne le concernent pas (autres équipes, autres commerciaux, etc.).

### 4.2 Performance

Même si l'application est développée dans un contexte pédagogique, elle doit rester **réactive** et agréable à utiliser.

- Les pages principales (tableaux de bord, listes de ventes, messagerie) doivent s'afficher dans un temps raisonnable, par exemple **inférieur à 2–3 secondes** dans des conditions normales d'utilisation.
- Les traitements lourds (calcul de statistiques, classements) doivent être optimisés autant que possible (requêtes SQL adaptées, filtrage par période, etc.).

L'objectif n'est pas d'optimiser à l'extrême, mais d'éviter une application trop lente ou difficile à utiliser.

### 4.3 Ergonomie et expérience utilisateur (UX)

L'application doit être simple d'utilisation pour tous les types d'utilisateurs, même non techniques.

- **Interface claire et lisible**  
Les pages doivent être structurées (titres, tableaux, boutons bien identifiés), avec une hiérarchie visuelle permettant de repérer rapidement les informations importantes (statistiques, messages, ventes, etc.).
- **Navigation cohérente**  
La navigation entre les différents espaces (**Administrateur**, **Gestionnaire**, **Commercial**) doit être intuitive :
  - accès rapide aux principales fonctionnalités ;
  - menus ou boutons clairement étiquetés ;

- retour simple au tableau de bord.

L'objectif est de limiter le nombre de clics nécessaires pour accéder aux informations essentielles.

#### 4.4 Compatibilité

L'application est destinée à être utilisée principalement sur des postes informatiques classiques.

- L'interface doit être **compatible avec les navigateurs récents**, tels que :
  - Google Chrome
  - Mozilla Firefox
  - Microsoft Edge
- La résolution cible principale est celle d'un écran PC ou d'un ordinateur portable.  
Une adaptation responsive (tablette, smartphone) peut être envisagée comme amélioration, mais n'est pas prioritaire dans le cadre de ce projet.

#### 4.5 Fiabilité et intégrité des données

Les données manipulées (utilisateurs, squads, ventes, messages) doivent être stockées de manière **fiable et cohérente** dans la base de données.

- **Intégrité référentielle**

Les clés primaires (PK) et clés étrangères (FK) définies dans le MCD/MLD doivent être respectées afin de garantir :

- qu'une vente est toujours associée à un commercial existant ;
- qu'un message possède toujours un expéditeur et un destinataire valides ;
- qu'une squad référencée existe réellement.

- **Conservation des données importantes**

Une fois une vente saisie et validée, elle ne doit pas être perdue accidentellement.

La suppression ou la modification de données critiques (ventes, utilisateurs) doit être encadrée (actions réservées à l'administrateur ou au gestionnaire selon le cas).

Ces exigences visent à garantir la cohérence du système et la fiabilité des résultats affichés dans les statistiques et tableaux de bord.

## 5. Interfaces utilisateur (maquettes)

Cette section présente les différents écrans prévus dans l'application.

L'objectif est de donner une **vision d'ensemble de l'interface** et de l'expérience utilisateur, avant la réalisation des maquettes détaillées.

### 5.1 Liste des écrans prévus

L'application comprendra les écrans suivants :

Écran	Acteurs concernés	Description
<b>Page de connexion</b>	Tous	Saisie de l'email et du mot de passe pour accéder au système
<b>Tableau de bord Administrateur</b>	Administrateur	Accès à la gestion des utilisateurs, squads et statistiques globales
<b>Page de gestion des utilisateurs</b>	Administrateur	Création, modification, désactivation des comptes
<b>Page de gestion des squads</b>	Administrateur	Création de squads et affectation des gestionnaires/commerciaux
<b>Tableau de bord Gestionnaire</b>	Gestionnaire	Visualisation de l'équipe, des ventes et des classements
<b>Page des ventes de l'équipe</b>	Gestionnaire	Consultation détaillée des enregistrements par commercial
<b>Système de messagerie interne</b>	Gestionnaire / Commercial	Envoi et consultation des messages internes
<b>Tableau de bord Commercial</b>	Commercial	Consultation des statistiques personnelles et classement
<b>Page de saisie des ventes</b>	Commercial	Formulaire simple pour enregistrer une nouvelle vente
<b>Page profil / modification mot de passe</b>	Commercial	Gestion des informations de connexion personnelles

## 6. Modélisation des données et du système

La modélisation du système repose sur deux approches complémentaires :

- **UML** pour modéliser la structure objet et les interactions
- **Merise** pour modéliser les données et leur organisation en base

Ces modèles permettent de structurer le système avant son développement.

### 6.1 Modélisation UML — Diagramme de classes

UML (Unified Modeling Language) est un langage standard de modélisation objet utilisé pour représenter les éléments d'un système et leurs relations.

Dans ce projet, le **diagramme de classes** permet de définir :

- les **classes principales** : Utilisateur, Squad, Vente, Message
- leurs **attributs**
- leurs **associations et cardinalités**

Ce schéma représente la structure statique du système.

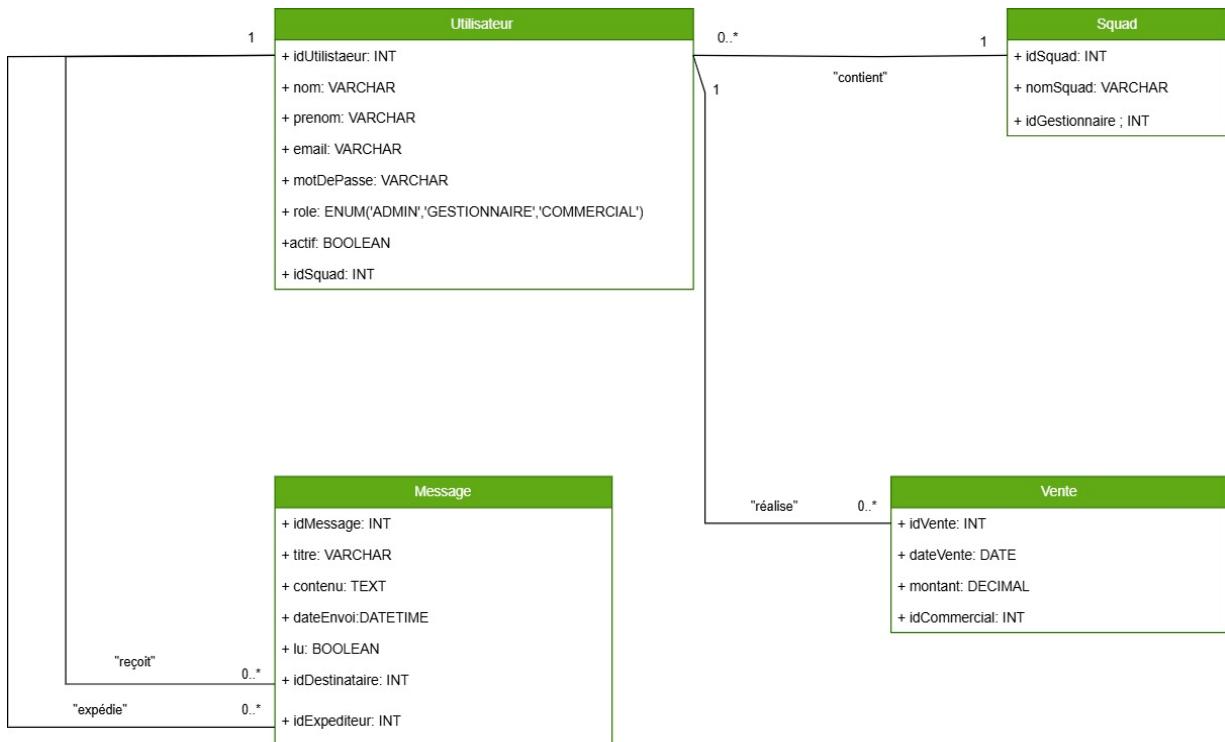


Figure 1 : Diagramme de classes UML

## 6.2 Modèle Conceptuel de Données — MCD

Le MCD (Méthode Merise) représente les données du système **au niveau conceptuel**, sans contrainte technique.

Il identifie :

MCD	Rôle
Entités	Objets manipulés par le système
Relations	Liens logiques entre entités
Cardinalités	Contraintes d'association

Dans notre système, le MCD contient :

- **Entités** : UTILISATEUR, SQUAD, VENTE, MESSAGE
- **Relations** : APPARTIENT\_A, REALISE, ENVOIE, RECOIT

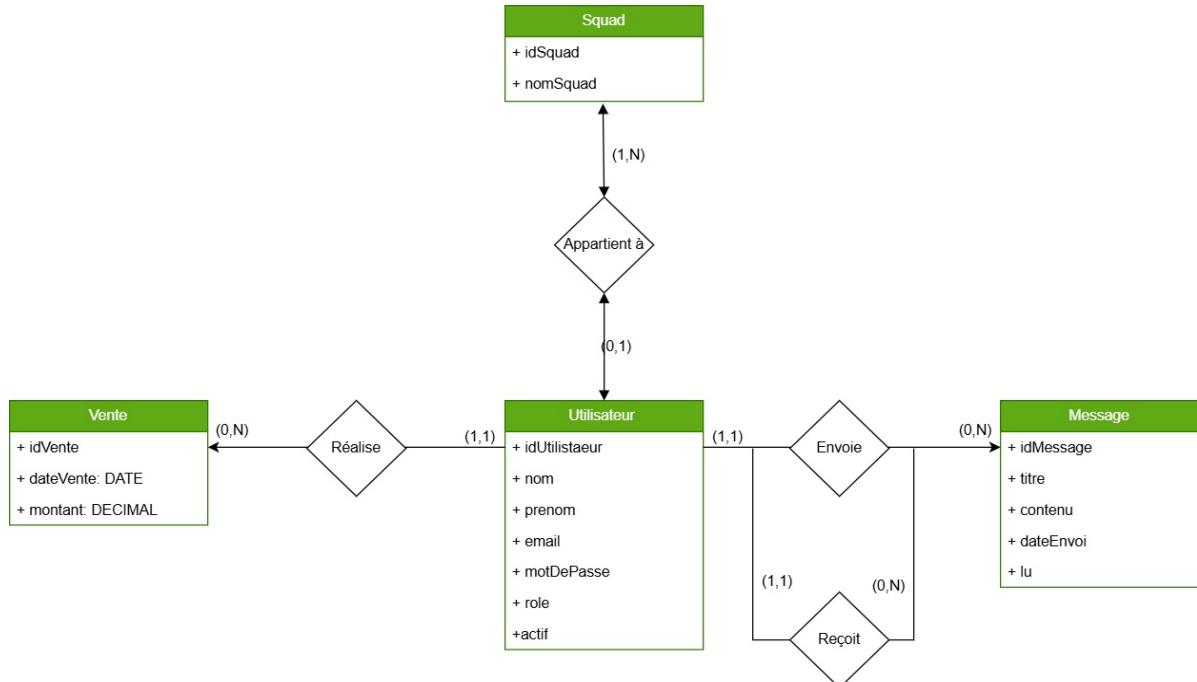


Figure 2 : MCD (Modèle Conceptuel de Données)

### 6.3 Modèle Logique de Données — MLD

Le MLD traduit le MCD en tables relationnelles avec **clés primaires (PK)** et **clés étrangères (FK)**.

Les tables définies sont :

- UTILISATEUR
- SQUAD
- VENTE
- MESSAGE

Les contraintes garantissent l'intégrité des données (une vente appartient à un commercial existant, un message a un expéditeur valide, etc.).

Table : UTILISATEUR

Attribut	Type	Contrainte	Description
<b>idUtilisateur</b>	INT	PK	Identifiant unique de l'utilisateur
<b>nom</b>	VARCHAR(100)	NOT NULL	Nom de l'utilisateur
<b>prenom</b>	VARCHAR(100)	NOT NULL	Prénom de l'utilisateur
<b>email</b>	VARCHAR(150)	UNIQUE, NOT NULL	Email de connexion
<b>motDePasse</b>	VARCHAR(255)	NOT NULL	Mot de passe haché
<b>role</b>	ENUM('ADMIN','GESTIONNAIRE','COMMERCIAL')	NOT NULL	Rôle dans le système
<b>actif</b>	BOOLEAN	NOT NULL (DEFAULT TRUE)	Statut du compte
<b>idSquad</b>	INT	FK → SQUAD(idSquad), NULL	Squad à laquelle appartient l'utilisateur

Table : SQUAD

Attribut	Type	Contrainte	Description
<b>idSquad</b>	INT	PK	Identifiant de la squad
<b>nomSquad</b>	VARCHAR(100)	NOT NULL	Nom de l'équipe commerciale
<b>idGestionnaire</b>	INT	FK → UTILISATEUR(idUtilisateur)	Gestionnaire responsable

Table : VENTE

Attribut	Type	Contrainte	Description
<b>idVente</b>	INT	PK	Identifiant de la vente
<b>dateVente</b>	DATE	NOT NULL	Date d'enregistrement
<b>montant</b>	DECIMAL(10,2)	NOT NULL	Montant de la vente
<b>idCommercial</b>	INT	FK → UTILISATEUR(idUtilisateur)	Commercial ayant réalisé la vente

Table : MESSAGE

Attribut	Type	Contrainte	Description
<b>idMessage</b>	INT	PK	Identifiant du message
<b>titre</b>	VARCHAR(150)	NOT NULL	Sujet rapide
<b>contenu</b>	TEXT	NOT NULL	Contenu du message
<b>dateEnvoi</b>	DATETIME	DEFAULT CURRENT_TIMESTAMP	Date d'envoi automatique
<b>lu</b>	BOOLEAN	DEFAULT FALSE	Indique si le message a été lu
<b>idExpediteur</b>	INT	FK → UTILISATEUR(idUtilisateur)	Utilisateur qui envoie
<b>idDestinataire</b>	INT	FK → UTILISATEUR(idUtilisateur)	Utilisateur qui reçoit

Le MLD permettra ensuite de produire le **script SQL de création de base**, qui sera présenté en annexe du rapport technique.

## 7. Technologies prévues

Le développement de l'application repose sur des technologies modernes et couramment utilisées dans le développement web. Le choix des outils répond aux objectifs du projet : performance, simplicité de mise en œuvre et respect des normes actuelles.

### Frontend (Interface utilisateur)

- **HTML5, CSS3, JavaScript**
  - pour la construction des pages et les interactions dynamiques

L'objectif est de proposer une interface moderne et intuitive, accessible via un simple navigateur web.

### Backend (Logique applicative)

- **Node.js** avec le framework **Express.js**
  - Permet la création d'une API web rapide, modulaire et évolutive

Le serveur gère :

- l'authentification des utilisateurs
- la logique de gestion des ventes, squads et messages
- les rôles et permissions
- la communication avec la base de données

### Base de données

- **MySQL**
  - Base de données relationnelle robuste, adaptée au modèle MLD établi via Merise

Elle assure l'intégrité des données via l'utilisation :

- des clés primaires (PK)
- des clés étrangères (FK)
- des contraintes d'intégrité référentielle

### Sécurité

- **brypt** : pour le hachage sécurisé des mots de passe
- **JWT (JSON Web Token)** : pour la gestion sécurisée des sessions utilisateurs et des rôles

Ces technologies garantissent un accès contrôlé aux données selon le rôle de l'utilisateur (ADMIN / GESTIONNAIRE / COMMERCIAL

### Visualisation des données

- **Chart.js**
  - affichage graphique des classements et statistiques commerciales

Son usage permet une meilleure interprétation des performances par les gestionnaires et les commerciaux.

## 8. Planning prévisionnel

Le développement du projet suit un cycle structuré en plusieurs phases successives. Chaque phase correspond à une étape importante de l'ingénierie logicielle, depuis l'analyse des besoins jusqu'aux tests et à la préparation de la soutenance.

Le planning prévisionnel est le suivant :

Phase	Intitulé	Contenu	Durée estimée
Phase 1	Analyse et conception	Rédaction cahier des charges, UML (cas d'utilisation, classes) et MCD	2 semaines
Phase 2	Mise en place de la base de données	MLD, script SQL et création du schéma de base	1 semaine
Phase 3	Développement du backend	API Node.js/Express, gestion des rôles, opérations CRUD	3 semaines
Phase 4	Développement du frontend	Interfaces utilisateur (Admin/Manager/Commercial)	3 semaines
Phase 5	Tests et corrections	Tests fonctionnels, corrections anomalies, optimisation	1 semaine
Phase 6	Rédaction du rapport et soutenance	Finalisation documentation + préparation présentation orale	1 semaine

Remarques :

- Ce planning pourra être **ajusté** en fonction des difficultés rencontrées et du rythme d'avancement.
- Certaines phases pourront se **chevaucher** (ex : début du frontend pendant la fin du backend), afin d'optimiser le temps de réalisation.

## 9. Conclusion

Ce projet s'inscrit dans le cadre du Bureau d'Études en Informatique et a pour objectif la conception et le développement d'une plateforme web permettant la gestion et l'évaluation des performances de commerciaux au sein d'équipes organisées en squads. Cette application offrira un suivi structuré des ventes, une visualisation des statistiques et une communication interne optimisée entre les gestionnaires et les commerciaux.

La réalisation de ce projet permet de mettre en pratique l'ensemble des compétences acquises en matière d'analyse des besoins, de modélisation (UML, Merise), de développement web et de gestion de base de données. Il constitue également une mise en situation professionnelle concrète, applicable au monde de l'entreprise où la performance commerciale représente un enjeu majeur.

À terme, la solution pourra être enrichie par des fonctionnalités supplémentaires telles que la gestion des objectifs commerciaux, l'intégration d'évaluations clients, l'ajout d'autres indicateurs de performance ou encore la synchronisation avec des systèmes commerciaux existants (ERP/CRM).

Ce Cahier des Charges constitue une base solide pour la poursuite du projet et guidera l'ensemble des étapes techniques jusqu'à la soutenance finale.